

**Santa Clara University
Scholar Commons**

Interdisciplinary Design Senior Theses

Engineering Senior Theses

6-6-2014

Distributed smart camera network for safety and security

Nathan Fox
Santa Clara University

Matthew Kelley
Santa Clara University

Christopher Rapa
Santa Clara University

Christopher Yarp
Santa Clara University

Follow this and additional works at: https://scholarcommons.scu.edu/idp_senior

 Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Fox, Nathan; Kelley, Matthew; Rapa, Christopher; and Yarp, Christopher, "Distributed smart camera network for safety and security" (2014). *Interdisciplinary Design Senior Theses*. 6.
https://scholarcommons.scu.edu/idp_senior/6

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Interdisciplinary Design Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact rscroggin@scu.edu.

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING

Date: June 4, 2014

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Christopher Yarp
Nathan Fox
Matthew Kelley
Christopher Rapa

ENTITLED

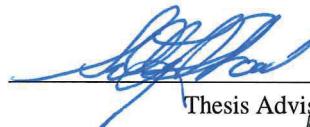
Distributed Smart Camera Network for Safety and Security

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

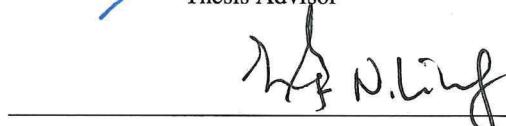
BACHELOR OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING
BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING



Thesis Advisor



Thesis Advisor



Department Chair



Department Chair

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING

Date: June 6, 2014

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Christopher Yarp
Nathan Fox
Matthew Kelley
Christopher Rapa

ENTITLED

Distributed Smart Camera Network for Safety and Security

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING
BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING

Thesis Advisor

Thesis Advisor

Department Chair

Department Chair

Distributed Smart Camera Network for Safety and Security

by

Christopher Yarp
Nathan Fox
Matthew Kelley
Christopher Rapa

Submitted in partial fulfillment of the requirements
for the degrees of
Bachelor of Science in Computer Science & Engineering
Bachelor of Science in Electrical Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 6, 2014

Distributed Smart Camera Network for Safety and Security

Christopher Yarp
Nathan Fox
Matthew Kelley
Christopher Rapa

Department of Computer Engineering
Department of Electrical Engineering
Santa Clara University
June 6, 2014

ABSTRACT

Current CCTV surveillance solutions are generally retrospective tools. Because real time use of CCTV requires human monitors to view a potentially exorbitant number of video feeds, CCTV is usually only useful after an incident has occurred. However, new technologies are making it possible for machines to perform some tasks that previously required a human monitor. The proposed project seeks to augment existing CCTV systems with behavioral analytics. The system uses a series of cameras, FPGAs, and computers to track object movement throughout a facility. This information is used to build a model of normal movement. Object movements are compared against this model and any ones that diverge from the model are flagged for review by security personnel.

Table of Contents

Table of Contents	iv
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Background and Motivation	1
1.2 Project Overview	1
1.3 Motivation	2
1.4 Significance	3
2 System Overview	4
2.1 High Level Architecture	4
2.1.1 Dataflow Example	4
2.2 System Component Descriptions	5
2.2.1 IP Camera	6
2.2.2 FPGA Back-end	6
2.2.3 Identity Engine	7
2.2.4 Database	7
2.2.5 Behavioral Analysis Engine	7
2.2.6 User Interface	8
2.3 Scope and Current Progress	8
2.3.1 System Progress	9
2.3.2 FPGA Progress	9
2.3.3 Identity Engine Progress	9
2.3.4 Database and Behavioral Analysis Engine Progress	9
2.3.5 User Interface Progress	10
2.3.6 Expected Results	10
3 Real World Constraints	11
3.1 Power	11
3.2 Network	12
3.3 FPGA Capacity and Speed	13
3.4 Database Connections and Storage	13
3.5 Behavioral Analysis Engine Processing	14
3.6 User Interface Scalability	14
3.7 Target Identification	14
4 Development Bill of Materials (BOM) & Budget	16
4.1 Real World Cost Estimates and BOM	17

5	FPGA Back-end Design and Implementation	20
5.1	Purpose	20
5.2	Scoping	21
5.2.1	FPGA Paradigm Shifts	21
5.2.2	Current Progress	22
5.3	Development Platform Overview	22
5.3.1	Hardware Overview	22
5.3.2	Software Overview	22
5.3.3	Embedded Software	23
5.4	FPGA Back-end Design - Data Flow Perspective	24
5.5	FPGA Back-end Design - Software Perspective	25
5.6	FPGA Back-end Design - Hardware Perspective	25
5.7	Video Format Selection	28
5.8	Object Detection - Contour Forming	28
5.8.1	General Method	28
5.8.2	Contour Lines vs. Edges	29
5.9	Marching Squares	30
5.10	Marching Squares Design	35
5.11	Object Detection Simulation	36
5.12	FPGA Back-end Dataflow Proof of Concept	40
6	Identity Engine Design and Implementation	43
6.1	Software Architecture	43
6.2	Canny Edge Detection	46
6.3	Facial Recognition	46
6.4	Testing	47
7	Database	48
7.1	Setup	48
7.2	Schema	48
7.2.1	Schema Modifications	49
8	Behavioral Analysis Engine Design and Implementation	51
8.1	Weighted Graph Analysis	51
8.2	Frame Duration Analysis	53
8.2.1	Implementation	53
8.2.2	Implementation Complications	54
8.2.3	Constraints and Assumptions	55
8.3	Exit and Entry Type Analysis	55
9	The User Interface	57
9.1	Mock-up Design	57
9.2	Requirements	59
9.2.1	Functional Requirements	59
9.2.2	Non-Functional Requirements	60
9.3	Implementation	60
9.4	Technologies Used	62
9.5	Risk Management	62
10	Test Plan	64
10.1	Unit Tests	64
10.2	Integration Tests	64
10.3	Test Media	64
10.3.1	System Testing	65
10.4	Test Results	65

11 Ethics	66
11.1 Societal Implications	66
11.2 Project Development	67
11.3 Team and Organization	68
11.4 Summary	68
12 Environmental Impact and Sustainability	69
12.1 Environmental Impact	69
12.1.1 Semiconductor (ICs)	69
12.1.2 Copper	69
12.1.3 Reducing Embodied Energy and Greenhouse Gas Emissions	69
12.2 Sustainability	70
12.3 Frugal Engineering	71
13 Aesthetics	73
13.1 Aesthetic Considerations for Productization of Project	73
13.1.1 Graphical User Interface (GUI)	74
13.1.2 Hardware	75
14 Conclusion	77
15 Bibliography	79
16 Appendix	82
16.1 FPGA Specifications	82
16.2 Object Detection - Matlab Simulation Code	83
16.3 FPGA Demo - Hardware Description	88
16.3.1 Directory Structure	88
16.3.2 Qsys System - GUI View (orion_system.qsys)	88
16.3.3 Qsys System - Code View (orion_system.qsys)	90
16.3.4 Top Level Verilog (orion_top.v)	120
16.3.5 Timing Constraint File (timing.base.sdc)	129
16.3.6 Quartus II Project File (orion_system.qpf)	132
16.3.7 Quartus II Settings File (orion_system.qsf)	132
16.3.8 Demo IP Top Verilog File (test_ip_top.v)	159
16.3.9 Demo IP Top Verilog File (test_ip.v)	166
16.3.10 Demo IP _hw.tcl File (IPTest_hw.tcl)	171
16.3.11 Avalon-MM Custom Master - Altera Template (custom_master.v)	178
16.3.12 Avalon-MM Burst Read Master - Altera Template (burst_read_master.v)	181
16.3.13 Avalon-MM Latency Aware Read Master - Altera Template (latency_aware_read_master.v)	185
16.3.14 Avalon-MM Write Master - Altera Template (write_master.v)	189
16.3.15 Avalon-MM Burst Write Master (burst_write_master.v)	192
16.3.16 Avalon-MM Interrupt Logic - Altera Template (interrupt_logic.v)	197
16.3.17 Avalon-MM Slave Template - Altera Template (slave_template.v)	199
16.3.18 Avalon-MM Slave Template Macros - Altera Template (slave_template_macros.h)	212
16.4 FPGA Demo - Software Code	214
16.4.1 Partial Directory Structure	214
16.4.2 Senior Design Test BitBake Recipe (smart-cam-ipitest_1.0.bb)	215
16.4.3 Senior Design Test Application Source (test.c)	215
16.4.4 Senior Design Layer Configuration File (layer.conf)	219
16.4.5 Modified bblayers File (bblayers.conf)	219
16.4.6 Modified Image File (altera-gsrd-image.bb)	219
16.4.7 Modified Device Tree File (socfpga.dts)	220
16.4.8 SD Card Flash Script (make-sd.sh)	227
16.5 Database - SQL Table Creation Code	227

16.6 Behavioral Analysis Engine - Primary File	229
16.7 Behavioral Analysis Engine - Database Configuration File	231
16.8 User Interface - Primary Form	249
16.9 User Interface - Primary Form Designer File	255
16.10 User Interface - Add Stream Interface	261
16.11 User Interface - Add Stream Interface Designer File	262
16.12 User Interface - Event Object Class	266
16.13 User Interface - RSTPStream Class	267
16.14 User Interface - Database Configuration File	268
16.15 Identity Engine - Matlab Demo Code	292

17 Appendix - FPGA Lab Notebook

296

List of Figures

2.1	High Level System Architecture and Data Flow Diagram	5
5.1	Example Target	21
5.2	Cyclone V SoC Development Kit - Image Courtesy Altera Corporation [1]	23
5.3	FPGA Back-end Data Flow Diagram	24
5.4	FPGA Back-end Software Data Flow Diagram	25
5.5	FPGA Back-end Hardware Architecture, Adapted from Altera Reference Design Diagram [2]	27
5.6	Marching Square Look Up Table	31
5.7	Original Image	32
5.8	Image After Comparison to Threshold 0x45	33
5.9	Image After Evaluating 2x2 Pixel Blocks for Contour Segments	34
5.10	Layout Processing Units and Connections Between Them	37
5.11	Processing Units After Setup	38
5.12	Processing Units During Step 1	38
5.13	Processing Units After Final Step	39
5.14	Simulation of Contour Based Object Detection	40
5.15	Simulation of Contour Based Object Detection - Zoomed	41
6.1	Sample Image as Received From FPGA	43
6.2	Image After Canny Edge Detection	44
6.3	Illustration of Outer Maxima and Ratio of Radii (Highlighted in Purple)	45
6.4	Bounding Circles Highlighted in Orange	45
7.1	Original Draft of the Database Schema	49
7.2	Final Schema Design	50
8.1	Sample of a Directed Weighted Graph	52
8.2	Sample of a Normal Distribution (Image Courtesy of Mathisfun.com [3])	54
9.1	Sample Layout of the User Interface	57
9.2	Recent Layout of the User Interface	60
9.3	User Interface Implementation Risks	63
13.1	User Interface Concept	74
16.1	Qsys Design - GUI View (orion_system.qsys)	89

List of Tables

4.1	Development BOM - For Development Setup + Development Tools	18
4.2	FPGA Estimated BOM - Based Largely on Cyclone V SoC Dev Kit Part List [4]	19
4.3	Server Estimated BOM - Based on Online Quote from Colfax International [5]	19
5.1	Video Formats	28
16.1	FPGA (5CSXFC6D6F31C8NES) Overview [6]	82
16.2	FPGA (5CSXFC6D6F31C8NES) Resource Table [6]	83

Chapter 1

Introduction

1.1 Background and Motivation

Many companies and organizations employ Closed Circuit Television (CCTV) systems for security and safety reasons. The current CCTV technology is largely passive and is mostly used for reviewing a course of events after an incident has occurred. Organizations that wish to react in real time to developing situations are limited by the passive nature of CCTV. Time sensitive events such as medical emergencies and security breaches rely on surveillance staff to manually view the video feeds and look for potential problems. In many installations of these systems, the number of cameras can easily outnumber attentive monitoring personnel. This often lead to situations where incidents go unnoticed by staff, further cementing CCTV systems as primarily retrospective tools. However, recent advancements in high density re-configurable hardware (FPGAs), parallel processing systems (GPUs), software, and other current fields of research have made it possible to develop automated and distributed systems that can operate without the need for constant human monitoring. This presents an opportunity to develop new systems that leverage the power of these new technologies to address the shortcomings of existing CCTV systems.

1.2 Project Overview

The proposed project is based around fixing the shortcomings of the currently used security systems by augmenting them with behavioral model processing. This “smart” surveillance system builds on the existing CCTV systems by attaching small processing modules to each camera. These processing modules monitor the video feeds, identifying objects and movement. Information about the movement of objects within the frame of view for a camera and information about the movement between cameras is collected and used to create a behavioral model. This information is accumulated over time and is formed into a definition of normal behavior. After this definition is created the real time object movements are compared against the normal behavioral and checked for abnormal behavior. If the system detects an anomaly, a notification is sent

to security informing them of the situation. The system is not meant to replace security staff but to provide them with a unique tool that can help direct their attention and increase safety and security in real time.

1.3 Motivation

The project was originally intended to provide real time behavioral analysis to enhance crime prevention and security. However two incidents that took place during the implementation of this project have helped to inspire and motivate the creation of the system by realizing its potential applications. The first of these events occurred in late September / early October 2013 where a patient at a San Francisco Hospital tragically passed away in an exterior stairwell [7]. While a surveillance system was installed at the hospital [7], the patient was not seen entering the stairwell by monitoring staff. During the investigation of this missing patient officials were hampered by hardware problem while attempting to review the CCTV footage [7]. The other inspiring incident occurred in April 2014 where a 15 year old boy was able to breach perimeter security at Mineta San Jose International Airport and crawl into the wheel well of a jet destined for Hawaii [8]. The boy miraculously survived the flight [8] but the fact that he was able to breach airport security unnoticed is a disturbing realization. The airport, similar to the hospital, had a surveillance system in place before this event occurred. The boy was spotted by a camera while walking to the plane [8], however no surveillance staff were able to catch this event until he climbed out of the plane in Hawaii [8].

Both of these incidents highlight the flaws of the current CCTV systems and need for smarter surveillance systems. The hospital incident could have been prevented by a smart camera system that analyzed foot traffic by identity. This system would have realized that the patient left their room and had wandered into a section of the hospital where foot traffic was uncommon. Additionally, a behavioral analysis algorithm using time could have identified that the patient entered the stairwell but did not come out within a reasonable time frame. If these warnings had been communicated to monitoring staff or nurses, perhaps the patient would have survived. Similarly, if a smart camera system was installed at Mineta San Jose International Airport it could have identified that an individual was walking through the secured area without passing through one of the standard entry points. In addition, if the boy was running in the secured area, the system should have also identified abnormal movement rates. While the risk was relatively low in this case, it shows a security weakness in airport security which could be exploited in the future for more malicious reasons. The system being proposed here aims to address the deficiencies with current CCTV systems in both scenarios.

1.4 Significance

A CCTV system with semi-automated behavioral monitoring would have several distinct benefits over existing passive CCTV systems:

- Increased Personal Safety: In hospitals, assisted living facilities, schools and playgrounds, the system could monitor patients and children at times and in locations where staff may be unavailable or outnumbered. For example, the system could alert staff if a confused patient wanders away from their room during the night.
- Increased Security: In security applications, the system could identify criminals who had bypassed perimeter security, something that is currently difficult with human monitors. The system could also enhance security by identifying individual abnormal behavior and analyzing movement rates.
- Increased Fairness: Since the system uses behavior analysis, it is not susceptible to the racial, gender, religious, and social prejudices of human monitors.

Chapter 2

System Overview

2.1 High Level Architecture

In order to accomplish the ambitious project, the system has been split into several different modules, each with a specialized role. The project begins with a raw video feed from the IP Cameras of an existing CCTV system. An image processing module located directly behind the camera known as the FPGA or camera Back-end takes the video feed, prepares it for analysis, and performs object detection. Once an object is detected the image is cropped around it and is sent to be identified. The next module, the Identity Engine, takes each cropped image from the FGPA and ascribes an identity to the pictured object (ex. with facial or object recognition). The information created in the system about the object up to this point is bundled and saved into a database. Next, another module called the Behavior Analysis Engine processes the information in the database and attempts to understand the behavior of the object. If a behavioral discrepancy is detected, it creates an anomaly entry in the database. The User Interface, the primary interaction between the system and user, then pulls this information and displays it to the monitoring staff. It should be noted that this project is a proof of concept and may make some assumptions about the existence of certain technologies. The effectiveness of this system is dependent on progress made in advancing fields of research including facial recognition and super resolution. The architecture of the system described above is outlined in Figure 2.1.

2.1.1 Dataflow Example

The operation of this system is quite complex due to the modular design and innovative ideas, however understanding the architecture is best done by explaining the flow of data within the system. An example commonly used as reference in implementation and during conversational descriptions of this project is a hospital deployment. It is assumed in this case study that many aspects of setup and initialization are already completed. The setup involves the ideal placement of IP Cameras and attached FPGA Back-ends (hallways, nurse stations, entrances, and exits) and an onsite server room. The initialization includes the existence of

enough already collected data concerning the movement and speed of individuals throughout the hospital.

Now we can begin analyzing the example a patient wandering out of their room at night and collapsing in the hall. The cameras, being positioned in the hallways, capture images of the patient exiting their room and entering the hallway. The FPGA Back-end detects the face of the patient and passes this information to the Identity Engine. The Identity Engine recognizes the patient and collects information on their movement creating an entry in the database. While this is happening, the Behavioral Analysis Engine is checking all new entries into the database to determine if anomalous behavior has occurred. The FPGA Back-end is constantly sending the Identity Engine small metadata updates on the patient's position. When the patient falls, they remain in the frame longer than would be expected in the behavioral model. Since the patient has collapsed, they have not exited the frame of view of the camera and no exit event has been recorded. The Behavioral Analysis Engine would detect that the patient had not left the camera's frame in the normal amount of time and generate an anomaly. The anomaly is entered into the database and picked up by the User Interface. The User Interface notifies the monitoring staff of this anomaly and directs them to the appropriate video feed. Action can then be taken to help the patient.

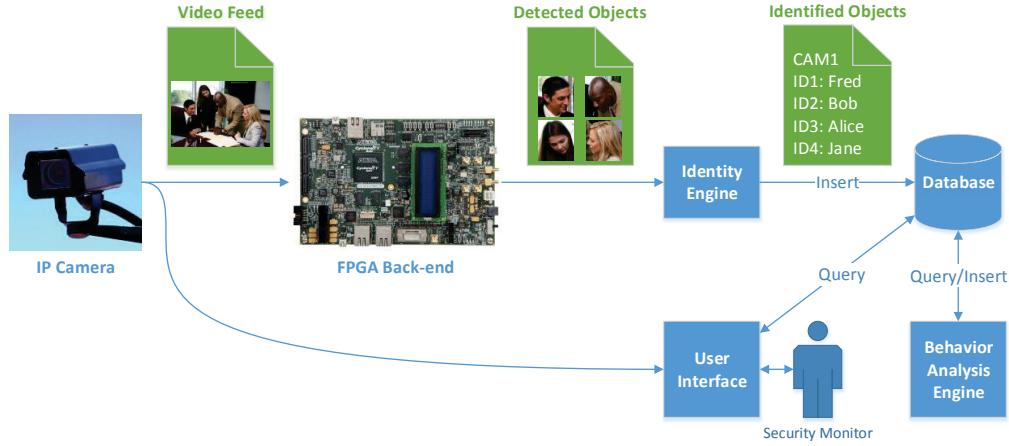


Figure 2.1: High Level System Architecture and Data Flow Diagram

2.2 System Component Descriptions

As the system is being designed modularly, each function of the system is assigned to a particular module. Detailed descriptions of each of these modules follow.

2.2.1 IP Camera

The purpose of the IP cameras is to capture video footage that will be analyzed by the rest of the system. IP cameras are one of the most common types of surveillance cameras used in modern CCTV installations. As one of the project goals is to be able to integrate with existing CCTV systems, IP cameras were chosen over other, less common, technologies. The project is only focusing on incorporating cameras that are ONVIF Profile S compliant. ONVIF is a security industry standard and Profile S is a sub-standard for IP cameras [9]. ONVIF guarantees a standard SOAP (Simple Object Access Protocol) interface to configure the cameras and request streams [10]. The cameras provide video streams in the form of MJPEG or H.264 compressed video over RTP/RTSP [9]. Focusing only on the cameras within the scope of this protocol allows us to ignore many of the compatibility issues that occur when developing a system for multiple camera vendors.

2.2.2 FPGA Back-end

The purpose of the FPGA Back-end is to receive a video stream from the existing IP Cameras and to ultimately detect objects within the video. When an object is detected, a message is sent to the Identity Engine with metadata and a cropped image of the object. The metadata includes the camera ID, a timestamp, the position in the image the object was detected, and the size of the image. In order to perform this object detection, the FPGA Back-end undergoes many steps each with a particular function. The operations conducted are listed in order of operation below:

1. Receives video from IP cameras
2. Processes video in preparation for analysis (ex. brightness / contrast)
3. Performs object detection
4. Crops image around objects
5. Performs additional processing on cropped image
6. Sends cropped images to Identity Engine
7. Continues to track objects

The FPGA Back-end is based on the Cyclone V SX SoC FPGA from Altera (5CSXFC6D6F31C8NES) [11]. This FPGA has the distinction of having a dual-core ARM Cortex A9 integrated as hard IP alongside the FPGA [12]. The ARM processor is utilized to run a version of embedded Linux. Linux software is used for managing the network functions of the FPGA Back-end as well as some of the video stream processing

(buffering and decompressing). The image processing is implemented on the FPGA and acts like a hardware peripheral from the perspective of the processor [2].

2.2.3 Identity Engine

The Identity Engine takes input from the FPGA Back-end in the form of cropped frames and assigns identities to them. First, cropped images of objects are received and compared to a set of known objects. In the case of the current implementation, this comparison is based on the arrangement of colors in the artificial target. Once the identity is determined, the Identity Engine queries the database to see if the particular object has already been seen in previous detections. At this point the database is queried for results on the existence of an object, if it has never been seen an identity is created. Then all the information created in the system thus far along with the queried information is grouped together to form an event. This event is then saved in the database for future analysis.

2.2.4 Database

The database is an essential back-end component for the system because it provides not only a place to store the vast quantities of data that the system will be generating, but it also provides a connection and method of communication between components of the project. The most important connection is to the Identity Engine. The Identity Engine sends new data to the database about the identities of objects and their events. The input of new data from the Identity Engine triggers the other database connections to begin analyzing. First, the Behavioral Analysis Engine pulls the information about object detection events and statistics model variables to begin analyzing. This data is used to detect anomalies by comparing new events against the saved models. The Behavioral Analysis Engine is constantly updating the saved behavioral statistical models based on new detection events. Simultaneously, the User Interface is pulling information from the database primarily to display recent anomalies and detection events. The User Interface also pulls information on initialization to setup the IP Camera connections. The database is based on SQL because it works well with the other modules of the system. This is primarily due the prebuilt libraries for SQL in the development languages of the modules within system.

2.2.5 Behavioral Analysis Engine

The Behavioral Analysis Engine is responsible for querying the database to compare recent movements against a model of normal movement. During the course of the project, the methods of determining abnormal behavior have changed dramatically. Directed Weighted Graph Analysis was originally intended to be implemented as our method of behavioral detection. However, a simpler statistical model was implemented

given the scale and experimental nature of the project. This behavioral analysis was termed Frame Duration Analysis, based on how it was implanted. This type of analysis determines anomalous behavior based on comparing the speed of a subject against a statistical model. This comparison can be made because statisticians have determined that human walking speed corresponds to a normal distribution [13]. Although the full extent of the behavioral analysis that was not implemented, the ideas associated with the original project are novel.

2.2.6 User Interface

The User Interface is the only module in the project which interacts with the monitoring staff. The User Interface attempts to display the most relevant information within the system to the users, primarily objects, events, anomalies, and video feeds. Although this is only meant to be a proof on concept project, the partial success of this project depends upon the User Interface being intuitive and efficient. The User Interface still had to meet the requirements set forth by the design of the system to display the potential of the system. It will be written using Visual Studio Ultimate 2013 for code management, Team Foundation Services for application life cycle management, and Git for revision control. It was written in the C# language and used many libraries, such as VLC ActiveX Plugin, Microsoft LINQ, and many other standard .Net Development libraries.

2.3 Scope and Current Progress

The number and complexity of components in this system make the development of a near-commercial solution unrealistic within the time-frame of the project. Instead, the system was designed as a proof of concept to prove the overall methodology and architecture rather than serve as a commercial product. In order to expedite the development of an operational system, certain components that would be expected in a commercial system were replaced with simplified components. The modular architecture of the system allows components to easily be replaced with more advanced or commercial versions in the future.

One of the changes involves replacing facial recognition with a simpler stand-in. Although facial detection and recognition are used in commercial applications their cost and complexity made them impractical for this proof of concept. An artificial target was designed instead to work with simpler object algorithms that could be implemented within the timeframe of this project. This target, shown in Figure 5.1 is an example of one of these targets.

Many different algorithms were proposed for the Behavioral Analysis Engine. One of the proposed algorithms involved the use of Directed Weighted Graphs to detect individuals taking uncommon paths through a facility. While it was agreed that this algorithm would likely yield useful information, the budget only al-

lowed for a three camera system which would not allow a proper evaluation of the algorithm's effectiveness. Instead, Frame Duration Analysis was implemented as the method of behavioral analysis for the purposes of this project.

Additional scoping information is provided in the detailed chapter for each component.

2.3.1 System Progress

The system is currently in the final stages of implementation. With the overall system architecture finalized, each module underwent its own design and implementation process. Due to time constraints with such an ambitious project along with unanticipated challenges, not all of the modules were completed by the end of the development period. While MATLAB based simulations were completed for the FPGA Back-end and for the Identity Engine, the FPGA and CUDA implementation are still in progress. The MATLAB simulations are used as stand-ins for the actual components in the initial end-to-end demonstration.

2.3.2 FPGA Progress

The FPGA Back-end was designed and various image processing algorithms were evaluated. MATLAB was used to simulate the functionality of the FPGA and to verify the effectiveness of the chosen image processing algorithm. The hardware development environment for the FPGA was brought up on a Windows machine and the software development environment was brought up on an Linux VM (virtual machine). A custom Linux distribution with the software libraries required by the FPGA Back-end was developed using the OE (Open Embedded) build environment. The hardware glue logic linking all the system components including the ARM processor, DDR3 memory, and custom IP was created using the Qsys system level integration tool. Example IP that conformed to the Avalon bus standard used by Qsys was created to test system functionality. Corresponding software was also written to exercise the custom IP. More details on the FPGA demo IP can be seen in section 5.12.

2.3.3 Identity Engine Progress

The Canny Edge Detection Algorithm was researched and a MATLAB simulation was created. Libraries for the decoding of images from the FPGA Back-end, for interaction with the database, and for communicating with the FPGA Back-end were also researched. A CUDA based implementation of the Identity Engine has been prototyped.

2.3.4 Database and Behavioral Analysis Engine Progress

For the extent of this project these sections are completed and working properly. That being said with a continuation of this project these modules defiantly should be revisited and improved. For more information

on future improvements these sections see their respective implementation chapters.

2.3.5 User Interface Progress

The User interface is working at a base level it can handle responses and accurately display the information being created by the system. During the implementation many assumptions about future progress were made that should be corrected. Also a few the functional requirements that were not implemented should be built upon.

2.3.6 Expected Results

It is expected that this system will be able to detect targets within the field of view of cameras, identify them, and place entries in the database. The system will be able to generate a behavioral model based on real-time movement data. After an initial initialization period, abnormal movements such as an object stopping for an unprecedented amount of time or moving exceedingly fast will be flagged as anomalous and reported via the User Interface.

As with any system that attempts to do pattern matching and automated statistical model generation, it is unreasonable to assume that the system will be anywhere near 100% accurate. False positives (normal behavior interpreted as anomalous) and false negatives (failure to detect anomalous behavior) will likely occur during normal operation of the system. However, all efforts will be made to make the system as accurate as possible.

Chapter 3

Real World Constraints

Since this project is a proof of concept, many of the real world constraints were circumvented during the implementation. In order to successfully deploy this system as a commercial product, these constraints will need to be addressed. These constraints can be as simple as budgetary issues or as complex as the effectiveness of a component of this project.

3.1 Power

All of the components used in the system require electricity. This includes components explicitly listed in this document as well as other general but necessary components like network switches. In order to ensure the best possible reliability and resilience, the server should be located in a server room with redundant power and an Uninterrupted Power Supply (UPS). The cameras and FPGA have their own power requirements but their placement largely determines how their needs must be considered. Some network cameras come with Power Over Ethernet (PoE) support which allows them to be powered over the network cable by a special PoE switch. A PoE system may need to be employed in certain scenarios to make the project viable. As a point of reference the PoE camera (D-Link DCS-2310L) used in the project has a maximum power consumption of 5.3 Watts [14]. The FPGA Back-ends also require power but, since they are also peripherals on the network in their current configuration, they could potentially be placed farther away from the cameras and closer to a power source. Based on the reference document for the development kit, a maximum of 62.172 Watts would be consumed by the components used in the project [4]. This value depends on the actual design flashed onto the FPGA as well as the workload conducted on the ARM. The FPGA development kit used in the project does not support PoE and further research would need to be conducted to determine if it is a viable option.

3.2 Network

The IP Cameras employed in the prototype system use 10/100 BASE-TX Ethernet [14]. This form of Ethernet has a maximum theoretical throughput of 100 Mbits/sec [15]. A simple test was conducted using the Live555 library's QoS measurement utility to determine the bitrate received from the cameras running in TCP mode. For a 33 second test, the minimum bitrate was 1082 Kbit/sec, the average was 5533 Kbit/sec, and the maximum was 10434 Kbit/sec. This means that the theoretical maximum amount of video data that could be transferred simultaneously over one line would be $\frac{1024*100 \text{ kBit/sec Ethernet}}{5533 \text{ kBit/sec/stream}} \approx 18.5$ streams. Better evaluations of capacity for multimedia do exist [16]. These metrics take into account the “bursty” network traffic [16]. However, additional characterization of the network traffic would be required to use the more complex models. For the purposes of this document, we will under-spec the theoretical capacity by half in an attempt account for network congestion and fluctuation in the media stream datarates. In that case, 9 MJPEG stream could share one line. 9 cameras is actually quite small for many applications. However, the FPGA crops images that are sent to the identity engine, drastically reducing load on the connections to the identity engine. The size of the cropped image depends on the size of the object in frame but a typical size would be 34.3 kB (PNG) based on a test performed with an object approximately three feet from the camera. The metadata passed from the FPGA Back-end would be exceedingly small as it only needs to contain the NTP timestamp (64 bit), the camera ID (64 bit), the FPGA ID (64 bit), the object ID (64 bit), the position of the object in the frame (2x 32 bit), and the width and height of the object (2x 32 bit). The metadata would only be approximately 48 bytes per message.

In order to utilize the load reduction of the FPGA Back-end to alleviate network congestion, the network must be set up in the following way:

- The FPGA and the camera it processes are all connected locally via a switch
- The FPGA and cameras are on the same subnet

The load reduction relies on the fact that devices on the same subnet can communicate directly without a router thanks to Layer 2. Since each camera can connect to the FPGA directly via a common switch, the stream data is not sent over the entire network, consuming bandwidth. The FPGA, after processing the video sends only a very small amount of information to the central Identity Engine which must traverse to the center of the network.

Note that this capacity analysis does not consider the case when a copy of the stream from each camera must be recorded or displayed at all times. There are many factors that can affect how this impact the capacity requirements of the network including the quality and compression of the video. If this implementation is

required, the maximum number of cameras per link will decrease. Scaling to Gigabit Ethernet may help to alleviate capacity concerns if a large quantity of video must be recorded. An alternative solution would be to distribute recording equipment so that they are connected to the same switch as the FPGAs and cameras.

3.3 FPGA Capacity and Speed

Since facial detection was not implemented on the FPGAs in this proof of concept, a figure on how many faces a given FPGA can be detected at once is unavailable. However, it is safe to assume that such a limitation does exist. There are a selection of FPGAs on the market. The FPGA used in the project is a high-end model of the Cyclone V line of FPGAs from Altera. The name “Cyclone” is used to represent Altera’s Low-Cost line of FPGAs with the midrange line referred to as “Arria” and the top-of-the-line being “Stratix”. In general, as one goes up in device class, the number of logic elements and potentially the speed grade of the device increases. For examples, the Cyclone V line ranges from FPGA with no transceivers and 25,000 LEs (Logic Elements) to FPGAs with an embedded ARM processor, 9 5-Gbps transceivers and 110 LEs [17]. The Aria V line goes up to 660 LEs with an ARM processor in its SX variant [17]. Going up in lines also introduces new features. It is possible that a larger and more costly FPGA than was used in the prototype would be required to detect all the faces in a typical scene. Additional characterization would be required to determine the optimal FPGA for the application.

3.4 Database Connections and Storage

One of the primary real world constraints of a database is the volume of data being stored and the rate at which data is being generated by the system. The initial observations of the prototype connections indicate a relatively small size per entry associated with the Detection Events, Anomaly and Individual Tables; however issues with quantity may arise when an operational system is implemented. The primary concern for data growth is expected to be the KnownIMG Table because the size of an image, even a cropped image, is exponentially larger than the other database entries. During commercial implementation of this project the object identification should be replaced with facial recognition. This may change the usage pattern of the database and the database performance should be reevaluated at that time. Fortunately, Microsoft SQL Server does support very large databases up to 524.272 petabytes in size [18], which is far more than would be expected in even the largest installations of this system. SQL Server also has features built in for dealing with large file storage and access [19]. Clustering of SQL Servers [20] could also be used to address scalability concerns.

Another issue that may arise during the commercialization of this project is the number of SQL queries

being executed by the system at any given time. Scalability issues may occur if multiple Identity Engines, User Interface, and Behavioral Analysis Engine instances are present in a system, all polling the database for information simultaneously. This database stress should be analyzed carefully for each deployment of the system. Fortunately, Microsoft SQL Server is specified to be able to support a maximum of 32,767 user connections [18], far more than would be expected in any installation. SQL Server is also a fast Relational Database Management System which can handle anywhere from hundreds to hundreds of thousands of transactions per second depending on the configuration [21].

3.5 Behavioral Analysis Engine Processing

The real world constraints associated with the Behavioral Analysis Engine mainly relate to the accuracy of its analysis. Users may expect the system to perform at or near 100% accuracy. Like any form of behavior modeling, the system does not provide 100% accuracy and the customer must be made aware of this fact.

The only technical issue that may arise is the amount of processing power required to continuously run the more sophisticated behavioral algorithms. This might pose a problem for implementation but, theoretically, should be divisible into unique threads, mitigating the issue.

3.6 User Interface Scalability

At the end of this project the User Interface will not be easily scalable because of dependencies created during implementation. During the commercialization of this project these dependencies will have to be removed to allow for multiple working environments. Some of these dependencies include the lack of a working installer, assumed database configuration and connection, file directory dependencies, and libraries not included in the files associated with the project.

3.7 Target Identification

There are four major factors that constrain real world target identification: lighting, distance, orientation, and number of targets. Lighting, distance and orientation affect the appearance of a target. Inconsistent appearance from frame to frame can cause erroneous target identification. Targets may either be misidentified, or not identified at all. It requires fairly sophisticated algorithms to combat these factors. For the purposes of testing, we held the targets at close range and parallel with respect to the camera lens. Additionally, the tests were conducted indoors under uniform lighting.

Total number of targets is a concern of the system capacity and throughput. For each target in the frame the FPGAs and Identity Engine GPUs must use a certain number of clock cycles to process it. Too many

targets may require too many clock cycles to complete the entire frame. This will propagate through the entire system as a delay. All delays affect system performance and may result in data degradation. This is especially true when dealing with time sensitive statistics.

Chapter 4

Development Bill of Materials (BOM) & Budget

The cost of the system is relatively high because not only is this an area of ongoing research but it also requires expensive new hardware and software platforms. However, the hardware costs of this system are not unreasonable given the cost of high end CCTV systems that may include high definition IP Cameras that can cost over \$1,000 a piece. The proposed development budget for the project is shown in Table 4.1. The budget provides an itemized breakdown of all of the system materials required for the initial proof of concept implementation. The selected IP cameras were chosen because they are used in many current CCTV systems and support the ONVIF standard. Altera Corporation has generously donated two Cyclone V SoC Development Kits and SoC Embedded Design Suites as well as three copies of Quartus II Subscription Edition. Several Microsoft products are being licensed through Microsoft's standard academic initiative, DreamSpark.

The development system only included three cameras but could likely support many more. The largest capital expenditures for the system are the servers. Thanks to the system architecture, much of the image processing is distributed to the FPGA Back-ends, reducing the load on the Identity Engine. While the Identity Engine has not undergone extensive load testing, it is expected to be able to support many cameras in a normal use case. Depending on the traffic density, potentially hundreds of cameras could be handled by one Identity Engine. See chapter 6 for more information on how the Identity Engine performance depends on traffic density. The Database Management System used in this project, Microsoft SQL Server, has a long running track record in industry. It is a scalable, high performance solution [20] that can support a large number of transactions per second [21]. A single SQL Server instance would be expected to support hundreds of cameras. The User Interfaces would each require a computer with a windows operating system, these are expected to exist in currently operational CCTV systems. The Behavioral Analysis Engine could either be distributed and run in the background on the user interface machines or piggy-backed onto the server.

4.1 Real World Cost Estimates and BOM

The costs shown in Table 4.1 represent the theoretical costs of developing the proof of concept system. It does not represent the actual cost of the end system because it includes several components which, while necessary for development, are not needed in a deployed system. These components include the development tools and, to some extent, the development kit which contains many features that were left unused.

Table 4.2 contains an itemized list of the major components that would need to be included in a production version of the FPGA back-end. It is largely based on the Cyclone V SoC Development Kit part list [4] with the unused components of the development kit removed. Overall, each FPGA back-end board is expected to cost approximately \$566.57 which is reasonable given that the price of some “smart cameras” on the market have prices that reach into the thousands of dollars.

Table 4.3 contains the specifications to a server comparable to the one used in the proof of concept system. This server would able to run both the SQL Server instance as well as the Identity Engine. In most installations one server of this type should be sufficient. In high load environments, additional RAM may be required.

Description	Mfr.	Mfr. Part Number	Qty.	Cost / Unit	Cost	Amount Requested from SCU	Donation Expected	Provided by
Quartus II Subscription Edition	Altera	SW-QUARTUS-SE-FIX	3	\$2,995.00	\$8,985.00	0	\$8,985.00	Altera
Cyclone V SoC Development Kit and SoC Embedded Design Suite	Altera	DK-DEV-5CSXC6NES	2	\$1,595.00	\$3,190.00	0	\$3,190.00	Altera
ONVIF Compliant Fixed IP Camera	D-Link	DCS-2310L	2	\$297.99	\$595.98	\$595.98	0	
ONVIF Compliant 720P PTZ IP Camera	Panasonic	WV-ST165	1	\$488.99	\$488.99	\$488.99	0	
Seagate Backup Plus 2 TB USB 3.0 Desktop External Hard Drive	Seagate	STCA2000100	1	\$89.99	\$89.99	\$89.99	0	
Seagate Barracuda 1 TB HDD SATA Internal Bare Drive	Seagate	ST1000DM003	3	\$67.19	\$201.57	\$201.57	0	
Misc. Cables / Connectors			1	\$80.00	\$80.00	\$80.00	0	
Microsoft Windows Server 2012 Standard Edition	Microsoft		3	\$882.00	\$2,646.00	0	\$2,646.00	Dreamspark Academic (Microsoft)
Microsoft SQL Server 2012 Standard	Microsoft	1 Core License (Min 4 Per Server)	4	\$1,793.00	\$7,172.00	0	\$7,172.00	Dreamspark Academic (Microsoft)
Microsoft Visual Studio Premium (w/ MSDN)	Microsoft		4	\$6,119.00	\$24,476.00	0	\$24,476.00	Dreamspark Academic (Microsoft)
Team Foundation Service (TFS)	Microsoft	1 License	1	0	0	0	0	Microsoft (Free Service)
				Total	\$47,925.53	\$1,456.53	\$46,469	
Key:								
System Materials								
Development Tools								

Table 4.1: Development BOM - For Development Setup + Development Tools

Description	Manufacturer	Manufacturer Part Number	Vendor	Vendor Part Number	Qty	Cost/Unit	Cost
Cyclone V SoC FPGA	Altera	5CSXFC6 D6F31C7N	Digikey	5CSXFC6D 6F31C7N-ND	1	\$283.61	\$283.61
MAX V CPLD System Controller	Altera	5M2210Z F256I5N	Digikey	544-2973-ND	1	\$25.90	\$25.90
Programmable LVDS clock	Silicon Labs	570FAB0 00973DG	Silicon Labs	Custom Part	1	\$40.00	\$40.00
50 MHz crystal oscillator	Silicon Labs	510GBA50 M0000BAG	Silicon Labs	Custom Part	1	\$5.00	\$5.00
MagJack 1000BaseT	Bel Fuse	L829-1J1T-43	Digikey	380-1110-ND	1	\$5.75	\$5.75
Real Time Clock	Maxim	DS1339C	Digikey	DS1339C-33#-ND	1	\$6.78	\$6.78
32M 16 8, 1024-MB DDR3 SDRAM	Micron	MT41K256 M16HA-125:E	Digikey	MT41K256M16 HA-125:E-ND	1	\$7.18	\$7.18
256-Mb NOR flash	Altera	EPCQ256SI16N	Digikey	544-2779-ND	1	\$50.00	\$50.00
32-Kb EEPROM	Microchip	24LC32A	Digikey	24LC32A-I/SN-ND	1	\$0.41	\$0.41
Micro SD Card Slot	Wurth	693 071 010 811	Digikey	732-3819-2-ND	1	\$2.10	\$2.10
Power Supply Controller	Linear	LTC2978	Digikey	LTC2978AI UP#PBF-ND	1	\$21.51	\$21.51
Misc Components					1	\$100.00	\$100
PCB (6" x 6" Standard 2 Layer) Low Quantity	Advance Circuits				1	\$18.33	\$18.33
						Total	\$566.57

Table 4.2: FPGA Estimated BOM - Based Largely on Cyclone V SoC Dev Kit Part List [4]

Description	Cost
Colfax ProEdge SXT8600: -Intel Xeon E5-2637 (x2) -16GB DDR3 RAM (4x2048MB each) -120GB SSD -6TB Hard Drive -Nvidia Tesla K20C -Microsoft Windows Server Standard 2008 R2 SP1	\$8,715.49

Table 4.3: Server Estimated BOM - Based on Online Quote from Colfax International [5]

Chapter 5

FPGA Back-end Design and Implementation

5.1 Purpose

Before a description of FPGA Back-end Design and Implementation can be presented, it is important to state the tasks that the FPGA is responsible for performing. The FPGA Back-end is tasked with:

1. Receiving video from IP cameras
2. Processing video in preparation for analysis (ex. Brightness / contrast)
3. Performing object detection
4. Cropping images around objects
5. Performing additional processing on cropped images
6. Sending cropped images to the Identity Engine
7. Continuing to track objects

One of the goals of the FPGA Back-end is to reduce the load on the central Identity Engine. It accomplishes this by distributing image processing. Since each camera is attached to an an FPGA Back-end, multiple FPGA Back-ends can be processing multiple video streams simultaneously. The FPGA Back-end is able to crop images around objects in the frame that must be identified by the identity engine. By performing this step, the bandwidth requirements between the FPGA Back-end and the Identity Engine is reduced along with the load on the Identity Engine since it knows images passed to it have an object in the center. This frees it from having to processes multiple 720P video streams simultaneously to find objects to identify. The decision to separate object detection and object identification was informed by the fact that facial detection and facial recognition are often separated with facial detection being much easier to implement and easier to

perform than facial recognition. By separating object detection and recognition, it allows these components to be fitted with updated versions of their respective algorithms at a later date.

5.2 Scoping

In a production system, the FPGA would use a facial detection algorithm, potentially coupled with additional object tracking. In scoping this project, the overall system architecture was considered to be the primary goal. As such, an artificially designed target was used in place of faces for this proof of concept. The system was designed so that the object detection method used for the target could later be replaced with facial detection IP. The artificial target (shown in Figure 5.1) makes use of a black circle at the center and a black ring around the outside for object detection. The inner two colored rings are used for identification with each permutation of colors representing a different target.



Figure 5.1: Example Target

5.2.1 FPGA Paradigm Shifts

The inclusion of the ARM Processor with the FPGA constitutes a major shift for the FPGA design methodology. While soft core processors have existed for some time, they entirely relied on the FPGA fabric to run and thus were subject to the performance constraints of the FPGA. While FPGAs are fast, they are not as fast as an ASIC. Because the included ARM processor is hard IP with its own clock, it can run at a much higher clock rate than the FPGA. This shifts the focus of the FPGA from being the center of the design to being a resource to be used by the ARM processor. Customs hardware accelerators are one possible application of the FPGA fabric. While this is an exciting new direction for FPGAs, it has required reworking the typical FPGA development process. While the development tools have been adapted to this new process, there is a substantial ease of use barrier and steep learning curve.

5.2.2 Current Progress

The FPGA development kit and the development process that came with it were brand new at the time that the project was being developed. As with any new technology, there were some issues getting this new platform up and running. How to utilize the FPGA from the ARM processor as well as how to streamline the development processes is still under investigation. As such, a complete end to end demo of the FPGA could not be completed by the project deadline. The underlying design of the FPGA Back-end, however, is still relevant and given more development time would likely be completed. The information given in section 5.3 represents the design of the FPGA Back-end while the information in section 5.11 section 5.12 discuss a simulation of the FPGA Back-end in Matlab and the work accomplished on the FPGA respectively.

5.3 Development Platform Overview

5.3.1 Hardware Overview

In order to reduce development time and complications, the FPGA Back-end was built using an existing development kit produced and provided by Altera Corporation. The development kit being used is the *Cyclone V SoC Development Kit and EDS*. It provides all the hardware required for the FPGA Back-end including one Gigabit Ethernet adapter, two 10/100 Ethernet adapters, DDR3 memory, power supply circuitry, and a LCD line display [1]. An image of the development is shown in Figure 5.2. The FPGA featured in the development kit is the 5CSXFC6D6F31C8NES model of the Cyclone V SoC [1]. Specifications for this chip are reproduced in section 16.1 for the reader's convenience. The key differentiator for this FPGA is that it includes a dual-core ARM Cortex-A9 as hard IP alongside the FPGA fabric [1]. Rather than having to instantiate a soft core processor, which takes up valuable LEs (Logic Elements) in the FPGA fabric, the higher performance ARM Cortex-A9 provides all of the functions of a standard embedded processor along with having the ability to manage the configuration of the FPGA fabric and to interface directly with some hard IP such as the Gigabit Ethernet controller. Bridges exist to allow the ARM processor to interact with soft IP configured in the FPGA fabric and vice versa [12].

5.3.2 Software Overview

Development Software

Several different software packages were used in the development of the FPGA Back-end. The main software packages that were used were:

- Quartus II Subscription Edition (Altera Corporation)
- OpenEmbedded Build System - Bitbake (Openembedded.org)

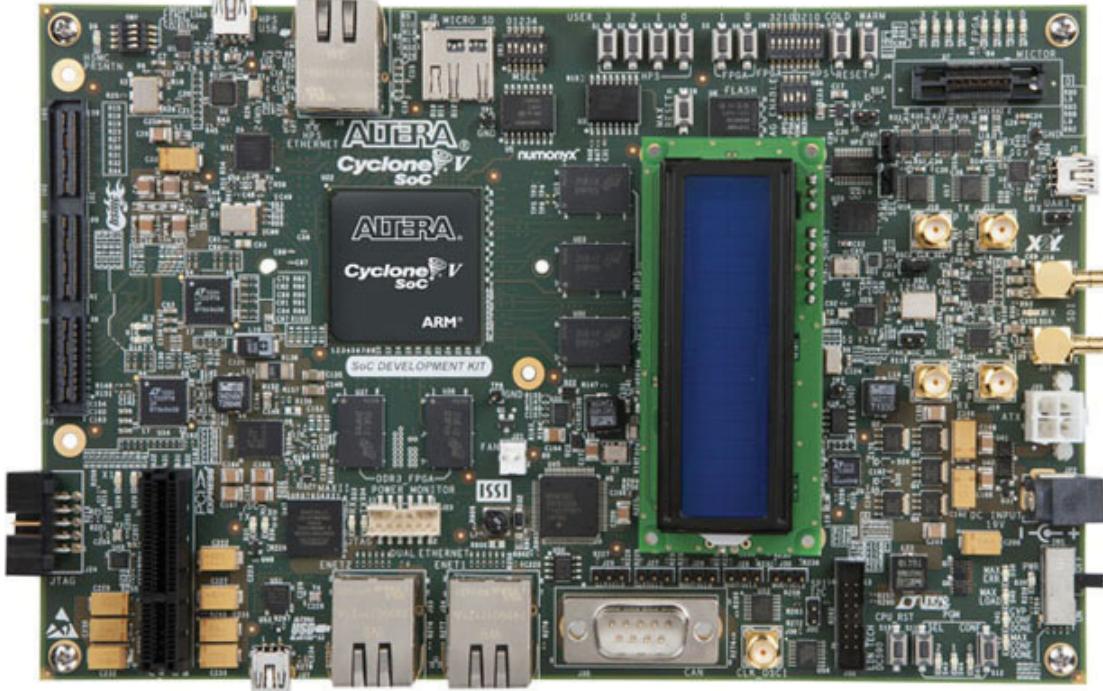


Figure 5.2: Cyclone V SoC Development Kit - Image Courtesy Altera Corporation [1]

Quartus II is part of the Altera suite of development tools for FPGAs. Implementation of the FPGA hardware accelerators used in the project was done in Quartus II and system level integration was done in the companion Qsys tool.

Bitbake, OpenEmbedded's build tool, was used to compile and package a custom Linux distribution based on the Yocto distribution which included the device drivers and libraries necessary to interface with the FPGA as well as our application software.

5.3.3 Embedded Software

Several software packages are used in the FPGA Back-end while it is running:

- Yocto Linux (Linux Foundation)
- LIVE555 (Live Networks, Inc.)
- libjpeg (Independent JPEG Group)

Based on reference designs from Altera [22] it was decided that Linux development would be done instead of bare-metal development. Several factors contributed to this decision including the large amount of software already available for Linux, the ability of Linux to handle low level functions such as the network

stack, and the relative difficulty and complexity of bare-metal development. Also, because of the common use of ARM processors in embedded systems such as smart phones, embedded Linux distributions have already been created which cross compile to the ARM architecture. Yocto Linux is the base distribution used in Altera example systems and is what the FPGA Back-end uses [22]. LIVE555 is a RTP/RTSP streaming library used to open and buffer the RTP/RTSP video streams [23] from the IP cameras. libjpeg is a library [24] used to decompress the MJPEG video received from the cameras. Other open source software packages are used in the custom distribution and are detailed in the *Licenses* directory of the distribution.

While the end-to-end application for the FPGA Back-end was not written due to the complexities discussed in subsection 5.2.2, live555 was tested with a sample application. Live555 was able to successfully open a connection to the camera and start a MJPEG stream. libjpeg was cross compiled based on an existing OE recipe and included with the custom Linux distribution loaded on the FPGA. All indications are that the application software would not take long to develop as most of the work is being conducted by libraries or the FPGA accelerators. It is likely that the end-to-end application could have been developed in a matter of days once the hardware was ready.

5.4 FPGA Back-end Design - Data Flow Perspective

Looking at how data flows through the FPGA Back-end provides a lot of insight into how the FPGA Back-end was designed and implemented. A data flow diagram for the FPGA Back-end is shown in Figure 5.3. This diagram shows how the FPGA Back-end interacts with other components in the system. It uses the SOAP protocol to configure the cameras and the video stream. This service is guaranteed by the ONVIF Profile S standard [9]. The video is streamed to the FPGA Back-end over the RTP/RTSP protocols and is decoded using MJPEG. The FPGA Back-end processes the video and uses sockets to send the cropped images, along with some metadata, to the Identity Engine.

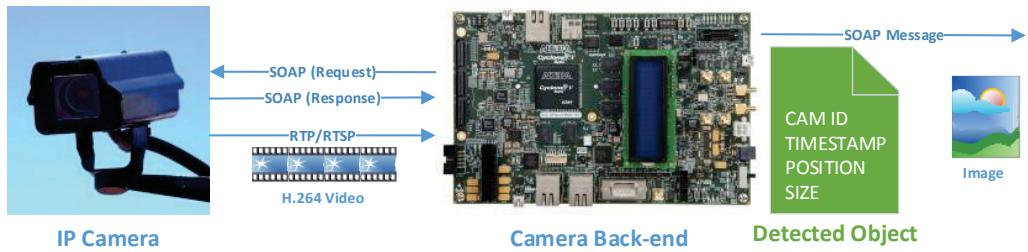


Figure 5.3: FPGA Back-end Data Flow Diagram

5.5 FPGA Back-end Design - Software Perspective

Now that the Data Flow perspective has been presented, the design of the FPGA Back-end software can be detailed. A data flow diagram from the software perspective is presented in Figure 5.4. Embedded Linux handles the network stack, sends and receives SOAP messages, manages/buffers the RTP/RTSP video stream, and decodes the MJPEG video. LIVE555 is used to manage the RTP/RTSP video stream while libjpeg is used to decode the MJPEG video. The image processing takes place in a hardware accelerator implemented in the FPGA. The software provides the decoded images to this hardware accelerator (which appears as a processor peripheral) and reads the results back when they are available. The results are packed up and sent to the Identity Engine via socket connections.

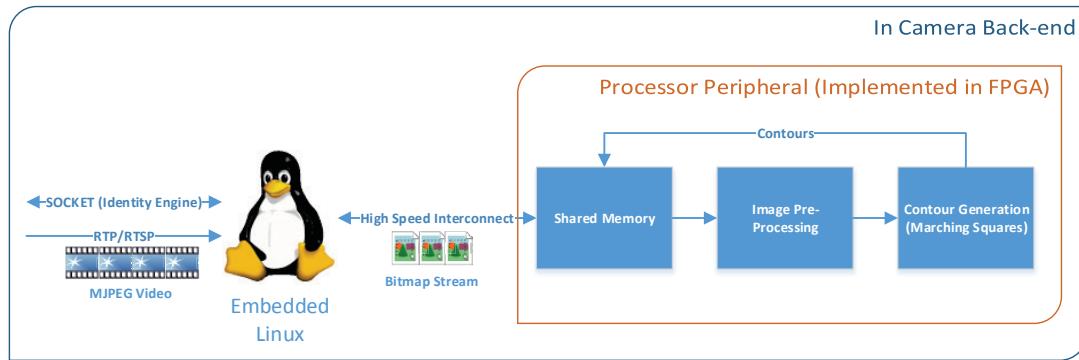


Figure 5.4: FPGA Back-end Software Data Flow Diagram

5.6 FPGA Back-end Design - Hardware Perspective

Now that the software architecture has been presented, the hardware architecture can be detailed. An architectural diagram is shown in Figure 5.5.

The components in the blue region of the diagram are part of the Hard Processor System (HPS). These components have minimal configurability and act just like the hardware peripherals of a standard embedded system. The green region represents the FPGA fabric, which can be reconfigured completely. Three bridges allow the HPS and components in the FPGA to communicate. The HPS2FPGA is a high speed, high bandwidth bus and is used for large data transfers [12]. The LWHPS2FPGA is a lightweight bus used mostly for controlling and checking the status of IP components in the FPGA [12]. The FPGA2HPS bridge provides a way for FPGA IP to access the HPS [12]. The interconnect between IP components in the FPGA fabric uses the proprietary open Avalon standard, created by the Altera Corporation [25]. Specifically, the Avalon-MM (memory mapped) standard is being used. This standard employs a master - slave methodology where mas-

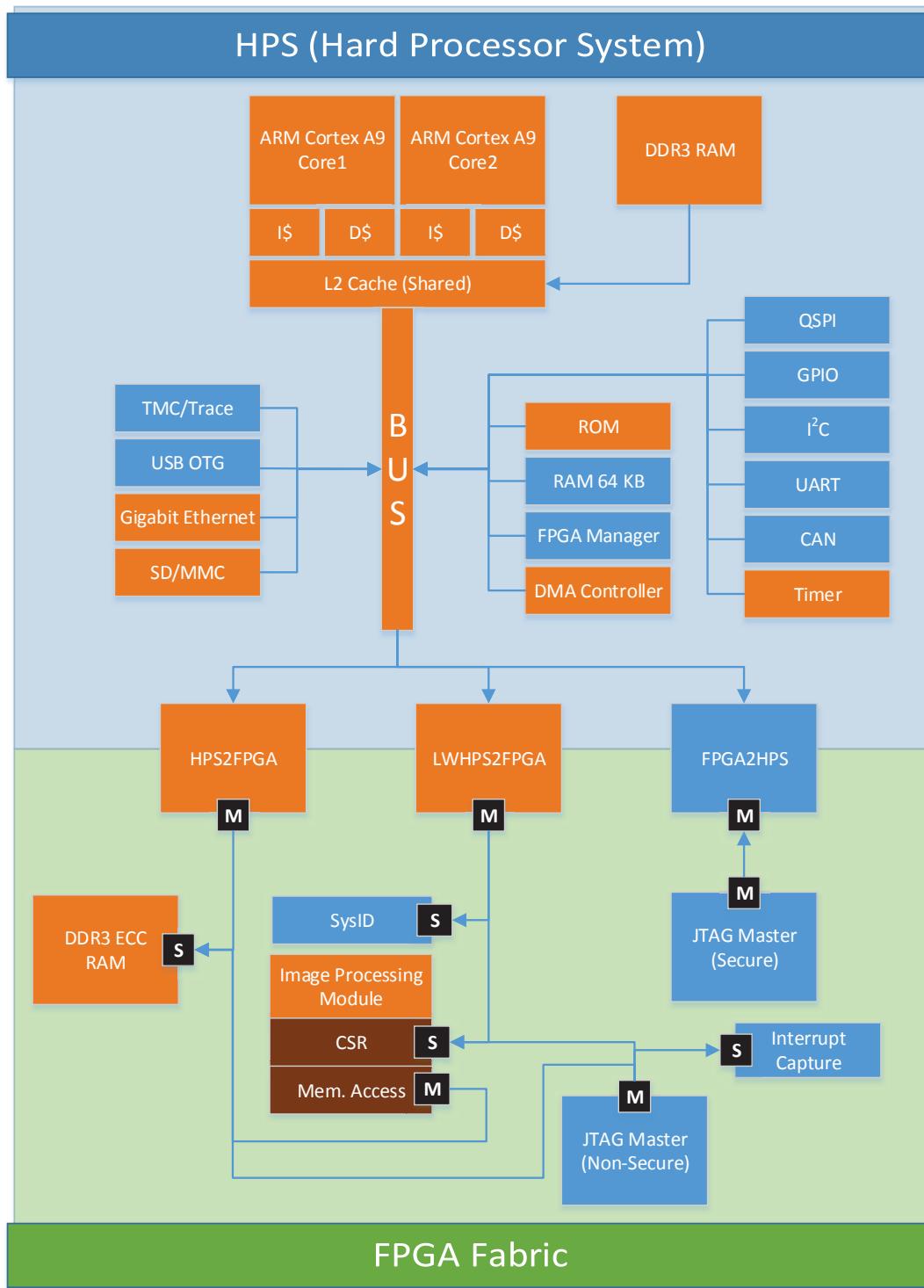
ters can exchange messages with slaves [25]. Each slave is mapped to the address space of the master [25]. Avalon allows many-to-many relationships between masters and slaves [25]. Master and slave interfaces are denoted in Figure 5.5 as black squares with a “M” for “Master” and “S” for “Slave”.

The FPGA Back-end architecture is based on the *GHRD (Golden Hardware System Reference Design)* from Altera [2] [22] [26]. Only certain components are used by the FPGA Back-end and are represented as the orange blocks in Figure 5.5. The ARM processor (along with its memory subsystem) is used to run Linux. The Gigabit Ethernet adapter is used to communicate with the IP cameras and the Identity Engine. The SD/MMC peripheral is used to store the Linux OS image. The ROM is used to store the FPGA configuration. The DMA controller is used to manage memory operations.

The Image Processing Module is the custom hardware accelerator created for the FPGA Back-end. It is a Verilog module and conforms to the Avalon-MM standard. It provides two interfaces: a Control Status Register (CSR) and a Memory Access interface. A DDR3 ECC Memory Controller is instantiated in the FPGA fabric. This is connected to the high bandwidth bridge as well as the Memory Access interface of the Image Processing Module. This memory is used as shared memory between the processor and the Image Processing Module. It is used to store video frames to be processed as well as the cropped output images and metadata from the Image Processing Module. Because the CSR does not have large bandwidth requirements, it is connected to the lightweight bridge.

In order to fully describe the motivation and utilization behind the hardware architecture, the operations the FPGA Back-end performs are presented below:

1. Video data is received and decoded by the ARM processor
2. The decoded raw RGB data is copied into the shared DDR3 memory (via the FPGA memory controller)
3. The address of the image data in the shared memory is set in the CSR of the Image Processing Module to tell it where to find the new image
4. The Image Processing Module processes the image and generates cropped images and metadata
5. The cropped images and metadata are stored in the shared memory
6. The CSR is updated with the address of the output data in shared memory
7. The ARM processor polls the CSR and, when the address of the output data changes, it copies the output data to the HPS memory
8. The ARM processor packages the output data and sends it to the Identity Engine



This diagram is a modified version of the “GHRD System Architecture Diagram” from the *Altera SoC Golden System Reference Design User Guide* by Altera Corporation.

Figure 5.5: FPGA Back-end Hardware Architecture, Adapted from Altera Reference Design Diagram [2]

5.7 Video Format Selection

The MJPEG format was selected as the format for the video being consumed by the FPGA Back-end for several reasons. First and foremost, it is a format that is guaranteed by the ONVIF Profile S standard [9]. Secondly, unlike H.264 or MPEG, each frame of MJPEG video is a complete frame. In H.264 or MPEG, only I-frames are complete frames. Other frames only contain differences from previous frames. The problem is that if a frame is lost or dropped, the video will be corrupted until the next I-frame is received. This can take quite some time depending on how the codec is configured. Since the ARM processor is managing the stream and doing the de-multiplexing and decoding of the video, it may be unable to keep up with the number of frames coming in and be forced to drop some. In this case, MJPEG's complete transmission of each frame is preferable. The difference in format are summarized in Table 5.1.

MJPEG	H.264
Each frame is a complete frame	On I-frames are complete. The rest are diffs from previous frames
Higher bandwidth required	Less bandwidth required
Lossy Compression	

Table 5.1: Video Formats

5.8 Object Detection - Contour Forming

5.8.1 General Method

The targets being detected (see Figure 5.1) all have a black circle in the center and a black ring around the outside. Each of these features has a white boarder. The boundaries between the black features and the white boundaries create clearly defined closed edges. The ratios of the edge diameters are set by the design of the target. Targets can therefore be identified by looking at edges and comparing their center points and diameters. For each target, the edges of the rings will have very similar center-points since the edges are all concentric. Finally, since the edges should be circular, the aspect ratio of the edges (ratio of width to height) should be 1:1. Since concentric edges, especially those with the ratios defied by the target, are uncommon it is not actually necessary to ensure the edges are circular. Checking aspect ratio has worked well in our tests as a way to slim down the number of edges to process without introducing the added complexity of determining if an edge is completely circular. In addition, the diameter of the circular edge is the same as the height or width of the edge's 1:1 bounding box, greatly simplifying calculation of the diameter. This does mean, however, that any series of concentric edges where the ratios of the bounding boxes are equivalent to those of the target will be recognized as a target. This is rare however and could be remedied by the inclusion additional processing step including more sophisticated ways of identifying edges as circles. This would

likely include checking the perimeter of the edge and the area enclosed are consistent with a circle.

The parameters defining the tolerance on edge center-point spacial locality, edge aspect ratio, minimum points in edge, and minimum edge dimensions can all be fine tuned to provide an optimal solutions. Because the focus of this project was to demonstrate the overall architecture, these values were set and tested until a target could be detected reliably. The values selected may not be the optimal values and further testing would likely improve object detection accuracy.

The steps to target identification can be broadly described below:

1. Extract closed edges from image
2. Discard edges that are too small to provide meaningful dimensions (noise in image or very small objects) - used to reduce data set
3. Discard edges with aspect ratios not close to 1:1
4. Group edges with close center-points and put them in “bins”
5. Discard any bin with less than 3 edges (targets have at least 3 edges)
6. In each bin, find the edge for the center ring (should be smallest edge)
7. Compare ratios of each edge in bin against the edge of the center ring and check for defined ratios.
8. If both the inner and outer ring edges are identified, then the collection of edges constitutes a target.
9. Crop image around bounding box of outer ring (will include fill target).

This algorithm could be modified and optimized further. One assumption that was made in this algorithm is that the smallest edge in a bin is the edge around the center circle of the target. Due to noise in the image or some other phenomena, this may not be the case. The algorithm could be made more robust by checking the ratio of edges to each successively larger edge until there are no more edges to compare against or the target is correctly identified. While this would increase the robustness of the algorithm to real world conditions, it would also add complexity and ensure a larger load on the system. Since the artificial targets were created for the proof of concept, and because this algorithms would be replaced by facial detection in a production system, it was decided that the less robust but faster algorithm would be suitable for the initial implementation.

5.8.2 Contour Lines vs. Edges

The algorithms discussed in subsection 5.8.1 assume that there is a way to extract closed edges from an image but does not describe how that operation is performed. There are several algorithms that can be used

for edge detection including the Canny Edge Detection algorithm which is discussed in detail in section 6.2. While Canny Edge Detection is a well researched algorithm, it does not guarantee that the edges it detects are closed. An example of this can be seen in the seminal paper for Canny Edge Detection [27] where the results of running example images through the edge detection algorithm yield some unclosed edges.

Project advisors Professor Sally Wood suggested that, since the targets had clearly defined, closed edges along the border of the black circle and the black ring (see Figure 5.1) that contour lines be considered instead of edge detection.

Contour lines and edges of an image appear very similar. However, contour lines denote the boundaries of the portions of the image that are below a certain intensity threshold and the segments of the image that are above that threshold. A good way to think about this is to visualize the image like a mountain range with different intensity values being different elevations. The contour lines are the different elevation lines on the mountain. In fact one of the most common applications of contour lines is in topographic maps as elevation lines.

5.9 Marching Squares

There are several method that can be employed to extract the contour lines of an image. One of the best known algorithms is commonly known as the “Marching Square” algorithm. The Marching Square Algorithm was never officially published but is the 2D case of the Marching Cubes Algorithm [28]. The seminal paper on the Marching Cubes algorithm, published in 1987 by William E. Lorensen and Harvey E. Cline [29], is referenced by most papers discussing the Marching Squares Algorithm. The 2D Marching Dquares case is considered a natural extension of the Marching Cubes Algorithm and is considered by many image processing insiders as common knowledge. As such, it is often not explained in depth in papers. However, it is explained in the paper *Marching Cube Algorithm: Review and Trilinear Interpolation Adaptation for Image-Based Dosimetric Models* by D.A. Rajon and W.E. Bolch [30].

The marching square algorithm consists of the following steps [30]:

1. Select a threshold value and compare each pixel in the image against the threshold.
2. Compare 2x2 pixel blocks (with no overlap) against a look up table to determine the contour segments in that block
3. Perform linear interpolation of the contour segments along the boarder of the 2x2 pixel blocks to obtain a more accurate contour

An illustrations of the first two steps of the algorithm are shown below. While the linear interpola-

tion is important to form very accurate contours, it is not required, strictly speaking. In order to simplify the implementation of the algorithm and the explanation here, the linear interpolation step will be omitted.

Figure 5.6 shows the look up table used by the marching square algorithm. The grey box represents a pixel that is above the threshold and a white box represents a pixel that is below the threshold. The red line represents a contour segment. Note that there are two cases that have both red and blue lines. These are ambiguous cases as either the red lines or the blue lines could be the contour segments for those cases [30].

Figure 5.7 shows the original image while Figure 5.8 shows the image after it is compared against the threshold value. Figure 5.9 shows the result of the image after each 2x2 block is compared against the look up table. The red lines represent the contours. Note that in order to know the direction of the contour at the edge, a 1 pixel overlap with an adjacent segment of the image is required. This is represented by the green line.

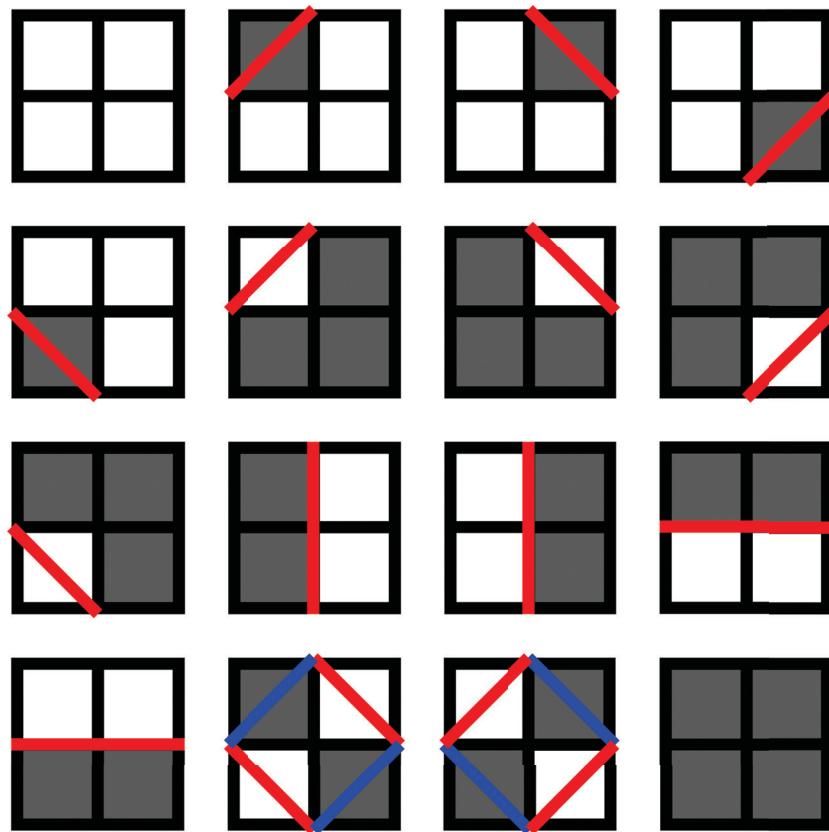


Figure 5.6: Marching Square Look Up Table

FF																
FF																
FF	FF	FF	FF	FF	95	60	60	60	95	FF						
FF	FF	FF	FF	95	60	00	00	00	60	95	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	60	00	00	00	00	00	60	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	60	00	00	00	00	00	60	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	60	00	00	00	00	00	60	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	95	60	00	00	00	60	95	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	95	60	60	60	95	FF						
FF																
00	FF	00														
00	00	FF	00	00												
30	00	30	FF	30	00	30										
FF	00	00	00	FF	00	00	00	FF								
FF	FF	30	00	00	FF	FF	FF	FF	FF	00	00	30	FF	FF		

Figure 5.7: Original Image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1
0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0
0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0

Figure 5.8: Image After Comparison to Threshold 0x45

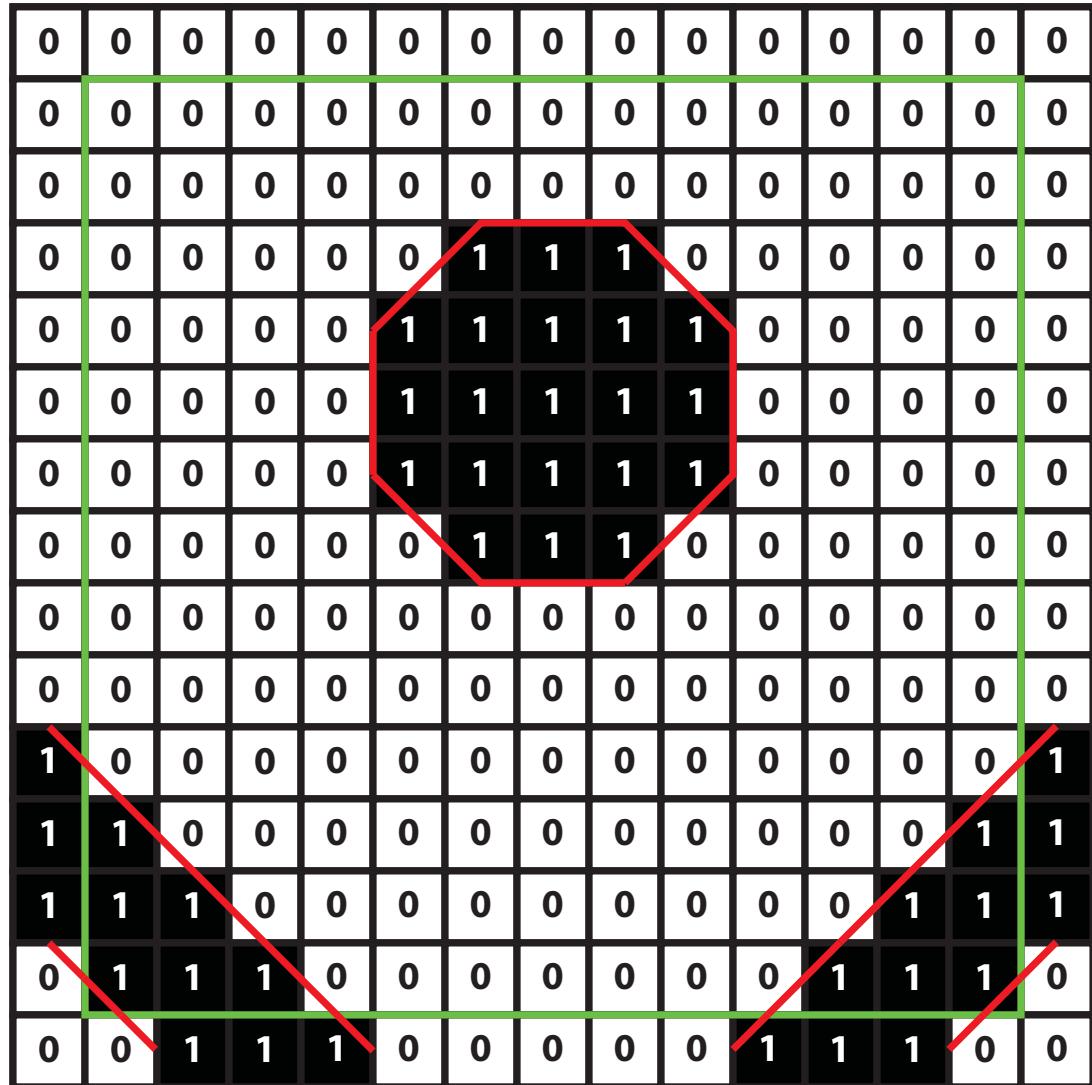


Figure 5.9: Image After Evaluating 2x2 Pixel Blocks for Contour Segments

5.10 Marching Squares Design

The marching square algorithm as described in section 5.9 is clearly parameterizable. However, it is not clear what the best parallel implementation of the algorithm is. Further more, the above description of the algorithm could be easily implemented by a processor based solution but the mapping of the solution to a FPGA platform is more nuanced. FPGAs do not behave in the same way as processors; rather than executing a set of instructions, FPGAs are essentially configurable logic networks. Multiple datapaths can exist and be processed completely in parallel. In an attempt to exploit the parallelism of the FPGA platform an alternate implementation of the marching squares algorithm was drawn up.

It should be noted that FPGAs have a limited number of configurable logic elements. Developing a design to process a full uncompressed HD image from the cameras would require more resources than the FPGA has. In order to avoid this issue, the image is segmented into blocks to be processed. Note that a 1 pixel overlap on each side is required to determine the shape of the contour at the edge of the block

The initial step of the marching square algorithm remains largely the same: individual pixel intensity values are compared against a threshold. The FPGA implementation does this operation for each pixel in parallel.

The next stage of the process is carried out by a network of processing units. Figure 5.10 shows the processing modules in blue and the connections in green. The white and black squares in between the green connections represent the pixels. Each processing unit has access to a 2x2 pixel segment of the image. Each processing unit is directly connected to the unit above, below, to the left, and to the right. When the threshold compared image block is ready, each processing unit looks at the values of the four pixels it is connected to. Each processing unit is an implementation of the look up table described in Figure 5.6. The processing units configure their external connections based on the look up table. This state is shown in Figure 5.11 with the red lines representing the mapping between external interfaces. Each processing unit also has two memory units (since there can be a maximum of two contours per computing unit). This memory unit is used to hold the min x, max x, min y, and max y values for each contour passing through the processing unit. *Note that, since contours cannot overlap or cross, the four tuple should be unique to each contour in the block.* Each processing unit knows its position within the network and is therefore able to set the initial maximum and minimum values for each contour segment. The state of the processing units after this initial setup is shown in Figure 5.11. The orange compute units represent units which have had a maximum or minimum change since the last clock cycle.

At this point, each compute unit believes that maximum and minimum coordinate of the contours that pass through it are the coordinates it initially assigned; this is of course incorrect. The next stages of the algorithm

involve propagating the maximum and minimum values throughout each contour. Propagation of maximum and minimum values occurs on each clock cycle by each compute unit sending its current maximum and minimum values to its connected neighbors that are part of the given contour (determined by the look up table in the setup phase). The neighbor compares their minimum and maximum values to the values received along the connection and update their maximum and minimum values accordingly. Figure 5.12 shows this process for one compute unit during the first round of the algorithm. A flag is set for each compute unit signifying if its min/max registers changed in the last cycle. The processes is completed when a clock cycle has passed and no min/max register has changed. At this point, the system is said to have “settled”. Figure 5.13 shows the system at this final stage. It should be noted that the propagation occurs in parallel for each contour but the total time depends on the contour with the longest settling time. The best case scenario is for a contour which is closed in the block. In this case, it takes 1/2 the length of the contour cycles for the min/max values to fully propagate. In the worst case scenario, when the contour segment crosses over the boundary of the image block being processed, it takes the length of the contour segment cycles for the min/max values to settle.

At this point the complete contours must be put into a set. This process is more difficult to conduct in parallel and the implementation proposed here involves visiting each compute unit to identify whether or not it contains a contour and if that contour has already been included in the set. There is likely a more efficient and elegant way of addressing this last stage of the algorithm but additional investigation would be required to find it.

There are two proposed methods for dealing with contour segments which enter and leave the image block being processed. One method involves creating a list of contour segments that are then “stitched” back together by the processor. An alternative method is to use memory units to store the coordinates of the contour segments as they enter or leave the bottom or right edge of the image block. The saved values are then used as inputs for later runs of the algorithm on the neighboring image blocks. This requires that the image blocks be evaluated in order since it introduces data dependency. However, it does provide the benefit of ignoring contour segments until they form a closed loop. Figure 5.10, Figure 5.11, and Figure 5.13 all show the memory hardware required for this method and how they are connected to the compute units.

5.11 Object Detection Simulation

The suitability of contour lines to perform the object detection detailed in subsection 5.8.1 was tested using a simulation in MATLAB. The `contour()` function was used in place of the FPGA marching square implementation to get the contours in the image. Video recorded from one of the IP cameras was converted to

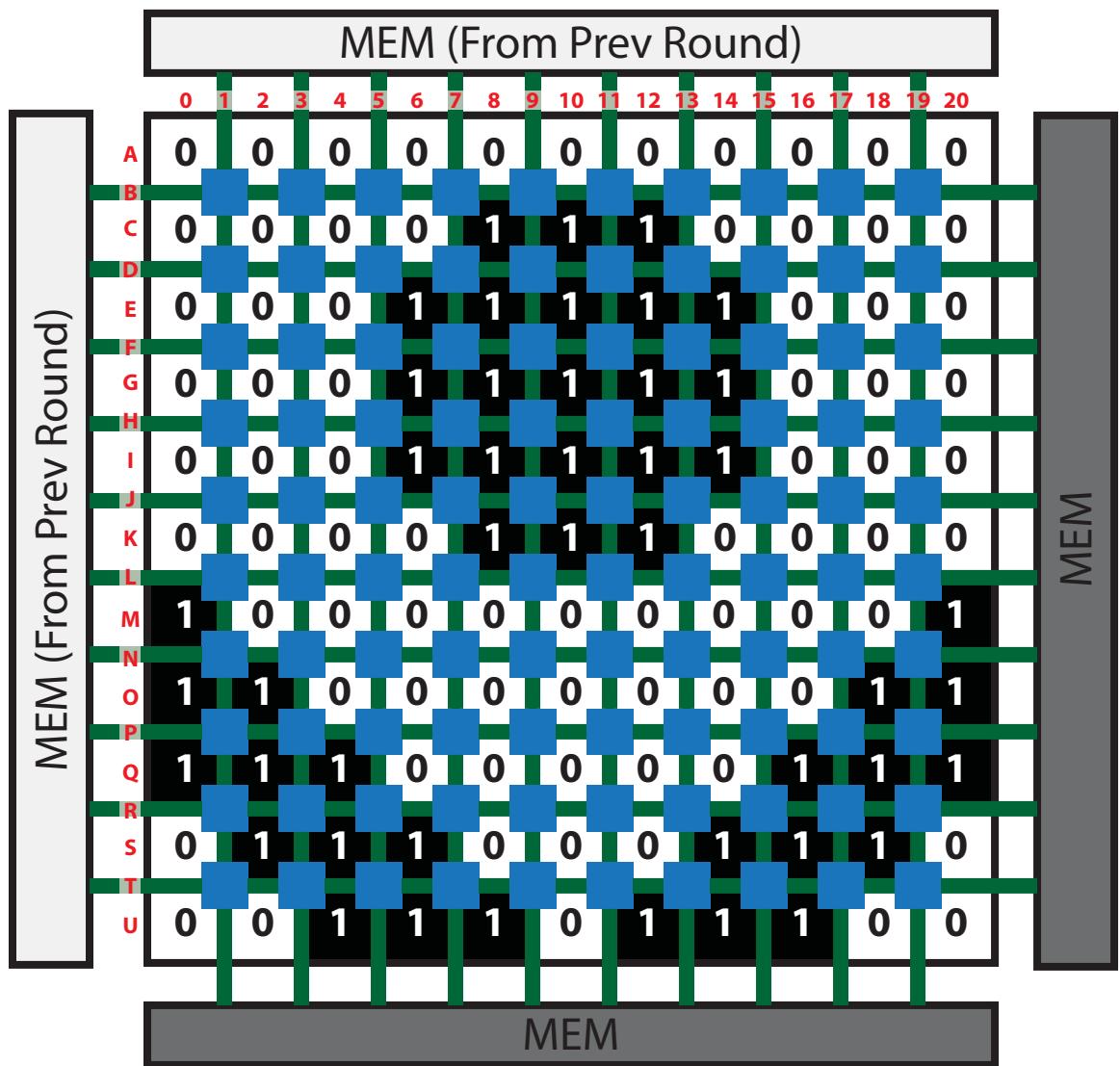


Figure 5.10: Layout Processing Units and Connections Between Them

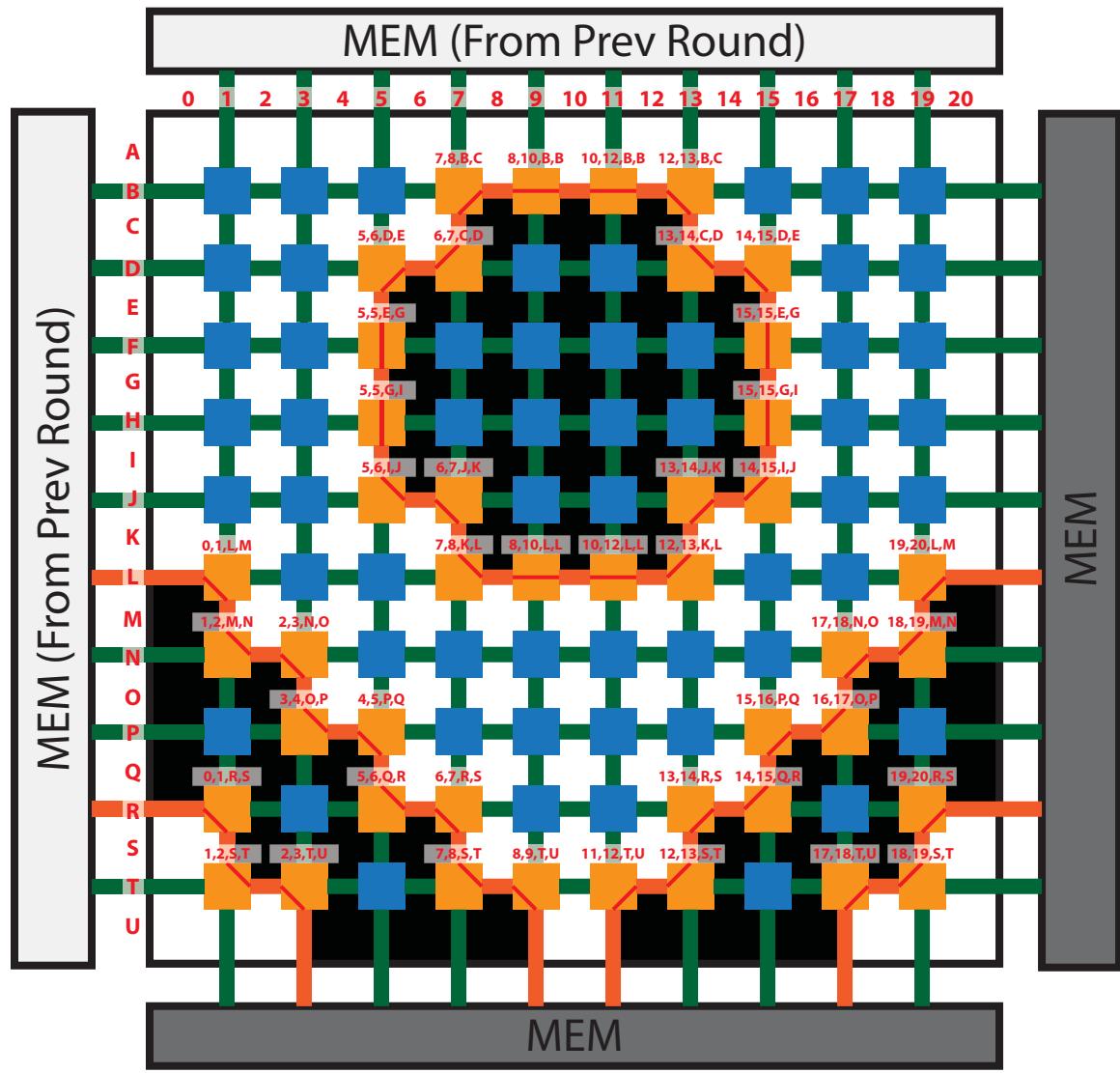


Figure 5.11: Processing Units After Setup

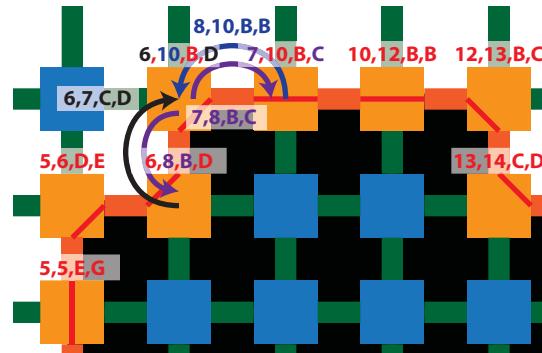


Figure 5.12: Processing Units During Step 1

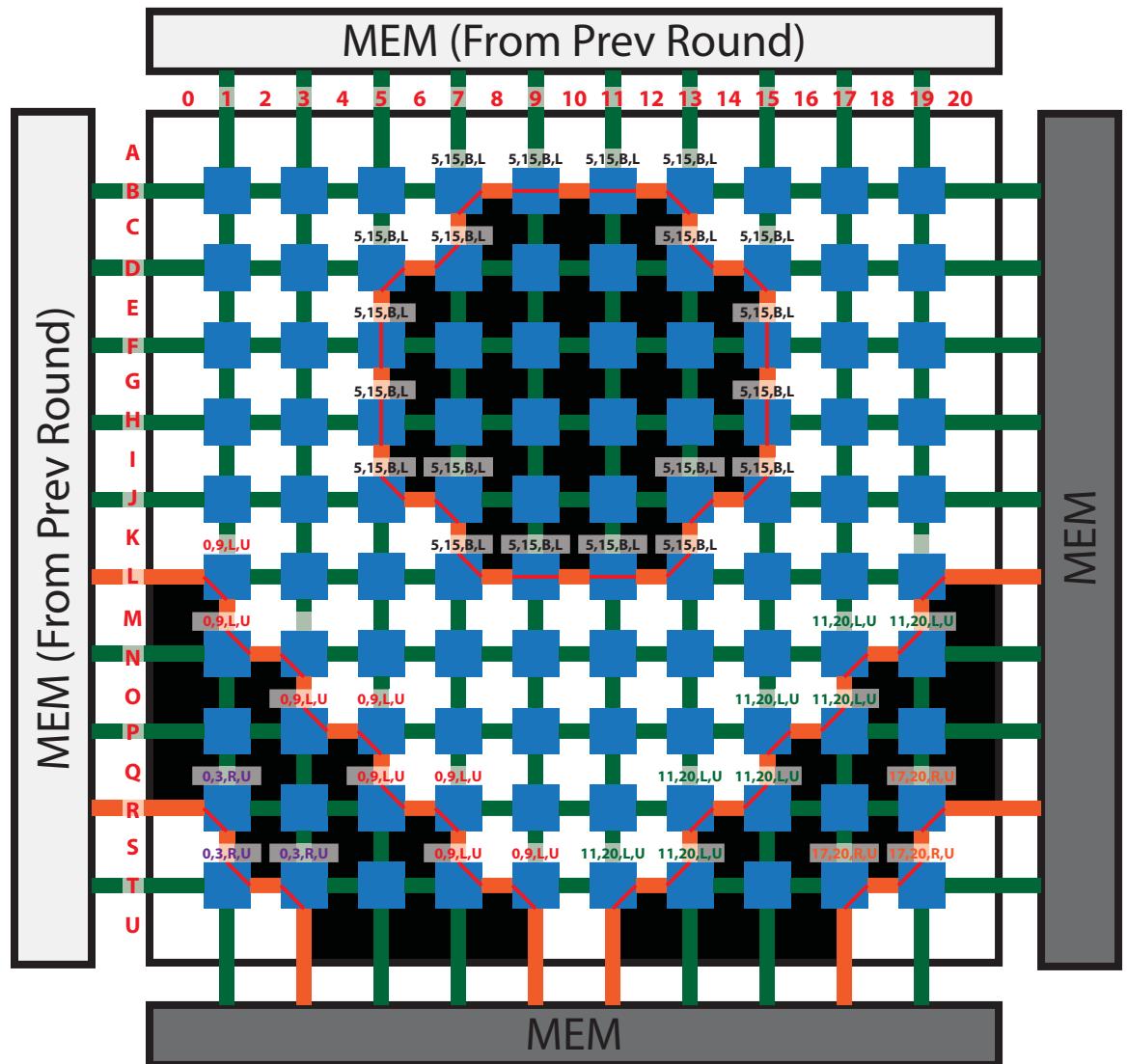


Figure 5.13: Processing Units After Final Step

images and read into MATLAB. The MATLAB code for this simulation can be seen in section 16.2. The result of the simulation is shown in Figure 5.14. The result of the simulation for the distant target is shown in more detail in Figure 5.15. The images shown were part of a preliminary test to determine the distance the targets could be from the camera while still being reliably detected. In the initial test, it was determined that the target detection became unreliable as the target got to the end of the image processing lab. Increasing the tolerance of the diameter ratios to 1.25 allowed the algorithm to be more reliable when the targets were farther away. Additional fine tuning of parameters would likely improve reliability further. The issues with distant targets can begin to be seen in Figure 5.15. As the objects get father away, the number of pixels which can represent the rings decreases. This problem is compounded by the fact that the camera use lossy compression schemes that result in compression artifacts. These artifices can be seen in Figure 5.15 with some irrelevant contours appearing inside the black ring.

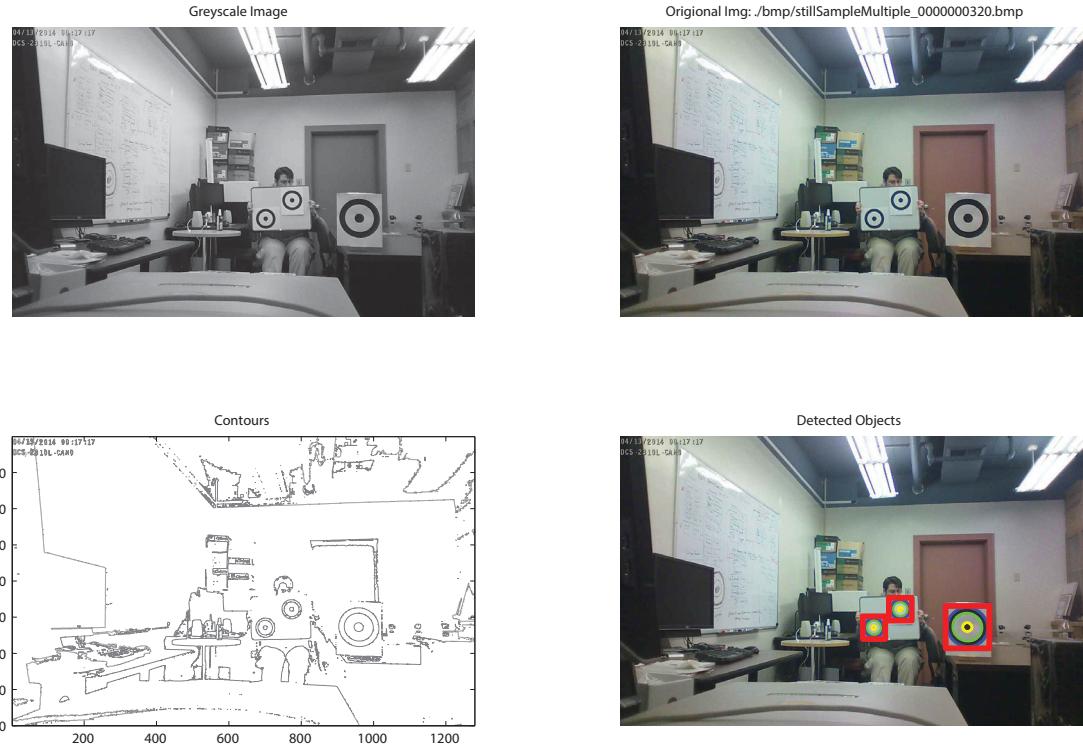


Figure 5.14: Simulation of Contour Based Object Detection

5.12 FPGA Back-end Dataflow Proof of Concept

Due to time limitations, the FPGA implementation of the marching squares algorithm detailed in section 5.10 could not be implemented by the time this paper was written. Instead, a demonstration of the accelerator work flow was developed to show that the general architecture was feasible. The system has the same structure

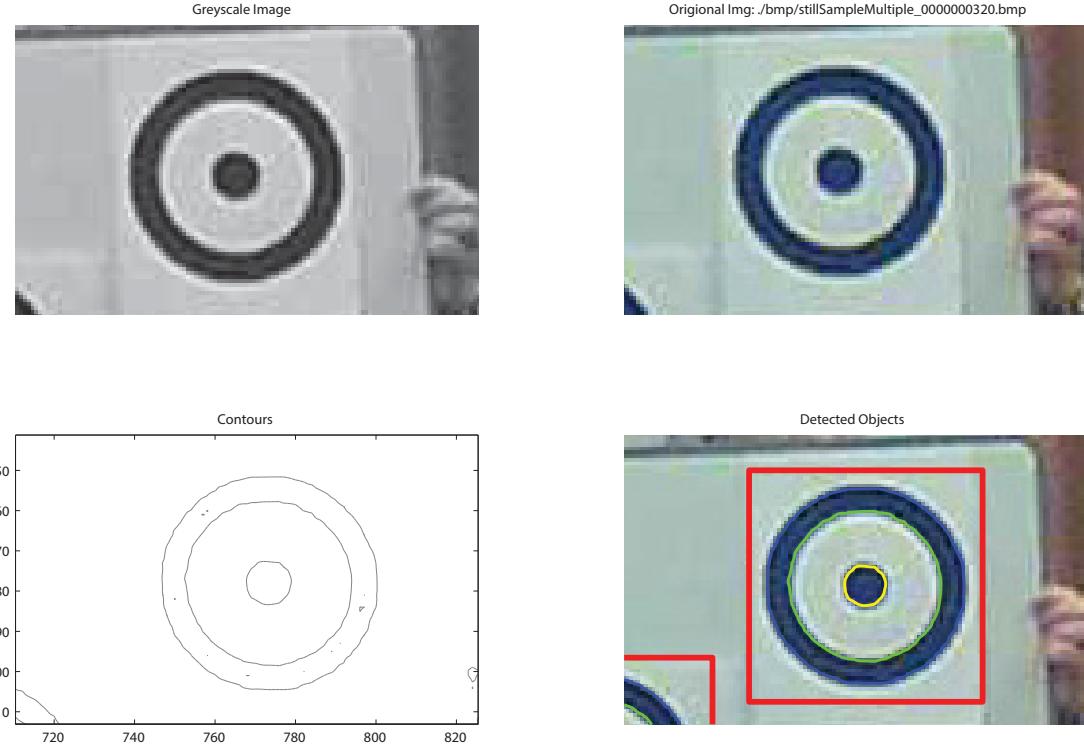


Figure 5.15: Simulation of Contour Based Object Detection - Zoomed

described in Figure 5.3 and Figure 5.5. This demonstration IP took the place of the “Image Processing IP” and demonstrated the ability of the ARM processor and the FPGA IP to communicate via the Control Status Register (CSR) and the shared DDR3 memory. The tasks performed by this demonstration IP are:

1. ARM writes data into shared memory
2. ARM commands the IP (via the CSR) to read the values stored in shared memory
3. IP returns data via the CSR
4. ARM checks data is the same
5. ARM commands the IP (via the CSR) to write data into shared memory
6. IP writes data passed in through CSR into shared memory
7. ARM reads data in shared memory
8. ARM checks data is the same

The hardware implementation was based on many Altera reference designs including the *Golden Hardware Reference Design (GHRD)* [2] [26], the Golden Top top level file [31], the Board Test System (BTS)

[32], the Avalon-MM Master Templates [33], and the Avalon-MM Slave Template `avalonMMMasterTemp`. The software implementation was based on the Golden System Reference Design [22] which is based on the Yocto Linux distribution and the OpenEmbedded build system.

The GHRD formed the basis for the Qsys system design while the BTS formed the bases for the DDR3 memory controller implementation. The demonstration IP relied on both the master and slave Avalon-MM templates.

Due to some technical difficulties, the demonstration is not yet running completely. A problem may exist in the IP, the driver software, or the DDR3 memory controller. Work is currently underway to debug this issue and bring the demonstration online.

The code and design for the demonstration are given in section 16.3 and section 16.4. This includes both code written by the team, Altera reference code, and modified versions of Altera reference code. The development environment was setup according to the instructions at the RocketBoards.org wiki [34]. The RocketBoards wiki was referenced by Altera as the source for documentation on running Linux on the development kit [35]. Several modifications to the distribution had to be performed to get it running correctly. Since an in depth look at the environment setup is not the focus of this paper, the step by step instructions are not given here. Additionally, the development process used is more nuanced than can be effectively described in this document. Instructions were collected in the FPGA lab notebook as the project progressed. This lab notebook is appended to this document in chapter 17.

Chapter 6

Identity Engine Design and Implementation

The primary objective of the Identity Engine is to identify detected objects so that they can be cataloged in the database. The time of first detection and the camera that detected it are stored in the database along with object attributes. Once an object has been logged in the database, each subsequent identification will add an additional detection time and detecting camera. These data points are used by the Behavior Analysis Engine to generate normal behavior patterns and recognize anomalies.

6.1 Software Architecture

The Identity Engine uses a three stage software architecture. In the first stage, the identity engine accepts incoming data through a socket connection. Since the server is Windows based, we used a built-in function called Winsock. Winsock provides the ability to communicate via network protocols, such as TCP/IP or IPX/SPX. Once the Identity receives image data from the FPGA via the socket, image processing can begin.



Figure 6.1: Sample Image as Received From FPGA

Image processing starts with determining the size, in pixels, of the cropped image received. After the height and width is determined, the Identity Engine can properly traverse the image array. This is important because Canny Edge detection works on a pixel by pixel basis. Once Canny Edge detection has been performed, the target boundaries are determined.

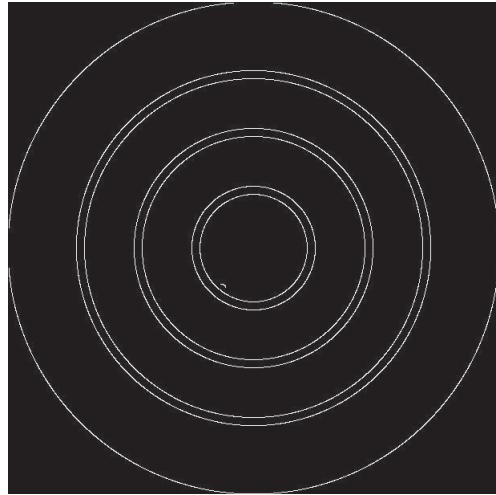


Figure 6.2: Image After Canny Edge Detection

As Figure 6.2 shows, boundaries are marked by binary ones. For the sake of demonstration, the ones are displayed as white pixels. The rest of the image is filled with binary zeroes, which are shown in the example as black pixels. This makes determining the bounding region of the object very simple. In the case of the circular demonstration targets, the algorithm finds the upper, lower, left, and right maxima of the circle. The difference between opposite sides is used to determine speculated radii. Both the speculated radii are averaged to find the radius of the outer bounding circle. Since we are using targets with a known pattern, inner rings are calculated using a ratio to the outer radius.

After establishing the boundaries of the targets inner rings, the Identity Engine works to attempt to identify it. Our target identification criteria are the color of the two ID Bands. In Figure 6.1, the ID Bands are green and red, respectively. To determine the color of each band, the Identity Engine normalizes the red, green, and blue value for each pixel within the established boundaries of each band. Once the normalized colors have been ascertained, they are used to query the database to check for prior identification of this individual. If an individual matches the description, the location and time of this detection are appended to the existing entry. Otherwise a new entry will be created for later reference. To communicate with the database, the Identity Engine uses a tool called Language-Integrated Query, abbreviated LINQ. LINQ is a Microsoft product which allows Windows applications to query SQL databases, ADO.NET Datasets and XML documents. This tool was selected since the Identity Engine is a Windows application and our database is based on Microsoft SQL.

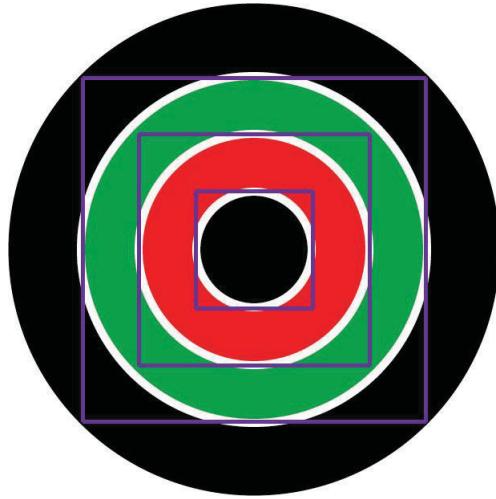


Figure 6.3: Illustration of Outer Maxima and Ratio of Radii (Highlighted in Purple)

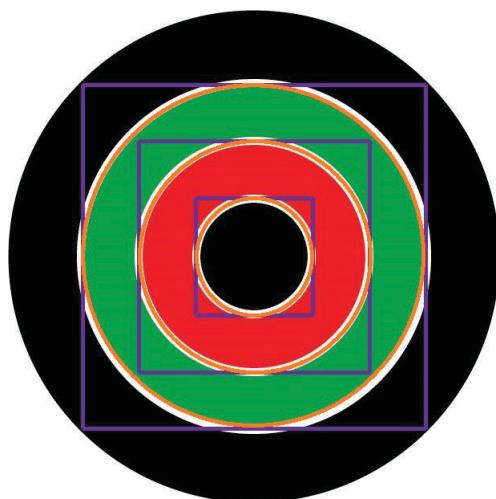


Figure 6.4: Bounding Circles Highlighted in Orange

6.2 Canny Edge Detection

The basis of Canny Edge Detection is three principals: accuracy in finding edges, the detected edge should be as close as possible to the true edge, and the detected edge must only be one pixel wide[36]. It achieves this by using the first derivative of Gaussian functions. The particular implementation used for this project, and the MATLAB implementation, use the thresholds found with these derivatives to generate a binary edge map. A binary edge map is an array of pixels where edge pixels are stored as ones or zeroes, and the rest of the image is the inverse. Which convention is used is a matter of the programmers personal preference. Our implementation used edges marked by ones with the remaining pixels as zeroes to maintain consistency between the final version and MATLAB prototype.

We selected Canny Edge Detection for multiple reasons. It is more likely to produce closed contours than the Sobel method. Perhaps more important, due to our heavy reliance on GPUs, is that Canny is highly parallelizable. Gaussian filters are applied on small groups of pixels. CUDA allows us the ability to apply these operators to multiple groups at the same time. This reduces the overall time required to identify an object.

6.3 Facial Recognition

For the purposes of this proof of concept we decided to forgo facial recognition. Facial recognition is a rich area of research. Simple target recognition provides enough data to verify the function of the Behavioral Analysis Engine. It should be noted that the term recognition encapsulates what we separately refer to here as detection and identification. We refer to the components individually due to the distributed nature of our system.

There are many different algorithms and implementations of facial detection. For the purposes of brevity, I will only discuss one implementation. DeepFace is an algorithm developed and used by Facebook. I have selected to discuss this algorithm for two reasons. First and foremost, it is arguably the most used implementation, since there are millions of users on Facebook at any one point in time. Second, it is purported to be the most accurate algorithm available. In their Conference on Computer Vision and Pattern Recognition paper, Facebook sighted 97.35% accuracy in matching a face between two images. This exceeded the previously most accurate algorithm by 27% [37].

DeepFace relies on a computer science theory called Deep Learning. Deep Learning is a highly complex and rich field of research, and there have already been many publications discussing it in depth. For the purposes of this brief overview, only how DeepFace leverages Deep Learning will be discussed. To identify unknown faces, DeepFace first aligns the facial features to the cameras. This is done by first registering

facial features in two dimensions. The registered features are then mapped onto a generic three dimensional facial model. Once the 3D representation of the unknown face has been generated, it is rotated so that it is aligned with the camera. The resulting image then goes through a series of filters and is normalized. After the post-processing, facial features are used to query the Deep Network of known faces. If the features match a known face, an identity is ascribed to it.

One of the goals of this project is to be modular and allow for integration of future technology. The current implementation of the Identity Engine is designed to be easily replaced with DeepFace or similar system. The only modifications that would need to be made to other sections of the pipeline would be modification of the database schema. All other aspects of the pipeline could be used without modification.

6.4 Testing

To verify the correctness of the overall algorithm, a prototype was first coded in MATLAB. This was due to the pre-integrated availability of all necessary functions in the MATLAB environment. These functions include Canny Edge Detection, sockets, SQL insert and query, and image output for debugging. Once the results were verified to be correct, the algorithm was translated to CUDA C/C++ and Windows functions. Afterward translation results were checked against MATLAB and the expected output.

Chapter 7

Database

The database is an essential back-end component for the system because it provides not only a place to store the vast quantities of data that the system will be generating, but it also provides a connection and method of communication between components of the project. The database interacts with the Identity Engine by receiving new images, identities and events of objects moving within the camera frames, as well as providing images and identities to compare against during object recognition. The behavioral analysis engine then pulls the new information on events and individual and process them, creating anomalies. Finally the user interface is consistently polling the database for new information on events and anomalies to display to the monitoring staff.

7.1 Setup

The database was set up on the server using Microsoft SQL Server Management Studio. Since this is a proof on concept project, many of the standard database components were ignored. These ignore components include audit support, database permissions, access levels, and security.

7.2 Schema

The schema was the primary area of development for the database. Since the database was used as a method of communication for many components of the system, the schema had to match the changing requirements of each component. Originally, the schema was designed before the implementation of many components of the project; this was done to create some structure in communication. As seen in Figure 7.1, the approach to the initial design overly complicated to satisfy all possible situations. However as components of the project began to implement the database connections, it was easy to realize change was needed in the design.

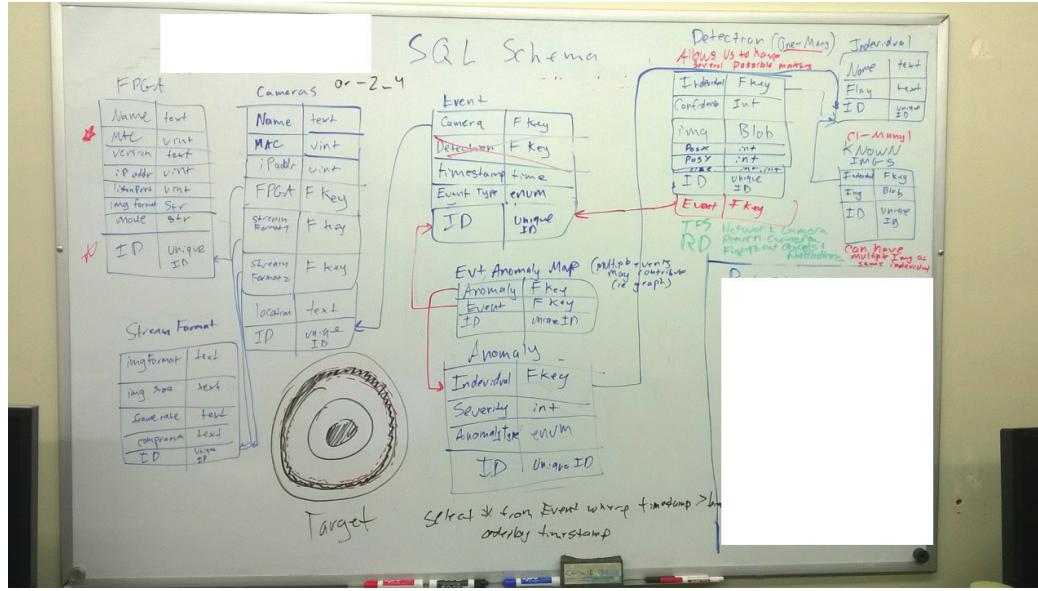


Figure 7.1: Original Draft of the Database Schema

7.2.1 Schema Modifications

The schema was modified multiple times during the implementation of the project and is likely to continue to change even after this thesis has been completed. The first major modification to the schema was the removal of the FPGA and Stream Format Tables. Since the FPGA will not be communicating with the database by the end of the project this information will not be used. In the situation that this system becomes a commercial product, or the FPGA had implemented a database connection for setup, these tables would be re-added in order to provide that information. The next major schema changes were due to the changing of behavioral analysis algorithms. The detection and EventAnomlyMap Tables were removed to simplify how the tables were being processed by the behavioral analysis engine. Instead the schema migrated to the Camera and DetectionEvent table structure shown in Figure 7.2, where only one event would be created for an individual entering and exiting the cameras frame. Finally the CameraStat table was added to easily save and load the statistical information from the database to the behavioral analysis engine.

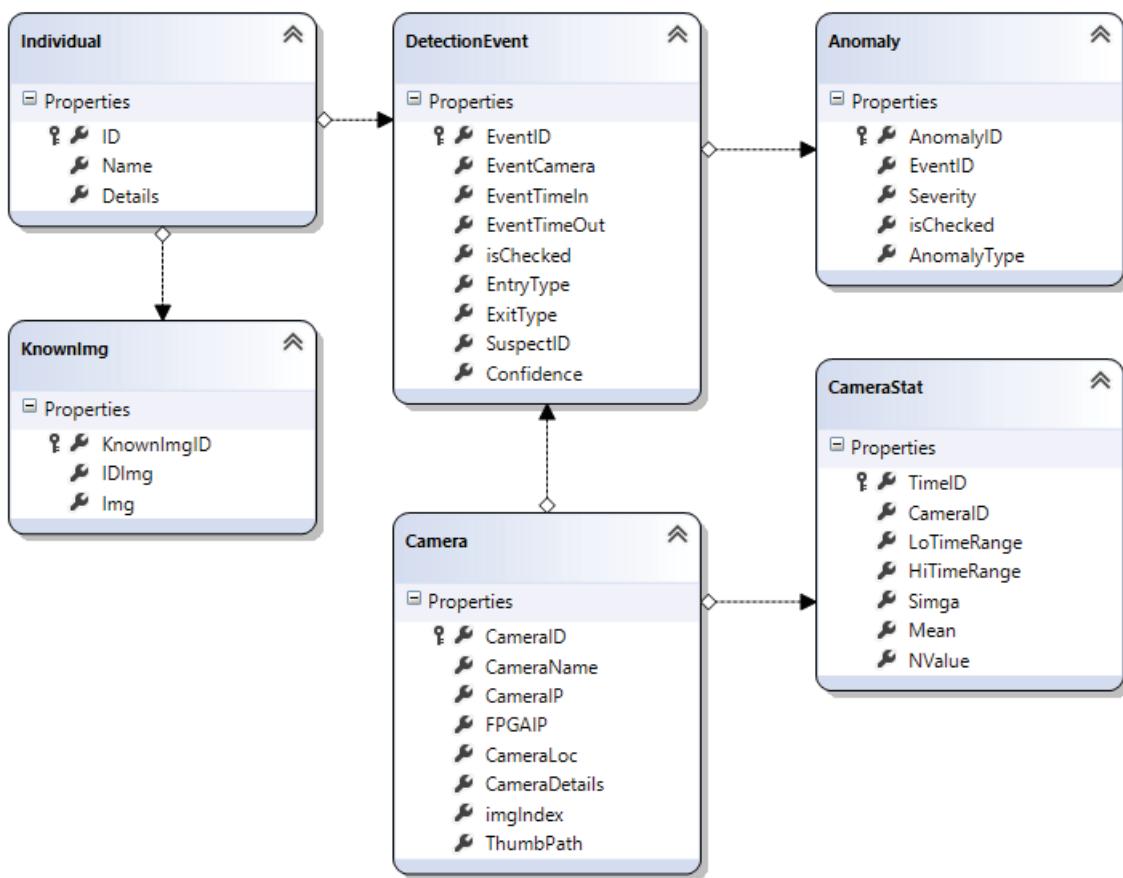


Figure 7.2: Final Schema Design

Chapter 8

Behavioral Analysis Engine Design and Implementation

The Behavioral Analysis Engine is an essential component to the system it because provides real time analysis of the data being produced by the other components of the project. During the conceptual creation of the project many ideas were produced about how to actually detect behavioral anomalies, which were distilled down to two major types of analysis. The original idea behind the behavioral detection of the project was the use of Directed Weighted Graph Analysis, but due to complications and constraints it could not be feasibly implemented. This limitation was due to many factors including time, budget and available technology. Instead Frame Duration Analysis, subset of the original idea, was implemented.

8.1 Weighted Graph Analysis

When the concept of this massive project was first presented it was primarily paired with using Directed Weighted Graph Analysis for behavioral detection. This behavioral analysis algorithm planned to take advantage of a multitude of information generated from the system, including providing analysis by time, location, pattern and identity.

Graphs are a popularly accepted mathematical idea that stems from what is known as contemporary graph theory. Graph theory involves the construction of a graph using nodes and vectors using specified rule sets. The graphs in the implementation would represent a building with IP Cameras as nodes at key locations with individuals creating vectors by moving between the cameras. The directed graph rule set states that a vector from Node A to Node B is independent and unique from a vector in the opposite direction. This rule set directly applies to the implementation because the analysis requires a person's exact location and, without a direction associated with vectors being declared, the location becomes ambiguous. This ambiguity becomes even more of a problem when including analysis based on time and identity. The weighted graph rule set states that a vector has a weight associated with it. This applies to the implementation because with specified

weights we can then assign probabilities to each path taken, and we can then modify the probability weights based on frequency, time, identity and number of connected nodes. These weights can also provide pattern based analysis especially when examining a graph pertaining only to individuals.

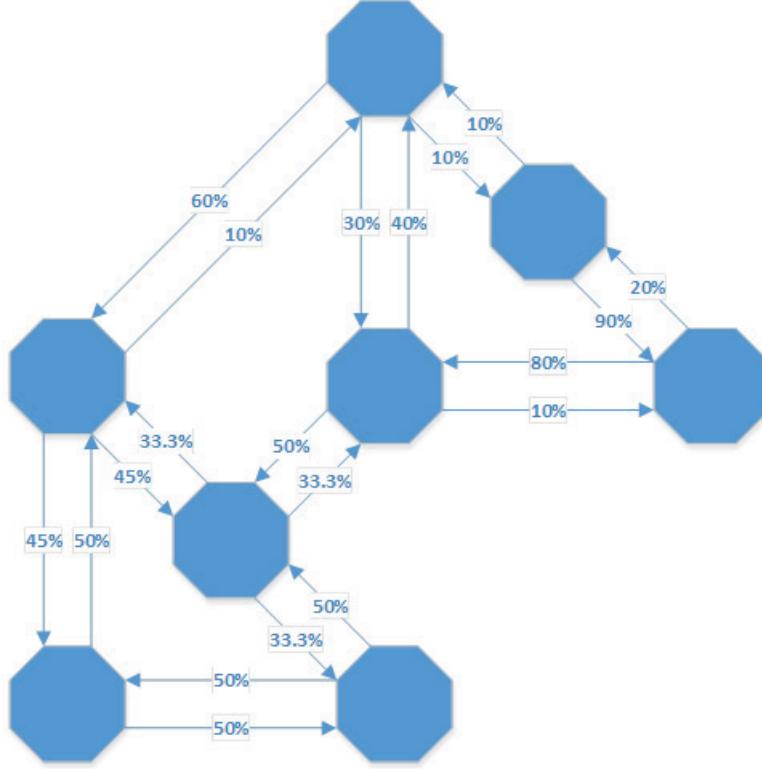


Figure 8.1: Sample of a Directed Weighted Graph

In order to show the possible capability of the directed weighted graph analysis a series of sample situations have been devised. Initially a person would walk into the view of the cameras and be detected, recording the relevant information into the database. After the database has saved enough information about the cameras it can begin to provide analysis on the actions of individuals. As a recognized person begins to move throughout the network of IP cameras, an anomaly can be generated if they move into an infrequently traveled area (taking a path with low weight). The system would save all the information on vectors including information regarding individual creating these vectors, this allows the creation of graphs for each individual or individually based analysis. These individually based graphs can then be used to further enhance the accuracy by examining personal movements with algorithms based on frequency and patterns. Another situation includes the generation of graphs based on time if an individual is entering the buildings at night where no night shift exists, an anomaly is generated. This type of analysis can also be expanded when using individual based graphs to recognize night shift employees based on their time patterns, creating even more cases for exceptions or confidence in anomaly detection. On top of all of these methods we can also add pattern based

analysis, which can also be applied system wide, individually, and to different times of the day. However pattern based analysis can also create unique forms of anomaly detection like the building entrance example, which says that if a camera is posted on every entrance and an individual appear inside a building without passing through an entrance creates an anomaly.

Directed Weighted Graph Analysis is still one of the most innovative ideas behind the project, but unfortunately due to various constraints it was not able to be implemented in the project. In order to implement this idea it required large amounts of already generated data. This lack of generated information became one of the primary constraints. During the construction of the project, parts of the system were implemented individually in enhance modularity. This proved a good strategy, but has delayed all data generation. This delay of information made this type of behavioral analysis effectively impossible to accurately implement in the time frame. Another major constraint for this analysis was the budget. This type of graph analysis requires many locations and nodes to be accurate and tested thoroughly, requiring many expensive IP Cameras. All these ideas involved with graph based analysis proved much too ambitious for the scope of the proof of concept project. In order to complete this project some type of behavioral analysis had to be generated, so we chose to only pursue Frame Duration Analysis which was originally going to be implemented into the project as a subsection of the anomaly generation.

8.2 Frame Duration Analysis

Frame Duration Analysis was originally a small component of behavioral detection software that the project was intended to run. This idea came from two potential applications for the project: security and safety. The intention of this algorithm is to identify anomalies based on the speed of an individual. For example, in a hospital if a patient collapses in an abandoned hallway or if a criminal is attempting to escape a building quickly the system would detect it. Frame duration analysis is actually quite simple. It takes the duration of an object within a camera frame and compares to the duration of previously seen objects. In both of the described situations the movement of the individual would quickly become anomalous because they are moving much slower or faster than the average human.

8.2.1 Implementation

Frame Duration Analysis uses a statistical model known as a normal distribution, shown in Figure 8.2. This can only be done because statisticians have determined that human walking speed is associated with random variables that approximately correspond to the normal distribution [13]. In order to compare the movements of individuals to each other, the system creates an Event for each object that appears in a camera. An Event contains lots of information including the individual, the camera, and timestamps of both appearance and

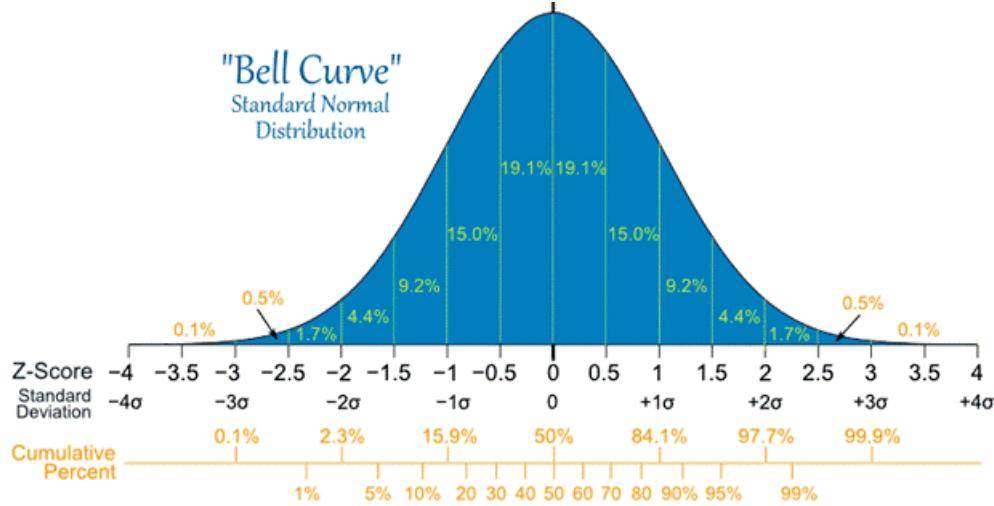


Figure 8.2: Sample of a Normal Distribution (Image Courtesy of Mathisfun.com [3])

disappearance. Using the change in time between entry and exit of an individual in a camera frame, the duration is calculated. Since the cameras do not move we can assume that the distance is constant, therefore we can use the duration as a relative speed variable. In order to generate anomalies the system saves the statistical information for each camera in the database. When an event is completed, we then take the saved camera information and calculate likelihood of this event occurring naturally. If this event has a 5% or less chance (two standard deviations from the mean) to occur it is assumed to be anomalous. The camera statistics are then updated by including the new Event information into the statistical model.

8.2.2 Implementation Complications

The simple implementation of Frame Duration Analysis as it is described above contained many assumptions and oversights. During the conceptual creation and implementation of the behavioral analysis engine many of these issues were identified and solved, based on various theoretical situations. The first situation we encountered was if a person collapses for extremely long period of time (a collapsed patient). Including their statistics in the model would skew the data and provide more inaccurate results. We have since included checks where if the individual falls within the most unlikely 0.3% chance (three standard deviations from the mean) of occurring the statistics are thrown out. Another of the biggest problems with the implementation of this behavioral analysis is the non-live factor of producing anomalies. Since the anomalies require that an Event be closed before generating the anomaly, monitoring staff would be pointed to a camera where the anomaly has previously occurred. In order to solve this problem, active events are polled against the slow end of the anomaly threshold constantly. However, it is impossible to generate an anomaly for the fast end of the anomaly threshold before the individual leaves the frame in the current implementation. Another major

problem with the system is that while reducing the amount of information needed before analysis, a normal distribution still requires at least 36 data points to be relatively accurate.

8.2.3 Constraints and Assumptions

The scope of the project, and focus on other components has forced us to make a few assumptions that do not hold in reality. The most important of which is the assumption of consistent distance within a camera frame. Simply put, we ignored the depth dimension of the camera video feeds. This can cause major problem in practical applications because an individual further from the camera will take longer to exit the frame than individuals much closer to the camera. Ideally with more time this could be implemented solving many of the biggest problems with behavioral detection by allowing real time anomaly generation and more accurate results. Since the implementation of this system was intended for indoors we have discarded the possibility of the use of vehicles, which could potentially cause many false positives in the anomaly detection.

Frame Duration Analysis is much easier to implement than Directed Weighted Graph Analysis because many of the previous constraints on budget and information no longer apply. Directed Weighted Graph Analysis has a macro based approach, viewing the building as a whole network of IP cameras, whereas the frame duration analysis only applies to the individual cameras. This allows the behavioral analysis engine to be tested without having large budgetary needs. The other major constraint associated with Directed Weighted Graph Analysis was the lack of generated information. Since Directed Weighted Graph Analysis is a macro based approach the amount of information required to generate accurate results was large. Frame Duration Analysis only requires 36 data points in a single camera frame to begin generating anomalies.

8.3 Exit and Entry Type Analysis

Exit and Entry Type Analysis was invented during implementation to solve one of the biggest problems associated with behavioral analysis, but after further consideration became an essential component of unifying the theoretical model of the project. This type of behavioral analysis examines the locations of the entry and exit of an individual in a camera frame.

A major flaw of Frame Duration Analysis is that it fails to account for the position of the object when it enters the frame. This can be a major problem because individuals that become obscured mid frame are likely to generate anomalies and negatively influence the accuracy of the system. In order to fix this problem a case was created to state that if an individual was detected to entering in the middle of the camera frame and anomaly would be created. Since the detection algorithms cannot guarantee the consistent detection of an object a buffer was created. This buffer allows for the brief lapse in detection without assuming the object has left the frame. Without this buffer any lapse in detection would generate anomalies, which proved to be a

common occurrence during tests.

The idea behind Entry and Exit Type Analysis could be expanded to incorporate benefits in both Frame Duration Analysis and Directed Weighted Graph Analysis. During the setup phase of the system the common entries and exits of individuals from a camera could be identified and used in graph analysis. As individuals travel between the cameras the exit position in a camera could greatly increase the probability the system could predict the next camera in which they will appear. For example an individual exits one of the cameras on the left hand side where a hallway exists only leading into the next camera, but does not appear, an anomaly can be generated.

Chapter 9

The User Interface

The User Interface is the only place for interaction between the user (the surveillance staff) and our system. It is built to engage the user by following modern design principals and relay the important components of our project efficiently. The User Interface design started with a mock up design seen in Figure 9.1 for conceptual confirmation, and placement of the functional requirements.

9.1 Mock-up Design

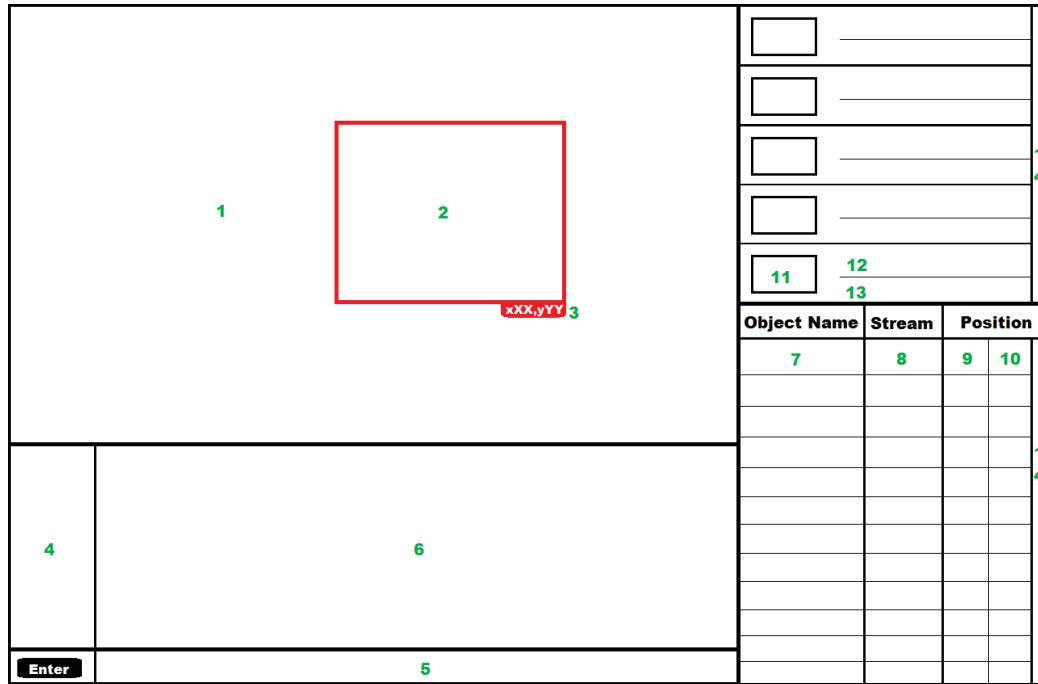


Figure 9.1: Sample Layout of the User Interface

The mock up interface was designed to incorporate modern design principals used in other popular User

Interfaces. The components of the User Interface were placed in similar locations to other popular interfaces, primarily having the text box or log at the bottom of the screen and the menu systems on the right or left hand sides for the screen. Another modern principal commonly used in photography is the rule of thirds [38], which states that the primary focus of the screen should have its center approximately a third of the way into the screen and was done for the video feed. This is a list describing the all components of the User Interface, shown in the Figure 9.1, and describes how each interacts and adds to the User Interface:

1. Video Steam - A simple video stream that displays the video feed from the IP Camera unmodified. This is an essential component to all surveillance programs.
2. Identified Object - Displays the most interesting/abnormal or selected object on screen. Clicking on the object in the object view on the right will switch the streams and highlight the individual on screen. (This was deemed an additional feature and could not be implemented in time.)
3. Position on video - Displays the position of the object in the bottom corner. (This was deemed an additional feature and could not be implemented in time.)
4. Time Stamps - These time stamps simply act as a log reference to help determine the frequency of actions.
5. Text box - This component acts as an input for the user in order to enhance the information recorded. (This could potentially be expanded to intake commands and provide additional information.)
6. Information Log - Displays all information about camera statistics, input text, setup and management notifications, and most importantly provides notifications of behavioral anomalies.
7. Object Name - This field displays object name and it should have the ability to rename objects. This should be linked with the database to save inputted information and load new information as it appears. It is part of the Object Row.
8. Stream Number - This field displays the Stream number the object is found on, and is not editable. It is part of the Object Row.
- 9 & 10. Position - This field displays the X and Y position of the object on the screen, and is not editable. It is part of the Object Row.
11. Thumbnail - This field displays a static image of the camera stream, and should be editable to the user. It is part of the Stream Row. The Stream row can be selected by calling a function that changes the stream selected.
12. Stream Number - This field displays the stream number, and is not editable. It is part of the Stream Row.
13. Location & Description - This field displays the description of the camera location, and is editable to the user. It is part of the Stream Row.

14. Scroll Down bar - Scrollable bars to allow for multitudes of objects improving scalability.

9.2 Requirements

This project will be expected to meet certain standards set forth by Santa Clara University; requirements have been determined in order to ensure the success of this project. However since this project is a collaboration between computer and electrical engineering the term “requirements” has been blurred. Also this is a proof of concept project only further confusing the result of this project. Unlike most traditional types of engineering, computer engineering views requirements as negotiable. It considers only the bare minimum to be truly required and the rest are extra features. In a computer engineering project features can greatly affect the outcome of the project for improving performance, ease of use, accessibility, and other widely scalable non-functional requirements.

9.2.1 Functional Requirements

A functional requirement is a component or feature of our system that either does or does not exist. As our project is highly modularized each section of system will have its requirement analyzed independently after they have been described in the design of our system. The requirements that are described in this section are for only the User Interface and are meant to help us judge whether or not the User Interface component of this project was a successful venture.

- Video Surveillance Feeds - The User Interface must be able display live security video feeds from the IP Cameras to the monitoring staff in real time. This includes the adding, swapping and loading of video feeds.
- Behavioral Analysis - The User Interface must be able to visibly notify the monitoring staff of behavioral anomalies in real time. In order to do this it must be able to establish a reasonable protocol for connecting to the database.
- Auditing Support - In order to ensure that the users are not abusing the power and information provided by our system, the User Interface must provide some way for the users actions to be saved in the database to be audited at a later date.
- Information Correction - The User Interface must be able provide some way for the users to change the information saved in the database. Our system is not guaranteed to be accurate so the users should be able to change inaccurate information. This information can also lead to corrections in our other systems to improve accuracy.

- Management & Setup - The User Interface must provide information and tools to easily setup and manage the system.

9.2.2 Non-Functional Requirements

Non-functional requirements are characteristics of the final project that cannot be classified as a feature or function, but rather are scalable and vaguely defined.

- Usability - The User Interface must be clean, intuitive, and easy to use for the monitoring staff.
- Modularity - The User Interface must be easily modifiable in case other components are replaced.
- Accuracy - The User Interface and the system must be able to provide some level of accuracy in our analysis, in order to make this a efficient product for the monitoring staff to use
- Security - The User Interface must provide easy access to important information while still remaining secure.

9.3 Implementation

After the components of the mock-up design had been established the actual implementation was able to take place. Just like any other project the idea put forth originally had to be re-evaluated and made more practical to implement. The functional requirements that were outlined during design of the mock up interface had to be prioritized forcing some features to be cut (e.g. the object information box in the middle of the video feed).

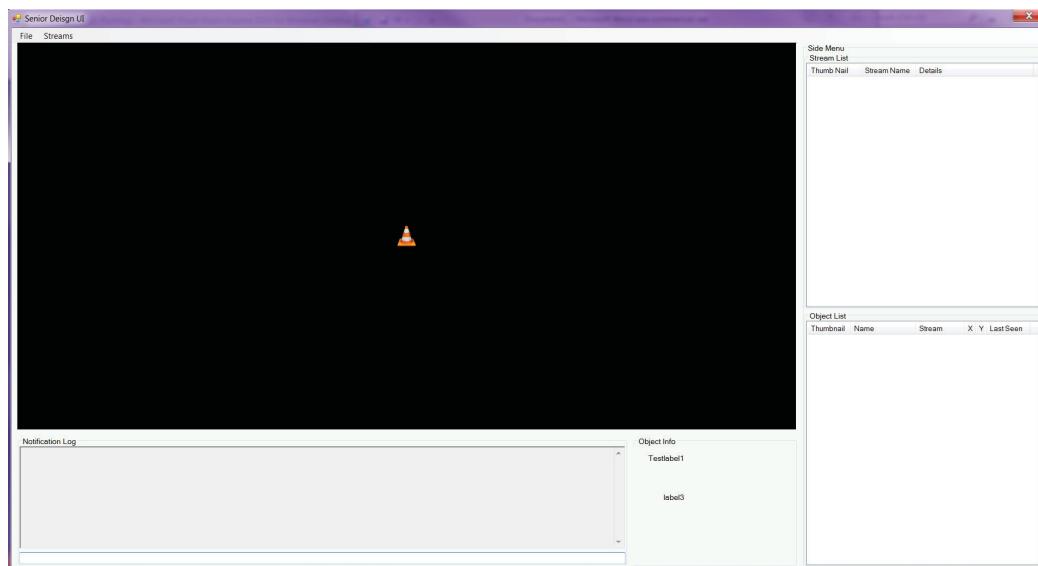


Figure 9.2: Recent Layout of the User Interface

The start of the implementation began with a massive struggle to effectively understand the tools we were generously given by Microsofts Dreamspark Program, primarily using Visual Studio Ultimate for the User Interface. After struggling for some time on the basic User Interface implementation, progress started on the primary focus for the screen - the video feed from the IP Cameras - because of its high priority. The video stream requires a connection from the User Interface to the IP Camera. In order to do this a series of calls must be made over the network creating a RSTP (Real Time Streaming Protocol) stream. While attempting to setup and understand the RSTP streams, the discovery of an API (Application Programming Interface) for RSTP streams allowed the bypass of a large part of this implementation. This discovery was a major boost to the progress of the User Interface because it saved hours of implementation and understanding of the complicated base level, as well as providing base level of implementation and accomplishment. This API was provided by VideoLAN, the developers of VLC Player, who created a VLC plugin (with RTSP support) for ActiveX in the .Net language libraries [39].

After the implementation of the video feed, the next largest priority was the implementation of the menus, lists, and other navigational components. This step of the implementation began with the placement and visual design of the User Interface by using and understanding the visual elements provided in Visual Studio Libraries. This level of implementation can be seen in the screenshot of the interface, Figure 9.2, as everything was visually in place but had no functionality. The next few steps in the development of the functionality of the User Interface became a mess of new ideas and backtracking, as the limits of actual functionality were realized. The current implementations of the next few steps to the User Interface have been outlined below.

- **RSTPStream** - RSTPStream is a class created to abstract away the elements required to setup a RSTP stream through VideoLANS API and to properly display the information on screen. This class has the necessary getters, setters, and constructors, but has very minimal implementation because of its use simply as a layer of abstraction.
- **ListView** - This is a built-in class in the .Net libraries but, in order to properly display the information in the right hand side of the User Interface, a firm understanding of its implementation is required. There was lots of testing done with custom modifications to the standard ListView class but none accurately portrayed the information and the solution was eventually found in the default class in obscured references to something called the Grid View.
- **Globals** - Globals was a class created in order to allow access to specified elements from all classes primarily from the main User Interface class Form1 to the Addstream class where the RSTPStream information needed to add and double check classes.

- AddStream - This is a secondary form used to popup when the user attempted to add another stream to the list. This has its own functions and methods enabling the adding of RSTPStream to the Database and the Globals class.
- Thumbnails - From the beginning thumbnails were wanted to be automatically added when a stream or object was added to their respective lists. Unfortunately these attempts were abandoned to be replaced with still images to save time.

The next important step was the construction and setup of the database; this can be seen in Chapter 7 the database implementation chapter. Since all of our software was provided by Microsofts Dreamspark, synergy was created between programs which allowed the database schema to be directly imported to the User Interface. This importation allowed the simple creation of many new classes and increased the speed of implementation during the rest of the project. Ideally many of the previous created classes could be modified to suit this schema to increase cohesiveness and future modularity but time constraints have limited this. The database connection also allowed the User Interface to directly pull the information saved in the database to be relevant information in behavioral analysis. The behavioral analysis requires that the information generated in Behavioral Analysis Engine be updated in real time to the User Interface. In order to do this without creating major stalls in the User Interface a second thread was created to poll the database on a second by second basis. This is where progress was halted in order to meet other deadlines for our project.

9.4 Technologies Used

The technologies used in this User Interface are listed below.

- Microsoft Visual Studio Ultimate 2013
- Microsoft LINQ Libraries
- VideoLAN ActiveX Libraries
- Team Foundation Service
- Git

9.5 Risk Management

Throughout the planning of this project we also took into account the risks involved, and the prospect of not finishing our design. In order to mitigate the results of our risks we rated each one based on severity and impact and created mitigation strategies. The likelihood of each risk occurring is outlined in Figures 9.3

under the P or Probability column and the difficulty of the solution to the risk is seen in the S or Severity column. The I or Impact column is the product of the other two columns and represents how greatly the risk will affect the project.

Now that the project has been mostly implemented this table, Figure 9.3 was relatively accurate, however not completely. The major differences are that the problems from new technologies were much more significant than expected and the electrical integration had been delayed repeatedly. Also, unexpectedly most software projects have rapidly changing requirements, whereas this project has had relatively no changes to requirements because this is a proof of concept project. This lack of changing requirements might also be because the database stands between the User Interface and the rest of the project so any changes can be incorporated in the database schema instead of the User Interface implementation.

Project Risks	Consequence	P	S	I	Mitigation
Electrical Integration	Delayed end to end functionality	0.5	6	3	Weekly Meetings, Allow for extra time
Changing Requirements	Incorrect project functionality	0.4	7	2.8	Design for Change
Miscommunication	Delayed end to end functionality	0.5	4	2	Weekly Meetings
Sickness	Delayed checkpoints, Specific Modules Delayed	0.2	5	1	Weekly Meetings, Long Term Planning
New Technologies	Time delays	0.2	5	1	Plan for Extra Time, Weekly Meetings
Data Loss	Major time setback, Unable to Meet Deadlines	0.1	5	0.5	Maintain up to date Backups

Figure 9.3: User Interface Implementation Risks

Chapter 10

Test Plan

Similar to the development of any system, a test plan was put in place at the beginning of our project in order to ensure a better final product. Since this project was a proof of concept, many the quality concerns were temporarily ignored in order to further implementation and attempt to get a working pipeline. Since many of the components of our project are still in development due to the massive scope of our project, extensive testing has not been conducted.

10.1 Unit Tests

Unit testing will be done throughout the implementation of every module contained within the project. Considering the independent development of each module, the developer of a module is responsible for the unit test of their components.

10.2 Integration Tests

Once modules have progressed far enough to be near independent completion, the connections between components will be established and tested. Integration tests are much more important to the construction of this project because these tests can be used to help prove the concept behind the project.

10.3 Test Media

For the video processing systems (FPGA back-end and Identity Engine), a multi-tired test plan was developed in order to assess the suitability and reliability of those subsystems. The test plan is described as follows:

1. Record video from actual cameras used in the project. This footage will be used to provide a consistent stimulus to all test cases.
2. Develop a simulation of the image processing component in MATLAB.

3. Run the simulation with the recorded test video, noting any false positives and false negatives.
4. Fine tune the simulation and retest until pass rates are within an acceptable range.
5. Develop the FPGA or GPU implementation of the system and run the hardware implementation with the same test video, checking for false positives and false negatives.
6. Compare these results to those from the simulation to ensure consistency.
7. Debug implementation, if possible, and account for discrepancies.
8. Perform a live stress test with many moving objects in an attempt to break the implementation.

10.3.1 System Testing

Finally, after all the components have been connected, system testing begins to ensure the entire system works as a unit. This consists of a full end-to-end test of the system with both recorded and live video.

10.4 Test Results

Due to the limited development time, extensive testing has not been conducted up to this point. However, simulations of the FPGA back-end have been run against test video captured in the SCU Image Processing Lab. It was determined that object detection was reliable when the objects were close to the camera, but began to become less reliable as the object reached the end of the laboratory (approximately 30 ft away). It was determined that this is likely due to the tolerance parameters on the algorithm being too strict. It is expected that further tuning of the parameters will result in improved stability and performance.

Chapter 11

Ethics

11.1 Societal Implications

Most people feel that they have a right to their own privacy. The Bill of Rights in the United States protects the privacy of its citizens from the government with the unreasonable search and seizure clause. While people are not afforded quite the same level of protection under the law when in public, a certain amount of privacy is expected in our society. Up until recently, having some relative degree of privacy in public was a reasonable assumption. In the years preceding the widespread adoption of CCTV systems, people could move throughout the public sphere and experience some privacy. Even after institutions began installing their own CCTV systems, it was still impractical to track every action of an individual even with monitoring staff. By inserting identification technology and behavioral analysis into networked CCTV systems, the project removes this last limitation to easily tracking the interactions of an individual. Without controls, this system could be hazardous to our society; it would be easy for someone with access to the system to put a tracker on a specific individual and receive updates on their interactions. For example, the system could leave subjects vulnerable to stalking by individuals with access to the system if safeguards to prevent this type of misuse are not put in place.

In order to address the privacy implications of the system, it was designed with limits placed on the types of interactions and amount of information staff can access. In a commercial implementation of the system, the monitoring staff would be able to view video feeds just as they do today. The video streams would be augmented to show what has been detected as a face and if it has been successfully identified. However, this view would not display any identification information about the detected individuals unless they exhibit a behavioral anomaly and the severity of that anomaly is considered high enough. One such scenario would be if a patient in an assisted living facility collapses. In this case, information such as location, name, and medical conditions would be disclosed to the monitoring staff to aid in helping the victim. Any use of the system to locate or back-trace individuals movements would require an individual's supervisory approval.

Any such requests and approvals would be logged by the system and would be subject to auditing to help prevent abuse.

While there are privacy concerns with a system like this, there are some aspects of a smart CCTV system which can actually correct injustices that are present in our current society. Because of the way this system was architected, alerts are only created based on behavior that deviates from a model built off of prior observations by the system. Unlike human monitors who are susceptible to race, gender, and other biases, the computer is not. This can lead to more equal and non-discriminatory utilization of CCTV systems in security applications.

11.2 Project Development

One of the biggest concerns and ethical implications of this project is the safety and security of people being monitored. Depending on the application, there may be different expectations of the performance of the system and consequences based on those expectations. For example, in a security application, the system may act in parallel with trained monitoring staff. The system would be able to alert them to incidents that it believes are suspicious. However, the monitors should still be attentively looking for incidents the system missed. In a medical application, there may not be staff monitoring the camera feeds in real time. If a patient collapses and the system does not catch it, there is the potential for the patient to die. A reality of behavioral analysis is that it is not entirely accurate or precise. In order to maintain the potential positive impact of the project, the team must effectively communicate the limitations of the system. However, if customers do not fully understand the limitations of the system, they may discontinue activities that catch events the cameras miss. The project is meant as a companion tool rather than a replacement for monitoring staff.

Another major concern is the privacy, security, and storage of personal data. Data anonymization and retention, specifically, are common contemporary issues. Many companies store this information in order to make more informed business decisions and dynamic marketing strategies. The project maintains an isolated database of personal information entirely generated by the system to ensure data integrity. In a commercial implementation, this system will restrict access to the data by only displaying information to authorized monitoring staff when the system believes it is important, and by tracking all other requests for information. The system will make it easy for the customer to configure the data retention policy to align with the organizations documented standards.

11.3 Team and Organization

With the project focusing on such a sensitive technology as CCTV, the need for personal and team ethics take on especially high importance. All of the team members understand that, while the project can have profound social benefits, it can also be used for more nefarious ends if the ethical issues are not adequately considered. As such, each team member has the responsibility to consider the ethics of each design decision. If a team member believes that there is an ethical issue that is going unnoticed or unaddressed, they have the responsibility to bring it up with the team and advisors. Issues are discussed within the group and with advisors so that the most ethical path is taken. If a team member feels that the system no longer fulfills its goal of benefiting society and is instead doing the opposite, they would be justified in leaving the project. Team members understand that their actions not only reflect on themselves but also on the group as a whole, the advisors, the university, and supporting corporations. As such, for each action, the team understands that it must consider the reputations of all stake holders and how a particular action will reflect on them. It is also possible that the team will be asked by one of the other stakeholders in the project to make certain design decisions. It is the responsibility of the team to consider the ethics of such requests. If a request appears to be ethically unsound, the team must discuss the request internally as well as with advisors. If an ethically appropriate course of action cannot be found to fulfill the request, the team must notify the stakeholder of their decision not to go through with the request and its reason for doing so. That way, the integrity of all parties involved is protected.

11.4 Summary

Like all CCTV technologies, this project has several points of ethical concern which must be addressed. The project runs the risk of violating individuals privacy by tracking their interactions within a monitored facility. In order to mitigate this issue, the development of strict controls were considered in the design of the system. The system actually has the potential to provide more ethical and non-discriminatory treatment of subjects than current human monitored CCTV systems. The design and development team must act responsibly throughout the project as well as ensuring the future use of this project is ethically oriented. The customer will need to be instructed on the proper use of the system while prevented from freely accessing personal information stored within it. The customer must also be informed about the limitations of the system in order to make informed decisions on how to properly integrate it into their environment. The project, while envisioned to have a positive ethical impact, must be closely monitored to ensure the goals of the system are achieved.

Chapter 12

Environmental Impact and Sustainability

12.1 Environmental Impact

Since this project is a form of computing system, it relies heavily on integrated circuits and copper for interconnects within printed circuit boards. The life cycle assessments of some components are presented to give some sense of the environmental impact of the electronics that make up the system.

12.1.1 Semiconductor (ICs)

According to *Life-cycle Assessment of Semiconductors* by Sara B. Boyd of the University of California, Berkeley [40], the total energy consumption per die (chip) of a semiconductor produced using the 45 nm process is 1699 MJ/die. Of that total, 1593 MJ/die is use, leaving 106 MJ/die in embodied energy. The same paper states that the total Global Warming Potential (GWP) per die of a semiconductor produced using the 45 nm process is $47 \text{ kgCO}_{2\text{eq}}/\text{die}$ with $43.2 \text{ kgCO}_{2\text{eq}}/\text{die}$ of that from use. This leaves $3.8 \text{ kgCO}_{2\text{eq}}/\text{die}$ from production.

12.1.2 Copper

According to the *Embodied energy and embodied carbon (ICE)* database by Dr. Craig Jones [41], the embodied energy of copper manufactured in bar or sheet form in the EU is 42.00 MJ/kg . Its embodied CO₂ is $2.60 \text{ kgCO}_2/\text{kg}$ and $2.71 \text{ kgCO}_{2e}/\text{kg}$.

12.1.3 Reducing Embodied Energy and Greenhouse Gas Emissions

Since this is a proof of concept project, it mainly uses off the shelf development kits for the hardware platform. No custom PCBs are being manufactured and no ICs are being fabricated. There is very little that can be done at this stage to reduce embodied energy and greenhouse gas emissions. However, should this project progress

and custom PCB be created, there are some measures that can be taken to improve the environmental impact. Since the system uses ICs from Altera, we have no control over the production process of these devices. However, we can select the IC with the smallest die which fulfills our needs. The smaller the die, the lower the environmental impact per die. A much more practical way to reduce the environmental impact of the project is to make the PCBs as small as possible with the shortest copper interconnects possible. Not only are short interconnects good for signal integrity, but they use less copper and thus lower the embodied energy and greenhouse gas emissions of the custom hardware components.

12.2 Sustainability

Since the project is a proof of concept system, the sustainability of the prototype was not of primary concern. However, ecological protection, coupled with economic considerations, were taken into account in the design of the overall system architecture. The system does require the purchase of FPGA Back-end boards for each camera as well as a server to contain the database and run the behavioral analysis engine. However, a conscious decision was made to architect the system so that the preliminary image processing would be conducted at FPGA boards attached to each camera rather than designing entirely new network cameras to perform this function. This prevents customers from needing to replace their entire IP camera installation with new ones when installing the behavioral analysis system which is the focus of this senior design project. In addition to being able to use existing IP cameras in an installation, the project was designed so that the existing recording and archival equipment could be used without modification. This is because the system being developed in the senior design project is designed to augment existing setups with behavioral analysis. The behavioral analysis system inspects video in real time and, because of this, does not need to manage or have access to video recordings. The system does store logs of what events happen and at what time, but these logs consist primarily of metadata and do not require the large storage capacity of video archive systems. The event logs can be combined with recorded video at a later time by matching the timestamps in the metadata logs to the timestamps in the recorded video.

It is also expected that, as is the case with many new systems, the cost of the system (if productized) would decrease over time. After the initial proof of concept, additional research could take place in how to improve the efficiency of the design. It is possible, for instance, that multiple cameras could be processed by a single FPGA Back-end board. If this is the case, fewer FPGA Back-end would be required in an installation, reducing the overall cost of the system. Further improvements in the software running on the server could also possibly increase the server capacity or lower the system requirements. These cost-lowering measures would make the technology more accessible to facilities which do not have the funds to install the initial

model of the behavioral analysis system.

12.3 Frugal Engineering

As with many new technologies, the initial cost of the system is expected to be high. As such, the first revision of the system was not designed for developing markets. The intention was that the early adopters of the system would be security conscious facilities with large budgets (such as secure government / military facilities or corporate R&D labs). However, as the technology matures, it is expected that the cost of the system will decrease. Even though the initial prototype is primarily using high cost, delicate development kits in place of custom hardware, several frugal engineering principles did inform the design of the system architecture. Because the system is meant to be used in safety and security applications, ruggedization was an important feature to consider. Since the Back-end FPGA boards need to be attached to each camera and, because cameras may be positioned outside, it is possible that the FPGA would need to reside outside. Since the functionality of the system depends on the FPGA board functioning correctly, it is important that the FPGA board enclosure be as weather proof as possible. The design of the enclosure would likely consist of high strength plastic coupled with rubber gaskets to protect the circuit board. All of the connectors would be rated for use outdoors. While the ruggedization has implications for the product application, it also makes it automatically suitable for deployment in developing countries where it would potentially not be housed in a heavily controlled indoor environment.

The project was designed to adapt to current IP surveillance installations. While potential customers in the developing world may not have existing IP camera installations, the system does provide the flexibility to use whatever IP cameras are available and within the price range of the customer. This could potentially improve the affordability of the system.

The frugal innovation concepts used in this project including ruggedization, adaptability, and affordability do provide some support for the professional issues and constraints detailed in the senior design requirements. The adaptability and affordability addresses the economics of the system. While it is likely that the system will initially be costly to create, the flexibility given to the consumer to use either existing IP cameras or to buy ones that are within their budget will likely help early adoption of the system. As the technology is refined and efficiency is improved, it is likely that the cost of installations will decrease, bringing the system within the budget of more organizations. The ruggedization and adaptability of the system also addresses the potential environmental impact of the system. Since the system works with existing IP cameras, facilities do not have to discard existing IP cameras after the installation. This reuse lowers the environmental impact of the project. The ruggedization of the boards, coupled with the fact that the FPGA designs can be reconfigured

at a later date with new designs means that the system should remain useful for a long period of time. By not forcing customers to upgrade all parts of the system constantly, the environmental impact of the system is reduced.

Chapter 13

Aesthetics

13.1 Aesthetic Considerations for Productization of Project

First impressions are some of the most influential factors when someone is making a purchasing decision. The aesthetics of a product are often the first impression a perspective customer has. If the aesthetics do not leave the customer with a good impression, the customer may take their business elsewhere even if the product is functionally superior to the competition. Since the project has commercial potential, it is especially important to take visuals into consideration.

While all users within a facility are being monitored by the system, they are not considered the primary end users. The end users of the system are the monitoring staff who are required to watch and analyze the CCTV video feeds. The bulk of user interaction with the system occurs through the Graphical User Interface or GUI. The GUI is the component of the software that shows information as graphical elements and supports interacting with the system via a mouse, touch screen, or other pointing device. GUIs are vastly superior to text-based user interfaces for conveying multiple pieces of information to the user at once. In this case, it also simplifies the user's experience by hiding the implementation details of the system from the end user.

In addition to providing a good first impression of the system, it is also important to provide an experience that abstracts away the complexity of the underlying implementation. Most people are not interested in the inner workings of their products and are only concerned that the product performs as advertised. The idea of this project is to provide a seamless experience to the client. This includes making our hardware as unobtrusive as possible, ideally keeping it out of sight. The goal of our project is to blend it in with standard CCTV equipment. To accomplish this, the hardware boxes connected to each camera need to be small, discrete, and capable of being placed where existing wiring is routed.

13.1.1 Graphical User Interface (GUI)

The user interface is designed to be used by a small, trained, monitoring team. As a result, the software design is not as aesthetically pleasing as some other more widely used consumer products. Due to the overall scope, the project is emphasizing functionality and features over aesthetics. Despite the circumstances of the project, a few software oriented aesthetic ideas were implemented including: graphical user interface layout and key aesthetically oriented functional features.

The user interface is oriented similarly to most modern user interfaces (see Figure 13.1). The main section of the interface is the surveillance video feed. This section is designed to occupy two thirds of the screen for visual appeal. The remaining part of the screen is divided up into two other sections. While complementing the modern look of the system, it grants additional functionality for information selection and control. This design feeds off a modern photography concept called the Rule of Thirds, which suggests that an image is more appealing if the main focus is off center [38]. Adding functionality to the sides of the window, not only makes the interface more usable, but also more aesthetically pleasing. This concept is commonly used in software engineering today, particularly in graphical user interfaces.

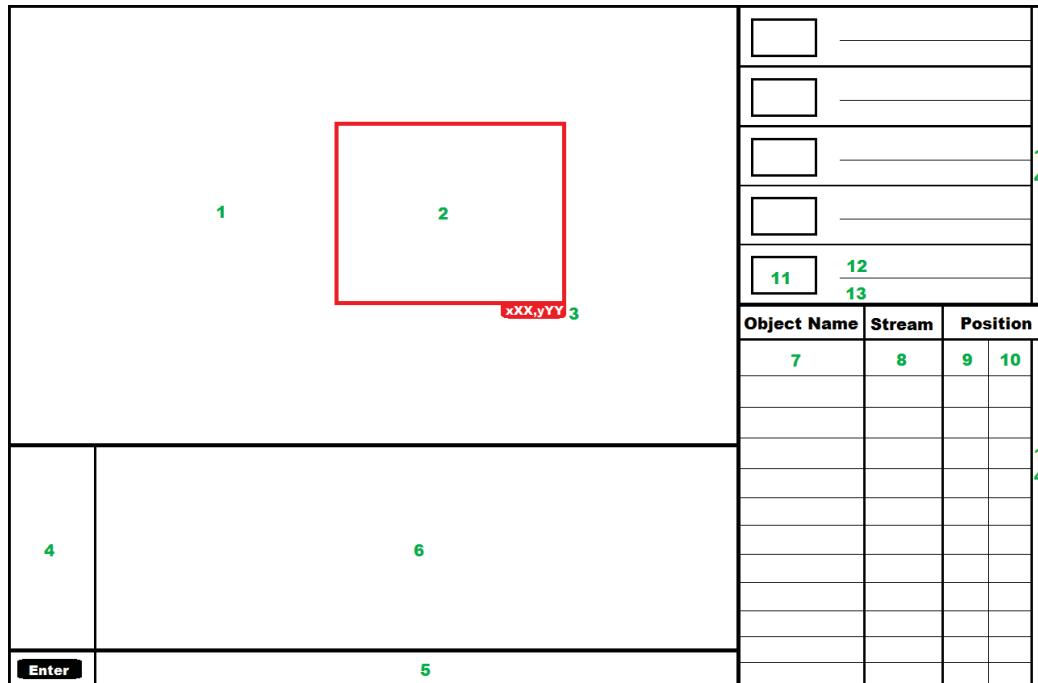


Figure 13.1: User Interface Concept

Some of the main features of the project were not only included for ease of use, but also for aesthetic appeal. The mouse can be used to imbue a more dynamic and cleaner feel to the user interface. Revealing dropdown menus after clicking on an object or video stream is one example of these features. The more time

available to develop the project, the more refined the interface can be made.

13.1.2 Hardware

There are three major hardware components that make up the system: the CCTV cameras, the servers, and the FPGA back-end boards. The aesthetics of each component is described in depth below.

CCTV Cameras

The CCTV cameras are purchased off-the-shelf for the system by the customer. The only requirement is that the cameras are ONVIF compliant, which many commercial cameras are. As such, the aesthetics of the cameras are outside of the scope of our project and are determined by the customers purchasing the system. Ideally, the customer purchases cameras which fit well with the environment in which they are being placed. This is the only part of the system that is visible to the general public with the possible exception of the FPGA back-end boards depending on the installation.

Servers

Like the CCTV cameras, the servers are provided by the customer. The system uses the server to run the Identity Engine, the Database, and the Behavioral Analysis Engine. While these components are crucial to the functionality of the system, the servers can be located anywhere in the facility. It is likely that the servers will be located in the organizations server room or datacenter where the general public does not see them. Since large organizations usually have preferred server providers and often have an existing server installation, the aesthetics of the server are outside of the scope of the project and rest with the customer.

FPGA Back-end Board

The FPGA Back-end board is the only the component of the system, besides the cameras, which may be visible to the general public. It is also the only hardware component in the project with which the team has direct control over aesthetics. Each FPGA Back-end board is linked to a camera via an Ethernet cable and switch. Because the board does not need to be physically attached to the camera body, it may be hidden discreetly. For example, the board may be hidden above ceiling tiles where the general public cannot see it. One of the guiding principles behind the project was that the behavioral analysis system would be able to integrate seamlessly into existing CCTV systems. As such, the FPGA Back-end boards should not draw attention to themselves. In order to maintain their low profile, commercial implementations of the boards would be encased in opaque plastic shells with slots for mounting brackets. The shape of the enclosures would feature rounded corners with tapered edges and would be thin so that it is less noticeable if a wall mount is required. The shells would have several status LEDs to indicate the operational state of the boards.

LCD line displays would be visible through a window in each enclosure and would display detailed diagnostic information such as the boards IP address and error codes. These features would allow a technician to diagnose problems with the board efficiently while still not drawing attention to the board from the general public.

It is worth noting that the scope of this senior design project is a proof of concept system. As such, a development kit has been used in place of a custom FPGA Back-end board. Since the boards being used are not the ones that would be used in a final product, the enclosure described above will not be produced for this project. The description above represents the aesthetics of the system after the development of custom PCBs (Printed Circuit Boards) for the FPGA Back-end.

Chapter 14

Conclusion

CCTV is one of the most commonly employed safety and security technologies in use today. However, the use of CCTV to identify and react to incidents in real time has been hampered by its need for human monitors to make sense of the video feeds. In most cases, the relatively small number of human monitors are unable to observe the potentially exorbitant number of cameras feeds simultaneously. This has led to CCTV being a primarily retrospective tool. Due to new advances in technology, it is now possible for machines to perform some tasks which previously required a human monitor. The project being proposed seeks to use this new technology to augment existing CCTV systems with behavioral analytics. The system uses a collection of FPGAs and computers to process video from the cameras and track object movement in a facility. It uses this information to build a model of normal behavior and flags abnormal movement for review by monitoring staff. This semi-automated system is not meant to replace human monitors but is meant to draw their attention to relevant camera feeds.

Since this is a Senior Design project, there was a very limited time span to develop the system. In light of this, the scope of the project was scaled back to be a proof of concept system. Because several components of the projects could be Senior Design projects in their own right (such as facial recognition), these components were replaced with simpler ones to demonstrate basic functionality. An assumption was made that these components would continue to undergo independent development and could replace the simpler components used in this project at a later time.

While a substantial amount of progress was made in implementing the system, the final implementations of the FPGA back-end and Identity Engine were not completed. However, MATLAB simulations were written for each components and were used in the first end-to-end demonstration of the system. In addition, an IP core was created to demonstrate the dataflow proposed for the FPGA based accelerator. This demo is currently non-operational but is actively being debugged and is expected to be operational shortly.

The \$47,925.53 theoretical cost of the project may appear exorbitant at a first glance, however, it is actually

on par with existing safety and security technologies. Many CCTV systems utilize cameras that cost over \$1,000 each. The proposed budget accounts for both the materials required in an end installation as well as the development tools which are not part of the end system. Additionally, since this is a proof of concept design, pre-made development kits are being used instead of custom designed boards. These development kits contain more components than are necessary for the project. If production boards were manufactured, they would likely cost much less than the development kits. A preliminary cost estimate suggests that each FPGA back-end would cost approximately \$566.57 to produce. This is competitive to the pricing of existing “smart cameras” on the market.

Overall, this was a very ambitious project with profound real world applications. A large amount of the work set out to be accomplished was completed but there is still room for additional development and refinement. With additional development time and resources, this system could potentially be productized and brought to market.

Chapter 15

Bibliography

- [1] Altera Corporation, “Cyclone v soc development kit and soc embedded design suite,” Nov. 2013. [Online]. Available: <http://www.altera.com/products/devkits/altera/kit-cyclone-v-soc.html>
- [2] *Altera SoC Golden System Reference Design User Guide*, Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, May 2013, docID: UG-01138.
- [3] R. Pierce, “Normal distribution,” Feb. 2014. [Online]. Available: <http://www.mathsisfun.com/data/standard-normal-distribution.html>
- [4] *Cyclone V SoC Development Board - Reference Manual*, Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, May 2013, docID: MNL-01077-1.0.
- [5] Colfax International, “Colfax proedge sxt8600 workstation.” [Online]. Available: <http://www.colfax-intl.com/nd/workstations/sxt8600.aspx>
- [6] *Cyclone V Device Overview*, Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, Dec 2013, docID: CV-51001.
- [7] H. Yan, “Body found in hospital stairwell: San francisco sheriff details what went wrong,” 2013. [Online]. Available: <http://www.cnn.com/2013/11/07/us/california-body-in-stairwell/>
- [8] R. Salonga, “Santa clara teen stowaway’s survival in jet’s wheelwell was literally death-defying,” 2014. [Online]. Available: http://www.mercurynews.com/bay-area-news/ci_25610408/san-jose-santa-clara-teens-survival-from-flight
- [9] *ONVIF Profile S Specification*, Open Network Video Interface Forum, December 2014. [Online]. Available: http://www.onvif.org/Portals/0/documents/op/20984_11763r01ONVIF_Members-Profile_S-Specification.pdf
- [10] *ONVIF Core Specification*, Open Network Video Interface Forum, December 2012. [Online]. Available: <http://www.onvif.org/specs/core/ONVIF-Core-Specification-v221.pdf>
- [11] *Cyclone V SoC Development Board - User Guide*, Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, May 2013, docID: UG-01135-1.0.
- [12] *Cyclone V Device Handbook*, Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, Jan 2014, docID: CV-5V2.
- [13] Serge P. Hoogendoorn, PhD and Winnie Daamen, PhD, “Free speed distributions for pedestrian traffic,” 2006.
- [14] D-Link, “Cloud camera 2300, outdoor hd day/night network cloud camera - dcs-2310l,” May 2014. [Online]. Available: <http://www.dlink.com/us/en/home-solutions/view/network-cameras/dcs-2310l-outdoor-hd-poe-day-night-cloud-camera>

- [15] J. McCool, “100base-t: an overview,” in *WESCON’95. Conference record. ’Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies’*, Nov 1995, pp. 342–. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=485300>
- [16] F. Edwards and M. Schulz, “Performance of vbr packet video communications on an ethernet lan: A trace-driven simulation study,” in *Computers and Communications, 1994., IEEE 13th Annual International Phoenix Conference on*, Apr 1994, pp. 427–. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=504149>
- [17] Altera Corporation, “Altera product catalog,” 2013, v. 13.1. [Online]. Available: <http://www.altera.com/literature/sg/product-catalog.pdf>
- [18] Microsoft Corporation, “Maximum capacity specifications for sql server,” 2014. [Online]. Available: <http://msdn.microsoft.com/en-us/library/ms143432.aspx>
- [19] ———, “Binary large object (blob) data (sql server),” 2014. [Online]. Available: <http://msdn.microsoft.com/en-us/library/bb895234.aspx>
- [20] R. Mistry and S. Misner, *Introducing Microsoft SQL Server 2014: Technical Overview*. Microsoft Press, 2014. [Online]. Available: <http://aka.ms/684751pdf>
- [21] Microsoft Corporation, “Microsoft sql server 2014: Record-setting performance,” 2014. [Online]. Available: http://download.microsoft.com/download/4/F/C/4FCAE4E4-4410-4586-8F6D-E0196F96ACFE/SQL_Server_2014_Performance_Benchmarks.pdf
- [22] Altera Corporation, “Golden system reference design (gsrd),” Nov. 2013. [Online]. Available: ftp://ftp.altera.com/outgoing/devkit/13.0/cycloneVSX_5csxfc6df31es_soc_v13.0.0.1.exe
- [23] Live Networks, Inc., “Live555 streaming media.” [Online]. Available: <http://www.live555.com/liveMedia/>
- [24] IJG, “Independent jpeg group.” [Online]. Available: <http://www.ijg.org/>
- [25] *Avalon Interface - Specifications*, Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, April 2014, docID: MNL-AVABUSREF-2.1.1.
- [26] Altera Corporation, “Golden hardware reference design (ghrd),” Nov. 2013. [Online]. Available: ftp://ftp.altera.com/outgoing/devkit/13.0/cycloneVSX_5csxfc6df31es_soc_v13.0.0.1.exe
- [27] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4767851>
- [28] D. Banks, S. Linton, and P. Stockmeyer, “Counting cases in substitope algorithms,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 10, no. 4, pp. 371–384, July 2004. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1298795>
- [29] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’87. New York, NY, USA: ACM, 1987, pp. 163–169. [Online]. Available: <http://doi.acm.org/10.1145/37401.37422>
- [30] D. Rajon and W. Bolch, “Marching cube algorithm: review and trilinear interpolation adaptation for image-based dosimetric models,” *Computerized Medical Imaging and Graphics*, vol. 27, no. 5, pp. 411 – 435, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895611103000326>
- [31] Altera Corporation, “Golden top,” Nov. 2013. [Online]. Available: ftp://ftp.altera.com/outgoing/devkit/13.0/cycloneVSX_5csxfc6df31es_soc_v13.0.0.1.exe

- [32] ——, “Board test system (bts),” Nov. 2013. [Online]. Available: ftp://ftp.altera.com/outgoing/devkit/13.0/cycloneVSX_5csxfc6df31es_soc_v13.0.0.1.exe
- [33] ——, “Avalon memory-mapped master templates,” Sept. 2008. [Online]. Available: <http://www.altera.com/support/examples/nios2/exm-avalon-mm.html>
- [34] RocketBoards.org, “Documentation,” Dec. 2013. [Online]. Available: <http://www.rocketboards.org/foswiki/Documentation/WebHome>
- [35] Altera *SOC Embedded Design Suite - User Guide*, Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, 2013.
- [36] X. Zhu, R. M. Rangayyan, and A. L. Ells, “Digital image processing for ophthalmology: Detection of the optic nerve head,” *Synthesis Lectures on Biomedical Engineering*, vol. 6, no. 1, pp. 1–106, 2011. [Online]. Available: <http://dx.doi.org/10.2200/S00335ED1V01Y201102BME040>
- [37] “DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [38] J. Smith, *Remarks on Rural Scenery; with Twenty Etchings of Cottages, from Nature; and Some Observations and Precepts Relative to the Picturesque. By John Thomas Smith, Engraver of the Antiquities of London.* June MDCCXCVII. printed [by Joseph Downes] for, and sold by Nathaniel Smith ancient Print seller at Rembrandts-Head May’s Buildings, St. Martin’s Lane, and I.T. Smith, at No 40 Trith Street Soho, 1797. [Online]. Available: <http://books.google.com/books?id=YWZHtwAACAAJ>
- [39] VideoLAN, “VLC ActiveX plugin for .Net libraries.” Feb. 2014. [Online]. Available: www.videolan.org
- [40] S. B. Boyd, “Life-cycle assessment of semiconductors,” Ph.D. dissertation, 2009. [Online]. Available: <http://escholarship.org/uc/item/8bv2s63d>
- [41] C. Jones, “Inventory of carbon and energy (ice),” University of Bath, Dataset, 2011. [Online]. Available: <http://www.circularrecology.com/ice-database.html>

Chapter 16

Appendix

16.1 FPGA Specifications

FPGA specifications were duplicated in Table 16.1 and Table 16.2 from *Cyclone V Device Overview* for the reader's convenience [6].

Attribute	Value
Family	Cyclone V
Variant	SoC with 3-Gbps Transceivers
Embedded Hard IP	Maximum 2 hard PCIe controllers and 1 hard memory controller
Number Logic Elements	110K
Number Transceiver	9
Transceiver Speed Grade	3.125 Gbps
Package Type	FBGA
Package Pins	896
Operating Temperature	Commercial (0 - 85 deg C)
Additional Notes	Lead Free, Engineering Sample

Table 16.1: FPGA (5CSXFC6D6F31C8NES) Overview [6]

Resource	Quantity
Logic Element (LE)	110K
ALM	41,509
Register	166,036
Memory (M10K)	5,570 Kb
Memory (MLAB)	621 Kb
Variable-precision DSP Block	112
18 x 18 Multiplier	224
FPGA PLL	6
HPS PLL	3
3 Gbps Transceiver	9
FPGA GPIO	228
HPS I/O	181
LVDS (Transmitter)	72
LVDS (Receiver)	72
PCIe Hard IP Block	2
FPGA Hard Memory Controller	1
HPS Hard Memory Controller	1
ARM Cortex-A9 MPCore Processor	Dual-core

Table 16.2: FPGA (5CSXFC6D6F31C8NES) Resource Table [6]

16.2 Object Detection - Matlab Simulation Code

```

1 % Christopher Yarp
2 % Contour Test
3 % Senior Design
4
5 close all;
6 clear; clc;
7
8 %x = imread('eight.tif');
9 thresh = 80;
10 minPts = 10;
11 minLength = 2;
12
13 slackSquareBounding = 0.70;
14 %slackCircle = ;
15 slackCenter = 6;
16
17 slackRatio = 1;
18
19 ratioCenterInner = 3;
20 ratioCenterOuter = 4.25;
21
22 boundingBoxMargin = 5;
23
24 imgNameFormat = './bmp/stillSampleMultiple_%010d.bmp';
25 startImgName = 2;
26 endImgName = 850;
27 %startImgName = 319;
28 %endImgName = 320;
29 imgNameStep = 1;
30 fps = 30/imgNameStep;
31
32 videoFileName = 'tracking.avi';
33
34 M(floor((endImgName-startImgName)/imgNameStep)) = struct('cdata',[],'colormap',[]);
35 vidFrameNum = 1;

```

```

37
38 videoWriter = VideoWriter(videoFileName);
39 open(videoWriter);
40
41 for frameNum = startImgName:imgNameStep:endImgName
42     clear_all_but('thresh', 'minPts', 'minLength', 'slackSquareBounding', ...
43         'slackCenter', 'slackRatio', 'ratioCenterInner', 'ratioCenterOuter', ...
44         'boundingBoxMargin', 'imgNameFormat', 'startImgName', 'endImgName', ...
45         'imgNameStep', 'fps', 'M', 'frameNum', 'vidFrameNum', 'videoFileName', ...
46         'videoWriter');
47
48     fileName = sprintf(imgNameFormat, frameNum);
49     %img = importdata('stillSample_0000000100.bmp');
50     %img = importdata('stillSample_0000000300.bmp');
51     %img = importdata('stillSample_0000000600.bmp');
52     img = importdata(fileName);
53
54     display(['frame: ' fileName]);
55
56     x = rgb2gray(img);
57     fig = figure(1);
58     %figure size from http://www.mathworks.com/matlabcentral/newsreader/view_thread/262957
59     set(fig, 'Position', [0, 100, 1280, 800]);
60
61     subplot(2,2,1);
62     imshow(x);
63     title('Greyscale Image');
64     subplot(2,2,2);
65     imshow(img);
66     title(['Origional Img: ' fileName], 'Interpreter', 'none');
67     %subplot(2,2,4)
68     %h1 = hist(double(x),256);
69     %plot(h1)
70     %h1 = imhist(x,256);
71     %plot(h1)
72
73     subplot(2,2,3);
74     %imcontour(x,[225, 225])
75     c1 = imcontour(x,[thresh, thresh]);
76     title('Contours');
77
78
79     ix = 1;
80     ic = 1;
81     npts = 0;
82     maxX = 0;
83     minX = 0;
84     maxY = 0;
85     minY = 0;
86     numCurvesRec = 1;
87
88     startInd = 1;
89     endInd = 1;
90
91     [c1Rows, c1Cols] = size(c1);
92
93     for i = 1:c1Cols
94         if(i == ix)
95             sizeX = maxX-minX;
96             sizeY = maxY-minY;
97
98             if(sizeX > sizeY)
99                 maxSide = sizeX;
100                minSide = sizeY;
101            else
102                maxSide = sizeY;
103                minSide = sizeX;
104            end

```

```

105
106    if (maxSide>0 && minSide >0)
107        boundingBoxRatio = minSide/maxSide;
108    else
109        boundingBoxRatio = 0;
110    end
111    %display(['minX ' num2str(minX) ', maxX ' num2str(maxX) ', minY ' num2str(
112        minY) 'maxY ' num2str(maxY)]);
113    %display(['sizeX ' num2str(sizeX) ', sizeY ' num2str(sizeY)]);
114    %display(['boundingBoxRatio ' num2str(boundingBoxRatio)]);
115
116    %filter out curves that have two few points or have nonsquare
117    %bounding boxes
118    if( i ~= 1 && npts>=minPts && sizeX > minLength && sizeY > minLength &&
119        boundingBoxRatio >= slackSquareBounding)
120        %record data of prev curve (including maxx, maxy)
121        s = struct('npts', npts, 'minX', minX, 'maxX', maxX, 'minY', minY, 'maxY',
122            'startInd', startInd, 'endInd', endInd);
123        curves(numCurvesRec) = s;
124        %display(['contour ' num2str(ic) ', points = ' num2str(npts) ', (['
125            num2str(minX) ', ' num2str(maxX) '], [ ' num2str(minY) ', ' num2str(
126                maxY), '])']);
127        numCurvesRec = numCurvesRec+1;
128    end
129
130    minX = c1(1, i+1);
131    maxX = c1(1, i+1);
132
133    minY = c1(2, i+1);
134    maxY = c1(2, i+1);
135
136    npts = c1(2, i);
137    startInd = ix+1;
138    ix = ix + npts+1;
139    endInd = ix-1;
140    ic = ic+1;
141
142    if(c1(1, i) < minX)
143        minX = c1(1, i);
144    end
145    if(c1(1, i) > maxX)
146        maxX = c1(1, i);
147    end
148
149    if(c1(2, i) < minY)
150        minY = c1(2, i);
151    end
152    if(c1(2, i) > maxY)
153        maxY = c1(2, i);
154    end
155
156    %record last curve
157    sizeX = maxX-minX;
158    sizeY = maxY-minY;
159
160    if(sizeX > sizeY)
161        maxSide = sizeX;
162        minSide = sizeY;
163    else
164        maxSide = sizeY;
165        minSide = sizeX;
166    end
167
168    if(maxSide>0 && minSide >0)
169        boundingBoxRatio = minSide/maxSide;
170    else
171        boundingBoxRatio = 0;
172    end

```

```

168
169 if (~isempty(c1) && npts >= minPts && sizeX > minLength && sizeY > minLength &&
170   boundingBoxRatio >= slackSquareBounding)
171 %record data of prev curve (including maxx, maxy)
172 s = struct('npts', npts, 'minX', minX, 'maxX', maxX, 'minY', minY, 'maxY', maxY, 'startInd', startInd, 'endInd', endInd);
173 curves(numCurvesRec) = s;
174 display(['contour ' num2str(ic), ' points = ', num2str(npts), ', [', num2str(minX),
175   ', ', num2str(maxX), '], [', num2str(minY), ', ', num2str(maxY), ']]']);
176 end
177
178 display(['num contours: ', num2str(ic)]);
179 display(['num potential contours: ', num2str(numCurvesRec)]);
180
181 %check center points (this is n^2 -> not good!!!!)
182 used = zeros(length(curves));
183
184 foundCount = 0;
185
186 for i = 1:length(curves)
187   if(used(i)==0)
188     centerX = (curves(i).maxX + curves(i).minX)/2;
189     centerY = (curves(i).maxY + curves(i).minY)/2;
190     %display(['(', num2str(centerX), ', ', num2str(centerY), ')']);
191
192     numConcentric = 1;
193     concentric(1) = curves(i);
194
195     centerCircleDim = ((curves(i).maxX - curves(i).minX)+(curves(i).maxY + curves(i).minY))/2;
196
197     for j = i+1:length(curves)
198       centerXnew = (curves(j).maxX + curves(j).minX)/2;
199       centerYnew = (curves(j).maxY + curves(j).minY)/2;
200
201       %check if curve already put in bin and if centers are same
202       if(used(j) == 0 && abs(centerX - centerXnew) < slackCenter && abs(centerY
203         - centerYnew) < slackCenter)
204         numConcentric = numConcentric + 1;
205         concentric(numConcentric) = curves(j);
206
207         if centerCircleDim > ((curves(j).maxX - curves(j).minX)+(curves(j).
208           maxY - curves(j).minY))/2;
209           centerCircleDim = ((curves(j).maxX - curves(j).minX)+(curves(j).
210             maxY - curves(j).minY))/2;
211           center = curves(j);
212         end
213         used(j) = 1;
214       end
215     end
216   end
217
218   % if more than 3 possibilities , we may have the target , we now need to
219   % check ratios of bounding boxes
220   if(numConcentric >= 3)
221     display(['possibly found one at (' num2str(centerX) ', ', num2str(centerY),
222       ')']);
223     %candidateGroups = candidateGroups +1;
224     %candidateGroup(candidateGroups) = concentric;
225
226     %now look at ratios -> may be overkill for our application
227     foundInner = 0;
228     foundOuter = 0;
229     for k = 1:length(concentric)
230       newCircleDim = ((concentric(k).maxX - concentric(k).minX)+(concentric(k).
231         .maxY - concentric(k).minY))/2;
232       if abs(newCircleDim/centerCircleDim - ratioCenterInner) <= slackRatio
233         foundInner = 1;
234         inner = concentric(k);

```

```

227         elseif abs(newCircleDim/centerCircleDim - ratioCenterOuter) <=
228             slackRatio
229             foundOuter = 1;
230             outer = concentric(k);
231         end
232     end
233
234     if(foundInner ~=0 && foundOuter ~=0)
235         %found it
236         foundCount = foundCount+1;
237         foundStr = struct('center', center, 'inner', inner, 'outer', outer);
238         found(foundCount) = foundStr;
239         display(['found one at (' num2str(centerX) ', ' num2str(centerY) ')']);
240     end
241 end
242
243 %now plot rings and where we found them
244 subplot(2,2,4)
245 imshow(img)
246 title('Detected Objects');
247 hold on;
248 for l = 1:length(found)
249     centerBegin = found(l).center.startInd;
250     centerEnd = found(l).center.endInd;
251     plot(c1(1, centerBegin:centerEnd), c1(2, centerBegin:centerEnd), 'y', 'LineWidth', 2);
252
253     innerBegin = found(l).inner.startInd;
254     innerEnd = found(l).inner.endInd;
255     plot(c1(1, innerBegin:innerEnd), c1(2, innerBegin:innerEnd), 'g', 'LineWidth', 2);
256
257     outerBegin = found(l).outer.startInd;
258     outerEnd = found(l).outer.endInd;
259     plot(c1(1, outerBegin:outerEnd), c1(2, outerBegin:outerEnd), 'b', 'LineWidth', 2);
260
261 %plot bounding box
262 plot([(found(l).outer minX - boundingBoxMargin), (found(l).outer minX -
263     boundingBoxMargin), (found(l).outer maxX + boundingBoxMargin), (found(l).outer
264     .maxX + boundingBoxMargin), (found(l).outer minX - boundingBoxMargin)], ...
265     [(found(l).outer minY - boundingBoxMargin), (found(l).outer maxY +
266     boundingBoxMargin), (found(l).outer maxY + boundingBoxMargin), (found(l).outer
267     .minY - boundingBoxMargin), (found(l).outer minY - boundingBoxMargin)],
268     'r', 'LineWidth', 4);
269 end
270 hold off;
271
272 M(vidFrameNum) = getframe(fig);
273 writeVideo(videoWriter,M(vidFrameNum));
274
275 vidFrameNum = vidFrameNum+1;
276
277 % Video writer: http://www.mathworks.com/help/matlab/ref/videowriter-class.html
278 close(videoWriter);
279
280 implay(M, fps);

```

Listing 16.1: Contour Matlab Simulation Code

16.3 FPGA Demo - Hardware Description

16.3.1 Directory Structure

```
orion/
└── orion_system.qpf
└── orion_system.qsf
└── orion_system.qsys
└── orion_system.sdc
└── ip
    └── ipTest
        ├── burst_read_master.v
        ├── burst_write_master.v
        ├── custom_master.v
        ├── IPTest_hw.tcl
        ├── latency_aware_read_master.v
        ├── SCU_Logo_Seperate_Small.gif
        ├── slave_template.v
        ├── slave_template_macros.h
        ├── test_ip.v
        ├── test_ip_top.v
        └── write_master.v
```

16.3.2 Qsys System - GUI View (orion_system.qsys)

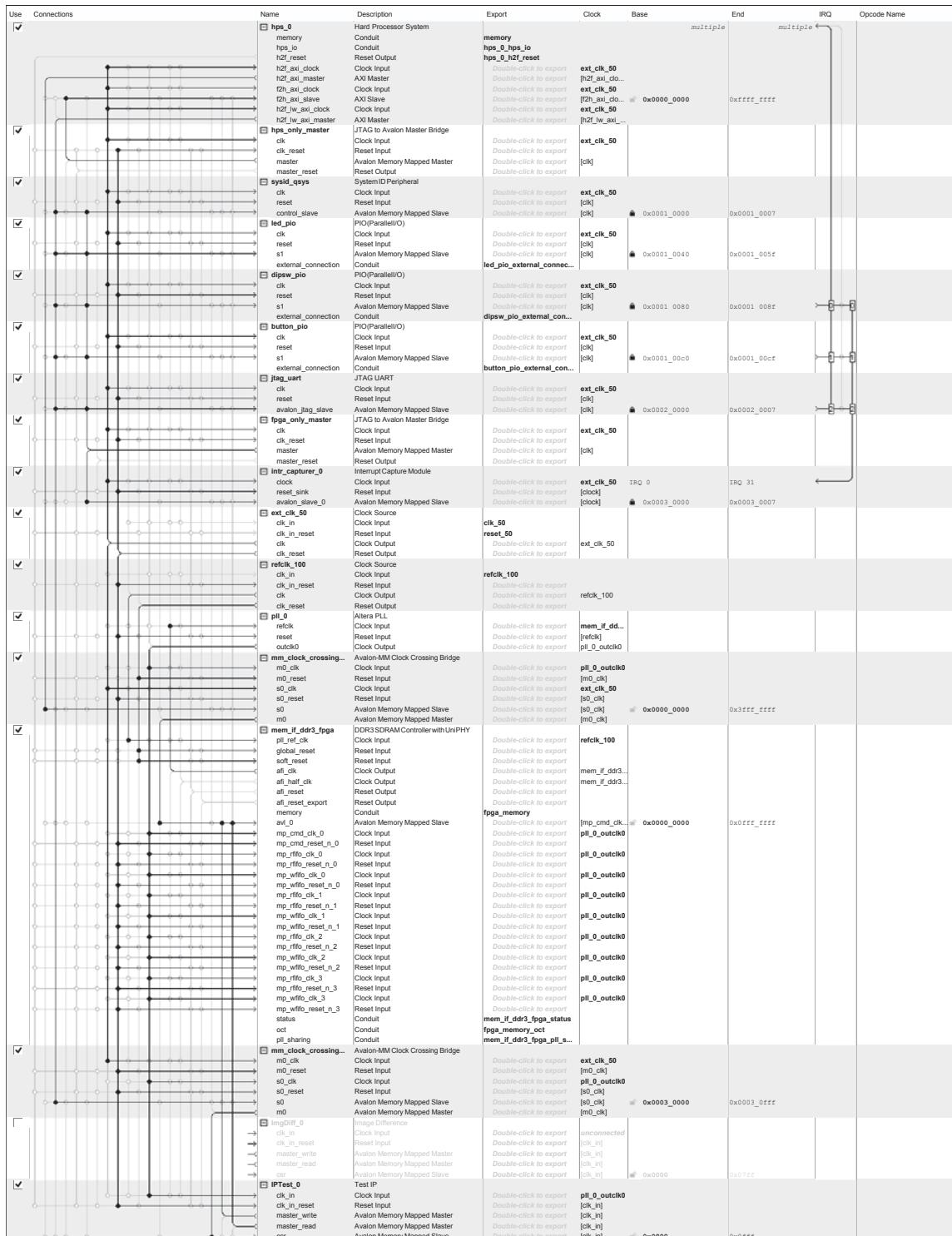


Figure 16.1: Qsys Design - GUI View (orion_system.qsys)

16.3.3 Qsys System - Code View (orion_system.qsys)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <system name="${FILENAME}">
3   <component
4     name="${FILENAME}"
5     displayName="${FILENAME}"
6     version="1.0"
7     description=""
8     tags=""
9     categories="System" />
10   <parameter name="bonusData"><! [CDATA[ bonusData
11   {
12     element ${FILENAME}
13     {
14     }
15     element IPTTest_0
16     {
17       datum _sortIndex
18       {
19         value = "16";
20         type = "int";
21       }
22     }
23     element ImgDiff_0
24     {
25       datum _sortIndex
26       {
27         value = "15";
28         type = "int";
29       }
30     }
31     element jtag_uart.avalon_jtag_slave
32     {
33       datum _lockedAddress
34       {
35         value = "1";
36         type = "boolean";
37       }
38       datum baseAddress
39       {
40         value = "131072";
41         type = "String";
42       }
43     }
44     element intr_capturer_0.avalon_slave_0
45     {
46       datum _lockedAddress
47       {
48         value = "1";
49         type = "boolean";
50       }
51       datum baseAddress
52       {
53         value = "196608";
54         type = "String";
55       }
56     }
57     element mem_if_ddr3_fpga.avl_0
58     {
59       datum baseAddress
60       {
61         value = "0";
62         type = "String";
63       }
64     }
65     element button_pio
66     {
```

```

67     datum _sortIndex
68     {
69         value = "5";
70         type = "int";
71     }
72     datum sopceditor_expanded
73     {
74         value = "1";
75         type = "boolean";
76     }
77 }
78 element sysid_qsys.control_slave
79 {
80     datum _lockedAddress
81     {
82         value = "1";
83         type = "boolean";
84     }
85     datum baseAddress
86     {
87         value = "65536";
88         type = "String";
89     }
90 }
91 element IPTest_0.csr
92 {
93     datum baseAddress
94     {
95         value = "2048";
96         type = "String";
97     }
98 }
99 element ImgDiff_0.csr
100 {
101     datum baseAddress
102     {
103         value = "0";
104         type = "String";
105     }
106 }
107 element dipsw_pio
108 {
109     datum _sortIndex
110     {
111         value = "4";
112         type = "int";
113     }
114 }
115 element ext_clk_50
116 {
117     datum _sortIndex
118     {
119         value = "9";
120         type = "int";
121     }
122 }
123 element hps_0.f2h_axis_slave
124 {
125     datum _lockedAddress
126     {
127         value = "0";
128         type = "boolean";
129     }
130     datum baseAddress
131     {
132         value = "0";
133         type = "String";
134     }

```

```

135 }
136 element fpga_only_master
137 {
138     datum _sortIndex
139     {
140         value = "7";
141         type = "int";
142     }
143 }
144 element hps_0
145 {
146     datum _sortIndex
147     {
148         value = "0";
149         type = "int";
150     }
151 }
152 element hps_only_master
153 {
154     datum _sortIndex
155     {
156         value = "1";
157         type = "int";
158     }
159 }
160 element intr_capturer_0
161 {
162     datum _sortIndex
163     {
164         value = "8";
165         type = "int";
166     }
167 }
168 element jtag_uart_irq
169 {
170     datum _tags
171     {
172         value = "";
173         type = "String";
174     }
175 }
176 element jtag_uart
177 {
178     datum _sortIndex
179     {
180         value = "6";
181         type = "int";
182     }
183 }
184 element led_pio
185 {
186     datum _sortIndex
187     {
188         value = "3";
189         type = "int";
190     }
191 }
192 element mem_if_ddr3_fpga
193 {
194     datum _sortIndex
195     {
196         value = "13";
197         type = "int";
198     }
199 }
200 element mm_clock_crossing_bridge_0
201 {
202     datum _sortIndex

```

```

203    {
204        value = "12";
205        type = "int";
206    }
207}
element mm_clock_crossing_bridge_1
{
208    datum _sortIndex
209    {
210        value = "14";
211        type = "int";
212    }
213}
element p11_0
{
214    datum _sortIndex
215    {
216        value = "11";
217        type = "int";
218    }
219}
element refclk_100
{
220    datum _sortIndex
221    {
222        value = "10";
223        type = "int";
224    }
225}
element jtag_uart.reset
{
226    datum _tags
227    {
228        value = "";
229        type = "String";
230    }
231}
element mm_clock_crossing_bridge_1.s0
{
232    datum baseAddress
233    {
234        value = "196608";
235        type = "String";
236    }
237}
element mm_clock_crossing_bridge_0.s0
{
238    datum baseAddress
239    {
240        value = "0";
241        type = "String";
242    }
243}
element dipsw_pio.s1
{
244    datum _lockedAddress
245    {
246        value = "1";
247        type = "boolean";
248    }
249    datum baseAddress
250    {
251        value = "65664";
252        type = "String";
253    }
254}
element led_pio.s1
{
255}

```

```

271     datum _lockedAddress
272     {
273         value = "1";
274         type = "boolean";
275     }
276     datum baseAddress
277     {
278         value = "65600";
279         type = "String";
280     }
281 }
282 element button_pio.s1
283 {
284     datum _lockedAddress
285     {
286         value = "1";
287         type = "boolean";
288     }
289     datum baseAddress
290     {
291         value = "65728";
292         type = "String";
293     }
294 }
295 element sysid_qsys
296 {
297     datum _sortIndex
298     {
299         value = "2";
300         type = "int";
301     }
302 }
303 ]
304 ]]</parameter>
305 <parameter name="clockCrossingAdapter" value="FIFO" />
306 <parameter name="device" value="5CSXFC6D6F31C8ES" />
307 <parameter name="deviceFamily" value="Cyclone V" />
308 <parameter name="deviceSpeedGrade" value="8_H6" />
309 <parameter name="fabricMode" value="QSYS" />
310 <parameter name="generateLegacySim" value="false" />
311 <parameter name="generationId" value="0" />
312 <parameter name="globalResetBus" value="false" />
313 <parameter name="hdlLanguage" value="VERILOG" />
314 <parameter name="maxAdditionalLatency" value="4" />
315 <parameter name="projectName" value="orion_system.qpf" />
316 <parameter name="socpBorderPoints" value="false" />
317 <parameter name="systemHash" value="1" />
318 <parameter name="timeStamp" value="1399677421041" />
319 <parameter name="useTestBenchNamingPattern" value="false" />
320 <instanceScript></instanceScript>
321 <interface name="memory" internal="hps_0.memory" type="conduit" dir="end" />
322 <interface name="hps_0.hps_io" internal="hps_0.hps_io" type="conduit" dir="end" />
323 <interface name="clk_50" internal="ext_clk_50.clk_in" type="clock" dir="end" />
324 <interface
325     name="led_pio_external_connection"
326     internal="led_pio.external_connection"
327     type="conduit"
328     dir="end" />
329 <interface
330     name="dipsw_pio_external_connection"
331     internal="dipsw_pio.external_connection"
332     type="conduit"
333     dir="end" />
334 <interface
335     name="button_pio_external_connection"
336     internal="button_pio.external_connection"
337     type="conduit"
338     dir="end" />

```

```

339 <interface
340   name="hps_0_h2f_reset"
341   internal="hps_0.h2f_reset"
342   type="reset"
343   dir="start" />
344 <interface
345   name="fpga_memory"
346   internal="mem_if_ddr3_fpga.memory"
347   type="conduit"
348   dir="end" />
349 <interface
350   name="fpga_memory_oct"
351   internal="mem_if_ddr3_fpga.oct"
352   type="conduit"
353   dir="end" />
354 <interface name="refclk_100" internal="refclk_100.clk_in" type="clock" dir="end" />
355 <interface
356   name="reset_50"
357   internal="ext_clk_50.clk_in_reset"
358   type="reset"
359   dir="end" />
360 <interface
361   name="mem_if_ddr3_fpga_status"
362   internal="mem_if_ddr3_fpga.status"
363   type="conduit"
364   dir="end" />
365 <interface
366   name="mem_if_ddr3_fpga_pll_sharing"
367   internal="mem_if_ddr3_fpga pll_sharing"
368   type="conduit"
369   dir="end" />
370 <module
371   kind="altera_avalon_sysid_qsys"
372   version="13.0"
373   enabled="1"
374   name="sysid_qsys">
375   <parameter name="id" value="-1395322102" />
376   <parameter name="timestamp" value="0" />
377   <parameter name="AUTO_CLK_CLOCK_RATE" value="50000000" />
378   <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
379 </module>
380 <module kind="altera_hps" version="13.0.1" enabled="1" name="hps_0">
381   <parameter name="MEM_VENDOR" value="JEDEC" />
382   <parameter name="MEM_FORMAT" value="DISCRETE" />
383   <parameter name="RDIMM_CONFIG" value="0000000000000000" />
384   <parameter name="LRDIMM_EXTENDED_CONFIG">0x0000000000000000</parameter>
385   <parameter name="DISCRETE_FLY_BY" value="true" />
386   <parameter name="DEVICE_DEPTH" value="1" />
387   <parameter name="MEM_MIRROR_ADDRESSING" value="0" />
388   <parameter name="MEM_CLK_FREQ_MAX" value="800.0" />
389   <parameter name="MEM_ROW_ADDR_WIDTH" value="15" />
390   <parameter name="MEM_COL_ADDR_WIDTH" value="10" />
391   <parameter name="MEM_DQ_WIDTH" value="40" />
392   <parameter name="MEM_DQ_PER_DQS" value="8" />
393   <parameter name="MEM_BANKADDR_WIDTH" value="3" />
394   <parameter name="MEM_IF_DM_PINS_EN" value="true" />
395   <parameter name="MEM_IF_DQSN_EN" value="true" />
396   <parameter name="MEM_NUMBER_OF_DIMMS" value="1" />
397   <parameter name="MEM_NUMBER_OF_RANKS_PER_DIMM" value="1" />
398   <parameter name="MEM_NUMBER_OF_RANKS_PER_DEVICE" value="1" />
399   <parameter name="MEM_RANK_MULTIPLICATION_FACTOR" value="1" />
400   <parameter name="MEM_CK_WIDTH" value="1" />
401   <parameter name="MEM_CS_WIDTH" value="1" />
402   <parameter name="MEM_CLK_EN_WIDTH" value="1" />
403   <parameter name="ALTMEMPHY_COMPATIBLE_MODE" value="false" />
404   <parameter name="NEXTGEN" value="true" />
405   <parameter name="MEM_IF_BOARD_BASE_DELAY" value="10" />
406   <parameter name="MEM_IF_SIM_VALID_WINDOW" value="0" />

```

```

407 <parameter name="MEM_GUARANTEED_WRITE_INIT" value="false" />
408 <parameter name="MEM_VERBOSE" value="true" />
409 <parameter name="PINGPONGPHY_EN" value="false" />
410 <parameter name="REFRESH_BURST_VALIDATION" value="false" />
411 <parameter name="MEM_BL" value="OTF" />
412 <parameter name="MEM_BT" value="Sequential" />
413 <parameter name="MEM_LASR" value="Manual" />
414 <parameter name="MEM_SRT" value="Normal" />
415 <parameter name="MEM_PD" value="DLL off" />
416 <parameter name="MEM_DRV_STR" value="RZQ/6" />
417 <parameter name="MEM_DLL_EN" value="true" />
418 <parameter name="MEM_RTT_NOM" value="RZQ/6" />
419 <parameter name="MEM_RTT_WR" value="Dynamic ODT off" />
420 <parameter name="MEM_WTCL" value="6" />
421 <parameter name="MEM_ATCL" value="Disabled" />
422 <parameter name="MEM_TCL" value="7" />
423 <parameter name="MEM_AUTO_LEVELING_MODE" value="true" />
424 <parameter name="MEM_USER_LEVELING_MODE" value="Leveling" />
425 <parameter name="MEM_INIT_EN" value="false" />
426 <parameter name="MEM_INIT_FILE" value="" />
427 <parameter name="DAT_DATA_WIDTH" value="32" />
428 <parameter name="TIMING_TIS" value="170" />
429 <parameter name="TIMING_TIH" value="120" />
430 <parameter name="TIMING_TDS" value="10" />
431 <parameter name="TIMING_TDHS" value="45" />
432 <parameter name="TIMING_TDQSQ" value="100" />
433 <parameter name="TIMING_TQHS" value="300" />
434 <parameter name="TIMING_TQH" value="0.38" />
435 <parameter name="TIMING_TDQSCK" value="225" />
436 <parameter name="TIMING_TDQSCKDS" value="450" />
437 <parameter name="TIMING_TDQSCKDM" value="900" />
438 <parameter name="TIMING_TDQSCKDL" value="1200" />
439 <parameter name="TIMING_TDQSS" value="0.27" />
440 <parameter name="TIMING_TDQSH" value="0.35" />
441 <parameter name="TIMING_TQSH" value="0.4" />
442 <parameter name="TIMING_TDSH" value="0.18" />
443 <parameter name="TIMING_TDSS" value="0.18" />
444 <parameter name="MEM_TINIT_NS" value="500" />
445 <parameter name="MEM_TMRD_NS" value="4" />
446 <parameter name="MEM_TRAS_NS" value="35.0" />
447 <parameter name="MEM_TRCD_NS" value="13.75" />
448 <parameter name="MEM_TRP_NS" value="13.75" />
449 <parameter name="MEM_TREFL_NS" value="7.8" />
450 <parameter name="MEM_TRFC_NS" value="260.0" />
451 <parameter name="CFG_TCCD_NS" value="2.5" />
452 <parameter name="MEM_TWR_NS" value="15.0" />
453 <parameter name="MEM_TWIR" value="4" />
454 <parameter name="MEM_TFAW_NS" value="30.0" />
455 <parameter name="MEM_TRRD_NS" value="10.0" />
456 <parameter name="MEM_TRTP_NS" value="10.0" />
457 <parameter name="POWER_OF_TWO_BUS" value="false" />
458 <parameter name="SOPC_COMPAT_RESET" value="false" />
459 <parameter name="AVL_MAX_SIZE" value="4" />
460 <parameter name="BYTE_ENABLE" value="true" />
461 <parameter name="ENABLE_CTRL_AVALON_INTERFACE" value="true" />
462 <parameter name="CTL_DEEP_POWERDN_EN" value="false" />
463 <parameter name="CTL_SELF_REFRESH_EN" value="false" />
464 <parameter name="AUTO_POWERDN_EN" value="false" />
465 <parameter name="AUTO_PD_CYCLES" value="0" />
466 <parameter name="CTL_USR_REFRESH_EN" value="false" />
467 <parameter name="CTL_AUTOPCH_EN" value="false" />
468 <parameter name="CTL_ZQCAL_EN" value="false" />
469 <parameter name="ADDR_ORDER" value="0" />
470 <parameter name="CTL_LOOK_AHEAD_DEPTH" value="4" />
471 <parameter name="CONTROLLER_LATENCY" value="5" />
472 <parameter name="CFG_REORDER_DATA" value="true" />
473 <parameter name="STARVE_LIMIT" value="10" />
474 <parameter name="CTL_CSR_ENABLED" value="false" />
```

```

475 <parameter name="CTL_CSR_CONNECTION" value="INTERNAL_JTAG" />
476 <parameter name="CTL_ECC_ENABLED" value="false" />
477 <parameter name="CTL_HRB_ENABLED" value="false" />
478 <parameter name="CTL_ECC_AUTO_CORRECTION_ENABLED" value="false" />
479 <parameter name="MULTICAST_EN" value="false" />
480 <parameter name="CTL_DYNAMIC_BANK_ALLOCATION" value="false" />
481 <parameter name="CTL_DYNAMIC_BANK_NUM" value="4" />
482 <parameter name="DEBUG_MODE" value="false" />
483 <parameter name="ENABLE_BURST_MERGE" value="false" />
484 <parameter name="CTL_ENABLE_BURST_INTERRUPT" value="true" />
485 <parameter name="CTL_ENABLE_BURST_TERMINATE" value="true" />
486 <parameter name="LOCAL_ID_WIDTH" value="8" />
487 <parameter name="WRBUFFER_ADDR_WIDTH" value="6" />
488 <parameter name="MAX_PENDING_WR_CMD" value="8" />
489 <parameter name="MAX_PENDING_RD_CMD" value="16" />
490 <parameter name="USE_MM_ADAPTOR" value="true" />
491 <parameter name="USE_AXI_ADAPTOR" value="false" />
492 <parameter name="HCX_COMPAT_MODE" value="false" />
493 <parameter name="CTL_CMD_QUEUE_DEPTH" value="8" />
494 <parameter name="CTL_CSR_READ_ONLY" value="1" />
495 <parameter name="CFG_DATA_REORDERING_TYPE" value="INTER_BANK" />
496 <parameter name="NUM_OF_PORTS" value="1" />
497 <parameter name="ENABLE_BONDING" value="false" />
498 <parameter name="ENABLE_USER_ECC" value="false" />
499 <parameter name="AVL_DATA_WIDTH_PORT" value="32,32,32,32,32,32" />
500 <parameter name="PRIORITY_PORT" value="1,1,1,1,1,1" />
501 <parameter name="WEIGHT_PORT" value="0,0,0,0,0,0" />
502 <parameter name="CPOR_TYPE_PORT">Bidirectional , Bidirectional , Bidirectional ,
      Bidirectional , Bidirectional , Bidirectional</parameter>
503 <parameter name="ENABLE_EMIT_BFM_MASTER" value="false" />
504 <parameter name="FORCE_SEQUENCER_TCL_DEBUG_MODE" value="false" />
505 <parameter name="ENABLE_SEQUENCER_MARGINING_ON_BY_DEFAULT" value="false" />
506 <parameter name="REF_CLK_FREQ" value="25.0" />
507 <parameter name="REF_CLK_FREQ_PARAM_VALID" value="false" />
508 <parameter name="REF_CLK_FREQ_MIN_PARAM" value="0.0" />
509 <parameter name="REF_CLK_FREQ_MAX_PARAM" value="0.0" />
510 <parameter name="PLL_DR_CLK_FREQ_PARAM" value="0.0" />
511 <parameter name="PLL_DR_CLK_FREQ_SIM_STR_PARAM" value="" />
512 <parameter name="PLL_DR_CLK_PHASE_PS_PARAM" value="0" />
513 <parameter name="PLL_DR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
514 <parameter name="PLL_DR_CLK_MULT_PARAM" value="0" />
515 <parameter name="PLL_DR_CLK_DIV_PARAM" value="0" />
516 <parameter name="PLL_MEM_CLK_FREQ_PARAM" value="0.0" />
517 <parameter name="PLL_MEM_CLK_FREQ_SIM_STR_PARAM" value="" />
518 <parameter name="PLL_MEM_CLK_PHASE_PS_PARAM" value="0" />
519 <parameter name="PLL_MEM_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
520 <parameter name="PLL_MEM_CLK_MULT_PARAM" value="0" />
521 <parameter name="PLL_MEM_CLK_DIV_PARAM" value="0" />
522 <parameter name="PLL_AFI_CLK_FREQ_PARAM" value="0.0" />
523 <parameter name="PLL_AFI_CLK_FREQ_SIM_STR_PARAM" value="" />
524 <parameter name="PLL_AFI_CLK_PHASE_PS_PARAM" value="0" />
525 <parameter name="PLL_AFI_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
526 <parameter name="PLL_AFI_CLK_MULT_PARAM" value="0" />
527 <parameter name="PLL_AFI_CLK_DIV_PARAM" value="0" />
528 <parameter name="PLL_WRITE_CLK_FREQ_PARAM" value="0.0" />
529 <parameter name="PLL_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
530 <parameter name="PLL_WRITE_CLK_PHASE_PS_PARAM" value="0" />
531 <parameter name="PLL_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
532 <parameter name="PLL_WRITE_CLK_MULT_PARAM" value="0" />
533 <parameter name="PLL_WRITE_CLK_DIV_PARAM" value="0" />
534 <parameter name="PLL_ADDR_CMD_CLK_FREQ_PARAM" value="0.0" />
535 <parameter name="PLL_ADDR_CMD_CLK_FREQ_SIM_STR_PARAM" value="" />
536 <parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_PARAM" value="0" />
537 <parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
538 <parameter name="PLL_ADDR_CMD_CLK_MULT_PARAM" value="0" />
539 <parameter name="PLL_ADDR_CMD_CLK_DIV_PARAM" value="0" />
540 <parameter name="PLL_AFI_HALF_CLK_FREQ_PARAM" value="0.0" />
541 <parameter name="PLL_AFI_HALF_CLK_FREQ_SIM_STR_PARAM" value="" />
```

```

542 <parameter name="PLL_AFI_HALF_CLK_PHASE_PS_PARAM" value="0" />
543 <parameter name="PLL_AFI_HALF_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
544 <parameter name="PLL_AFI_HALF_CLK_MULT_PARAM" value="0" />
545 <parameter name="PLL_AFI_HALF_CLK_DIV_PARAM" value="0" />
546 <parameter name="PLL_NIOS_CLK_FREQ_PARAM" value="0.0" />
547 <parameter name="PLL_NIOS_CLK_FREQ_SIM_STR_PARAM" value="" />
548 <parameter name="PLL_NIOS_CLK_PHASE_PS_PARAM" value="0" />
549 <parameter name="PLL_NIOS_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
550 <parameter name="PLL_NIOS_CLK_MULT_PARAM" value="0" />
551 <parameter name="PLL_NIOS_CLK_DIV_PARAM" value="0" />
552 <parameter name="PLL_CONFIG_CLK_FREQ_PARAM" value="0.0" />
553 <parameter name="PLL_CONFIG_CLK_FREQ_SIM_STR_PARAM" value="" />
554 <parameter name="PLL_CONFIG_CLK_PHASE_PS_PARAM" value="0" />
555 <parameter name="PLL_CONFIG_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
556 <parameter name="PLL_CONFIG_CLK_MULT_PARAM" value="0" />
557 <parameter name="PLL_CONFIG_CLK_DIV_PARAM" value="0" />
558 <parameter name="PLL_P2C_READ_CLK_FREQ_PARAM" value="0.0" />
559 <parameter name="PLL_P2C_READ_CLK_FREQ_SIM_STR_PARAM" value="" />
560 <parameter name="PLL_P2C_READ_CLK_PHASE_PS_PARAM" value="0" />
561 <parameter name="PLL_P2C_READ_CLK_MULT_PARAM" value="" />
562 <parameter name="PLL_P2C_READ_CLK_DIV_PARAM" value="0" />
563 <parameter name="PLL_C2P_WRITE_CLK_FREQ_PARAM" value="0.0" />
564 <parameter name="PLL_C2P_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
565 <parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_PARAM" value="0" />
566 <parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
567 <parameter name="PLL_C2P_WRITE_CLK_MULT_PARAM" value="0" />
568 <parameter name="PLL_C2P_WRITE_CLK_DIV_PARAM" value="0" />
569 <parameter name="PLL_HR_CLK_FREQ_PARAM" value="0.0" />
570 <parameter name="PLL_HR_CLK_FREQ_SIM_STR_PARAM" value="" />
571 <parameter name="PLL_HR_CLK_PHASE_PS_PARAM" value="0" />
572 <parameter name="PLL_HR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
573 <parameter name="PLL_HR_CLK_MULT_PARAM" value="0" />
574 <parameter name="PLL_HR_CLK_DIV_PARAM" value="0" />
575 <parameter name="PLL_AFI_PHY_CLK_FREQ_PARAM" value="0.0" />
576 <parameter name="PLL_AFI_PHY_CLK_FREQ_SIM_STR_PARAM" value="" />
577 <parameter name="PLL_AFI_PHY_CLK_PHASE_PS_PARAM" value="0" />
578 <parameter name="PLL_AFI_PHY_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
579 <parameter name="PLL_AFI_PHY_CLK_MULT_PARAM" value="0" />
580 <parameter name="PLL_AFI_PHY_CLK_DIV_PARAM" value="0" />
581 <parameter name="PLL_CLK_PARAM_VALID" value="false" />
582 <parameter name="ENABLE_EXTRA_REPORTING" value="false" />
583 <parameter name="NUM_EXTRA_REPORT_PATH" value="10" />
584 <parameter name="ENABLE_ISS_PROBES" value="false" />
585 <parameter name="CALIB_REG_WIDTH" value="8" />
586 <parameter name="USE_SEQUENCER_BFM" value="false" />
587 <parameter name="DEFAULT_FAST_SIM_MODEL" value="true" />
588 <parameter name="PLL_SHARING_MODE" value="None" />
589 <parameter name="NUM_PLL_SHARING_INTERFACES" value="1" />
590 <parameter name="EXPORT_AFI_HALF_CLK" value="false" />
591 <parameter name="ABSTRACT_REAL_COMPARE_TEST" value="false" />
592 <parameter name="INCLUDE_BOARD_DELAY_MODEL" value="false" />
593 <parameter name="INCLUDE_MULTIRANK_BOARD_DELAY_MODEL" value="false" />
594 <parameter name="USE_FAKE_PHY" value="false" />
595 <parameter name="FORCE_MAX_LATENCY_COUNT_WIDTH" value="0" />
596 <parameter name="ENABLE_NON_DESTRUCTIVE_CALIB" value="false" />
597 <parameter name="TRACKING_ERROR_TEST" value="false" />
598 <parameter name="TRACKING_WATCH_TEST" value="false" />
599 <parameter name="MARGIN_VARIATION_TEST" value="false" />
600 <parameter name="EXTRA_SETTINGS" value="" />
601 <parameter name="MEM_DEVICE" value="MISSING_MODEL" />
602 <parameter name="FORCE_SYNTHESIS_LANGUAGE" value="" />
603 <parameter name="FORCED_NUM_WRITE_FR_CYCLE_SHIFTS" value="0" />
604 <parameter name="SEQUENCER_TYPE" value="NIOS" />
605 <parameter name="ADVERTISE_SEQUENCER_SW_BUILD_FILES" value="false" />
606 <parameter name="FORCED_NON_LDC_ADDR_CMD_MEM_CK_INVERT" value="false" />
607 <parameter name="PHY_ONLY" value="false" />
608 <parameter name="SEQ_MODE" value="0" />
609

```

```

610 <parameter name="ADVANCED.CK.PHASES" value="false" />
611 <parameter name="COMMAND.PHASE" value="0.0" />
612 <parameter name="MEM.CK.PHASE" value="0.0" />
613 <parameter name="P2C_READ_CLOCK_ADD_PHASE" value="0.0" />
614 <parameter name="C2P_WRITE_CLOCK_ADD_PHASE" value="0.0" />
615 <parameter name="ACV_PHY_CLK_ADD.FR.PHASE" value="0.0" />
616 <parameter name="MEM_VOLTAGE" value="1.5V DDR3" />
617 <parameter name="PLL_LOCATION" value="Top_Bottom" />
618 <parameter name="SKIP_MEM_INIT" value="true" />
619 <parameter name="READ.DQ_DQS.CLOCK_SOURCE" value="INVERTED.DQS.BUS" />
620 <parameter name="DQ_INPUT.REG_USE_CLKN" value="false" />
621 <parameter name="DQS_DQSN.MODE" value="DIFFERENTIAL" />
622 <parameter name="AFI_DEBUG_INFO_WIDTH" value="32" />
623 <parameter name="CALIBRATION_MODE" value="Skip" />
624 <parameter name="NIOS.ROM.DATA_WIDTH" value="32" />
625 <parameter name="READ_FIFO_SIZE" value="8" />
626 <parameter name="PHY_CSR_ENABLED" value="false" />
627 <parameter name="PHY_CSR_CONNECTION" value="INTERNAL.JTAG" />
628 <parameter name="USER_DEBUG_LEVEL" value="1" />
629 <parameter name="TIMING.BOARD_DERATE_METHOD" value="AUTO" />
630 <parameter name="TIMING.BOARD.CK_CKN.SLEW_RATE" value="2.0" />
631 <parameter name="TIMING.BOARD.AC_SLEW_RATE" value="1.0" />
632 <parameter name="TIMING.BOARD.DQS_DQSN.SLEW_RATE" value="2.0" />
633 <parameter name="TIMING.BOARD.DQ_SLEW_RATE" value="1.0" />
634 <parameter name="TIMING_BOARD_TIS" value="0.0" />
635 <parameter name="TIMING_BOARD_TIH" value="0.0" />
636 <parameter name="TIMING_BOARD_TDS" value="0.0" />
637 <parameter name="TIMING_BOARD_TDH" value="0.0" />
638 <parameter name="TIMING_BOARD_ISI_METHOD" value="AUTO" />
639 <parameter name="TIMING_BOARD_AC_EYE.REDUCTION_SU" value="0.0" />
640 <parameter name="TIMING_BOARD_AC_EYE.REDUCTION_H" value="0.0" />
641 <parameter name="TIMING_BOARD.DQ_EYE.REDUCTION" value="0.0" />
642 <parameter name="TIMING_BOARD_DELTA_DQS.ARRIVAL_TIME" value="0.0" />
643 <parameter name="PACKAGE_DESKEW" value="false" />
644 <parameter name="AC PACKAGE DESKEW" value="false" />
645 <parameter name="TIMING_BOARD_MAX_CK_DELAY" value="0.6" />
646 <parameter name="TIMING_BOARD_MAX_DQS_DELAY" value="0.6" />
647 <parameter name="TIMING_BOARD_SKew_CKDQS_DIMM_MIN" value="-0.01" />
648 <parameter name="TIMING_BOARD_SKew_CKDQS_DIMM_MAX" value="0.01" />
649 <parameter name="TIMING_BOARD_SKew_BETWEEN_DIMMS" value="0.05" />
650 <parameter name="TIMING_BOARD_SKew_WITHIN_DQS" value="0.02" />
651 <parameter name="TIMING_BOARD_SKew_BETWEEN_DQS" value="0.02" />
652 <parameter name="TIMING_BOARD.DQ_TO_DQS_SKew" value="0.0" />
653 <parameter name="TIMING_BOARD_AC_SKew" value="0.02" />
654 <parameter name="TIMING_BOARD.AC_TO_CK_SKew" value="0.0" />
655 <parameter name="RATE" value="Full" />
656 <parameter name="MEM.CLK_FREQ" value="400.0" />
657 <parameter name="USE_MEM.CLK_FREQ" value="false" />
658 <parameter name="FORCE_DQS.TRACKING" value="AUTO" />
659 <parameter name="FORCE_SHADOW_REGS" value="AUTO" />
660 <parameter name="MRS.MIRROR_PING_PONG_ATSO" value="false" />
661 <parameter name="SYS_INFO.DEVICE_FAMILY" value="Cyclone V" />
662 <parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM_VALID" value="false" />
663 <parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM" value="" />
664 <parameter name="DEVICE_FAMILY_PARAM" value="" />
665 <parameter name="SPEED.GRADE" value="7" />
666 <parameter name="IS_ES_DEVICE" value="false" />
667 <parameter name="DISABLE_CHILD_MESSAGEAGING" value="false" />
668 <parameter name="HARD_EMIF" value="true" />
669 <parameter name="HHP.HPS" value="true" />
670 <parameter name="HHP.HPS.VERIFICATION" value="false" />
671 <parameter name="HHP.HPS.SIMULATION" value="false" />
672 <parameter name="HPS.PROTOCOL" value="DDR3" />
673 <parameter name="CUT_NEW_FAMILY_TIMING" value="true" />
674 <parameter name="ENABLE_EXPORT_SEQ_DEBUG_BRIDGE" value="false" />
675 <parameter name="CORE_DEBUG_CONNECTION" value="EXPORT" />
676 <parameter name="ADD_EXTERNAL_SEQ_DEBUG_NIOS" value="false" />
677 <parameter name="ED_EXPORT_SEQ_DEBUG" value="false" />

```

```

678 <parameter name="ADD EFFICIENCY MONITOR" value="false" />
679 <parameter name="ENABLE_ABS_RAM_MEM_INIT" value="false" />
680 <parameter name="ABS_RAM_MEM_INIT_FILENAME" value="meminit" />
681 <parameter name="DLL_SHARING_MODE" value="None" />
682 <parameter name="NUM_DLL_SHARING_INTERFACES" value="1" />
683 <parameter name="OCT_SHARING_MODE" value="None" />
684 <parameter name="NUM_OCT_SHARING_INTERFACES" value="1" />
685 <parameter name="MPU_EVENTS_Enable" value="false" />
686 <parameter name="GP_Enable" value="false" />
687 <parameter name="DEBUGAPB_Enable" value="false" />
688 <parameter name="STM_Enable" value="false" />
689 <parameter name="CTI_Enable" value="false" />
690 <parameter name="TPIUFPGA_Enable" value="false" />
691 <parameter name="BOOTFROMFPGA_Enable" value="false" />
692 <parameter name="TEST_Enable" value="false" />
693 <parameter name="HLGPI_Enable" value="false" />
694 <parameter name="BSEL_EN" value="false" />
695 <parameter name="BSEL" value="1" />
696 <parameter name="CSEL_EN" value="false" />
697 <parameter name="CSEL" value="0" />
698 <parameter name="F2S_Width" value="2" />
699 <parameter name="S2F_Width" value="2" />
700 <parameter name="LWH2F_Enable" value="true" />
701 <parameter name="F2SDRAM_Type" value="" />
702 <parameter name="F2SDRAM_Width" value="" />
703 <parameter name="S2FCLK_COLDIRST_Enable" value="false" />
704 <parameter name="S2FCLK_PENDINGRST_Enable" value="false" />
705 <parameter name="F2SCLK_DBGRST_Enable" value="false" />
706 <parameter name="F2SCLK_WARMRST_Enable" value="false" />
707 <parameter name="F2SCLK_COLDRST_Enable" value="false" />
708 <parameter name="DMA_Enable">No, No, No, No, No, No, No</parameter>
709 <parameter name="F2S_INTERRUPT_Enable" value="true" />
710 <parameter name="S2F_INTERRUPT_CAN_Enable" value="false" />
711 <parameter name="S2F_INTERRUPT_CLOCKPERIPHERAL_Enable" value="false" />
712 <parameter name="S2F_INTERRUPT_CTL_Enable" value="false" />
713 <parameter name="S2F_INTERRUPT_DMA_Enable" value="false" />
714 <parameter name="S2F_INTERRUPT_EMAC_Enable" value="false" />
715 <parameter name="S2F_INTERRUPT_FPGAMANAGER_Enable" value="false" />
716 <parameter name="S2F_INTERRUPT_GPIO_Enable" value="false" />
717 <parameter name="S2F_INTERRUPT_I2CEMAC_Enable" value="false" />
718 <parameter name="S2F_INTERRUPT_I2CPERIPHERAL_Enable" value="false" />
719 <parameter name="S2F_INTERRUPT_L4TIMER_Enable" value="false" />
720 <parameter name="S2F_INTERRUPT_NAND_Enable" value="false" />
721 <parameter name="S2F_INTERRUPT_OSCTIMER_Enable" value="false" />
722 <parameter name="S2F_INTERRUPT_QSPI_Enable" value="false" />
723 <parameter name="S2F_INTERRUPT_SDMMC_Enable" value="false" />
724 <parameter name="S2F_INTERRUPT_SPIMASTER_Enable" value="false" />
725 <parameter name="S2F_INTERRUPT_SPISLAVE_Enable" value="false" />
726 <parameter name="S2F_INTERRUPT_UART_Enable" value="false" />
727 <parameter name="S2F_INTERRUPT_USB_Enable" value="false" />
728 <parameter name="S2F_INTERRUPT_WATCHDOG_Enable" value="false" />
729 <parameter name="EMAC0_PinMuxing" value="Unused" />
730 <parameter name="EMAC0_Mode" value="N/A" />
731 <parameter name="EMAC1_PinMuxing" value="HPS I/O Set 0" />
732 <parameter name="EMAC1_Mode" value="RGMII" />
733 <parameter name="NAND_PinMuxing" value="Unused" />
734 <parameter name="NAND_Mode" value="N/A" />
735 <parameter name="QSPI_PinMuxing" value="HPS I/O Set 0" />
736 <parameter name="QSPI_Mode" value="1 SS" />
737 <parameter name="SDIO_PinMuxing" value="HPS I/O Set 0" />
738 <parameter name="SDIO_Mode" value="4-bit Data" />
739 <parameter name="USB0_PinMuxing" value="Unused" />
740 <parameter name="USB0_Mode" value="N/A" />
741 <parameter name="USB1_PinMuxing" value="HPS I/O Set 0" />
742 <parameter name="USB1_Mode" value="SDR" />
743 <parameter name="SPIM0_PinMuxing" value="HPS I/O Set 0" />
744 <parameter name="SPIM0_Mode" value="Single Slave Select" />
745 <parameter name="SPIM1_PinMuxing" value="Unused" />

```

```

746  <parameter name="SPIM1_Mode" value="N/A" />
747  <parameter name="SPIS0_PinMuxing" value="Unused" />
748  <parameter name="SPIS0_Mode" value="N/A" />
749  <parameter name="SPIS1_PinMuxing" value="Unused" />
750  <parameter name="SPIS1_Mode" value="N/A" />
751  <parameter name="UART0.PinMuxing" value="HPS I/O Set 2" />
752  <parameter name="UART0_Mode" value="No Flow Control" />
753  <parameter name="UART1_PinMuxing" value="Unused" />
754  <parameter name="UART1_Mode" value="N/A" />
755  <parameter name="I2C0_PinMuxing" value="HPS I/O Set 1" />
756  <parameter name="I2C0_Mode" value="I2C" />
757  <parameter name="I2C1_PinMuxing" value="Unused" />
758  <parameter name="I2C1_Mode" value="N/A" />
759  <parameter name="I2C2_PinMuxing" value="Unused" />
760  <parameter name="I2C2_Mode" value="N/A" />
761  <parameter name="I2C3_PinMuxing" value="Unused" />
762  <parameter name="I2C3_Mode" value="N/A" />
763  <parameter name="CAN0_PinMuxing" value="HPS I/O Set 0" />
764  <parameter name="CAN0_Mode" value="CAN" />
765  <parameter name="CAN1_PinMuxing" value="Unused" />
766  <parameter name="CAN1_Mode" value="N/A" />
767  <parameter name="TRACE_PinMuxing" value="HPS I/O Set 0" />
768  <parameter name="TRACE_Mode" value="HPS" />
769  <parameter name="GPIO_Enable">No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
    No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
    No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
    No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,</
    parameter>
770  <parameter name="LOANIO_Enable">No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
    ,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
    ,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
    ,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No</
    parameter>
771  <parameter name="S2FCLK_USER0CLK_Enable" value="false" />
772  <parameter name="S2FCLK_USER0CLK_FREQ" value="50" />
773  <parameter name="S2FCLK_USER1CLK_Enable" value="false" />
774  <parameter name="S2FCLK_USER1CLK_FREQ" value="100" />
775  <parameter name="S2FCLK_USER2CLK_Enable" value="false" />
776  <parameter name="S2FCLK_USER2CLK_FREQ" value="100" />
777  <parameter name="F2SCLK_PERIPHCLK_Enable" value="false" />
778  <parameter name="F2SCLK_PERIPHCLK_FREQ" value="100" />
779  <parameter name="F2SCLK_SDRAMCLK_Enable" value="false" />
780  <parameter name="F2SCLK_SDRAMCLK_FREQ" value="100" />
781  <parameter name="F2H_AXI_CLOCK_FREQ" value="50000000" />
782  <parameter name="H2F_AXI_CLOCK_FREQ" value="50000000" />
783  <parameter name="H2F_LW_AXI_CLOCK_FREQ" value="50000000" />
784  <parameter name="F2H_SDRAM0_CLOCK_FREQ" value="100" />
785  <parameter name="F2H_SDRAM1_CLOCK_FREQ" value="100" />
786  <parameter name="F2H_SDRAM2_CLOCK_FREQ" value="100" />
787  <parameter name="F2H_SDRAM3_CLOCK_FREQ" value="100" />
788  <parameter name="F2H_SDRAM4_CLOCK_FREQ" value="100" />
789  <parameter name="F2H_SDRAM5_CLOCK_FREQ" value="100" />
790  <parameter name="H2F_CTI_CLOCK_FREQ" value="100" />
791  <parameter name="H2F_TPIU_CLOCK_IN_FREQ" value="100" />
792  <parameter name="H2F_DEBUG_APB_CLOCK_FREQ" value="100" />
793  <parameter
      name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC_PTP_REF_CLOCK"
      value="100" />
794  <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_RX_CLK_IN" value="100" />
795  <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_TX_CLK_IN" value="100" />
796  <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC0_MD_CLK" value="100" />
797  <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC0_GTX_CLK" value="100" />
798  <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_RX_CLK_IN" value="100" />
799  <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_TX_CLK_IN" value="100" />
800  <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC1_MD_CLK" value="100" />
801  <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC1_GTX_CLK" value="100" />
802  <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_QSPI_SCLK_OUT" value="100" />
803  <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SDIO_CLK_IN" value="100" />

```

```

806 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SDIO_CCLK" value="100" />
807 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USB0_CLK_IN" value="100" />
808 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USB1_CLK_IN" value="100" />
809 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SPIM0_SCLK_OUT" value="100" />
810 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SPIM1_SCLK_OUT" value="100" />
811 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SPISO_SCLK_IN" value="100" />
812 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SPIS1_SCLK_IN" value="100" />
813 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C0_SCLK_IN" value="100" />
814 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C0_CLK" value="100" />
815 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C1_SCLK_IN" value="100" />
816 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C1_CLK" value="100" />
817 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C2_SCLK_IN" value="100" />
818 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C2_CLK" value="100" />
819 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C3_SCLK_IN" value="100" />
820 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C3_CLK" value="100" />
821 <parameter name="device_name" value="5CSXFC6D6F31C8ES" />
822 <parameter
823   name="quartus_ini_hps_ip_enable_all_peripheral_fpga_interfaces"
824   value="false" />
825 <parameter
826   name="quartus_ini_hps_ip_enable_emac0_peripheral_fpga_interface"
827   value="false" />
828 <parameter name="quartus_ini_hps_ip_enable_test_interface" value="false" />
829 <parameter name="quartus_ini_hps_ip_fast_f2sdram_sim_model" value="false" />
830 <parameter name="quartus_ini_hps_ip_suppress_sdram_synth" value="false" />
831 <parameter name="quartus_ini_hps_ip_enable_loanio" value="false" />
832 <parameter
833   name="quartus_ini_hps_ip_enable_low_speed_serial_fpga_interfaces"
834   value="false" />
835 <parameter name="quartus_ini_hps_ip_enable_bsel_csel" value="false" />
836 </module>
837 <module
838   kind="altera_jtag_avalon_master"
839   version="13.0"
840   enabled="1"
841   name="fpga_only_master">
842   <parameter name="USE_PLI" value="0" />
843   <parameter name="PLI_PORT" value="50000" />
844   <parameter name="COMPONENT_CLOCK" value="0" />
845   <parameter name="FAST_VER" value="0" />
846   <parameter name="FIFO_DEPTHS" value="2" />
847   <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
848   <parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
849 </module>
850 <module
851   kind="intr_capturer"
852   version="100.99.98.97"
853   enabled="1"
854   name="intr_capturer_0">
855   <parameter name="NUM_INTR" value="32" />
856   <parameter name="AUTO_CLOCK_CLOCK_RATE" value="50000000" />
857 </module>
858 <module kind="clock_source" version="13.0" enabled="1" name="ext_clk_50">
859   <parameter name="clockFrequency" value="50000000" />
860   <parameter name="clockFrequencyKnown" value="true" />
861   <parameter name="inputClockFrequency" value="0" />
862   <parameter name="resetSynchronousEdges" value="NONE" />
863 </module>
864 <module
865   kind="altera_jtag_avalon_master"
866   version="13.0"
867   enabled="1"
868   name="hps_only_master">
869   <parameter name="USE_PLI" value="0" />
870   <parameter name="PLI_PORT" value="50000" />
871   <parameter name="COMPONENT_CLOCK" value="0" />
872   <parameter name="FAST_VER" value="0" />
873   <parameter name="FIFO_DEPTHS" value="2" />

```

```

874 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
875 <parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
876 </module>
877 <module
878   kind="altera_avalon_jtag_uart"
879   version="13.0.1.99.2"
880   enabled="1"
881   name="jtag_uart">
882   <parameter name="allowMultipleConnections" value="true" />
883   <parameter name="hubInstanceID" value="0" />
884   <parameter name="readBufferDepth" value="64" />
885   <parameter name="readIRQThreshold" value="8" />
886   <parameter name="simInputCharacterStream" value="" />
887   <parameter name="simInteractiveOptions">INTERACTIVE_ASCII_OUTPUT</parameter>
888   <parameter name="useRegistersForReadBuffer" value="false" />
889   <parameter name="useRegistersForWriteBuffer" value="false" />
890   <parameter name="useRelativePathForSimFile" value="false" />
891   <parameter name="writeBufferDepth" value="64" />
892   <parameter name="writeIRQThreshold" value="8" />
893   <parameter name="avalonSpec" value="2.0" />
894 </module>
895 <module
896   kind="altera_avalon_pio"
897   version="13.0.1.99.2"
898   enabled="1"
899   name="button_pio">
900   <parameter name="bitClearingEdgeCapReg" value="true" />
901   <parameter name="bitModifyingOutReg" value="false" />
902   <parameter name="captureEdge" value="true" />
903   <parameter name="direction" value="Input" />
904   <parameter name="edgeType" value="FALLING" />
905   <parameter name="generateIRQ" value="true" />
906   <parameter name="irqType" value="EDGE" />
907   <parameter name="resetValue" value="0" />
908   <parameter name="simDoTestBenchWiring" value="false" />
909   <parameter name="simDrivenValue" value="0" />
910   <parameter name="width" value="2" />
911   <parameter name="clockRate" value="50000000" />
912 </module>
913 <module
914   kind="altera_avalon_pio"
915   version="13.0.1.99.2"
916   enabled="1"
917   name="dipsw_pio">
918   <parameter name="bitClearingEdgeCapReg" value="true" />
919   <parameter name="bitModifyingOutReg" value="false" />
920   <parameter name="captureEdge" value="true" />
921   <parameter name="direction" value="Input" />
922   <parameter name="edgeType" value="ANY" />
923   <parameter name="generateIRQ" value="true" />
924   <parameter name="irqType" value="EDGE" />
925   <parameter name="resetValue" value="0" />
926   <parameter name="simDoTestBenchWiring" value="false" />
927   <parameter name="simDrivenValue" value="0" />
928   <parameter name="width" value="4" />
929   <parameter name="clockRate" value="50000000" />
930 </module>
931 <module
932   kind="altera_avalon_pio"
933   version="13.0.1.99.2"
934   enabled="1"
935   name="led_pio">
936   <parameter name="bitClearingEdgeCapReg" value="false" />
937   <parameter name="bitModifyingOutReg" value="true" />
938   <parameter name="captureEdge" value="false" />
939   <parameter name="direction" value="InOut" />
940   <parameter name="edgeType" value="RISING" />
941   <parameter name="generateIRQ" value="false" />

```

```

942 <parameter name="irqType" value="LEVEL" />
943 <parameter name="resetValue" value="0" />
944 <parameter name="simDoTestBenchWiring" value="false" />
945 <parameter name="simDrivenValue" value="0" />
946 <parameter name="width" value="4" />
947 <parameter name="clockRate" value="50000000" />
948 </module>
949 <module
950   kind="altera_mem_if_ddr3_emif"
951   version="13.0"
952   enabled="1"
953   name="mem_if_ddr3_fpga">
954   <parameter name="MEM_VENDOR" value="Micron" />
955   <parameter name="MEM_FORMAT" value="DISCRETE" />
956   <parameter name="RDIMM_CONFIG" value="0000000000000000" />
957   <parameter name="LRDIMM_EXTENDED_CONFIG">0x0000000000000000</parameter>
958   <parameter name="DISCRETE_FLY_BY" value="true" />
959   <parameter name="DEVICE_DEPTH" value="1" />
960   <parameter name="MEM_MIRROR_ADDRESSING" value="0" />
961   <parameter name="MEM_CLK_FREQ_MAX" value="800.0" />
962   <parameter name="MEM_ROW_ADDR_WIDTH" value="15" />
963   <parameter name="MEM_COL_ADDR_WIDTH" value="8" />
964   <parameter name="MEM_DQ_WIDTH" value="32" />
965   <parameter name="MEM_DQ_PER_DQS" value="8" />
966   <parameter name="MEM_BANKADDR_WIDTH" value="3" />
967   <parameter name="MEM_IF_DM_PINS_EN" value="true" />
968   <parameter name="MEM_IF_DQSN_EN" value="true" />
969   <parameter name="MEM_NUMBER_OF_DIMMS" value="1" />
970   <parameter name="MEM_NUMBER_OF_RANKS_PER_DIMM" value="1" />
971   <parameter name="MEM_NUMBER_OF_RANKS_PER_DEVICE" value="1" />
972   <parameter name="MEM_RANK_MULTIPLICATION_FACTOR" value="1" />
973   <parameter name="MEM_CK_WIDTH" value="1" />
974   <parameter name="MEM_CS_WIDTH" value="1" />
975   <parameter name="MEM_CLK_EN_WIDTH" value="1" />
976   <parameter name="ALTMEMPHY_COMPATIBLE_MODE" value="false" />
977   <parameter name="NEXTGEN" value="true" />
978   <parameter name="MEM_IF_BOARD_BASE_DELAY" value="10" />
979   <parameter name="MEM_IF_SIM_VALID_WINDOW" value="0" />
980   <parameter name="MEM_GUARANTEED_WRITE_INIT" value="false" />
981   <parameter name="MEM_VERBOSE" value="true" />
982   <parameter name="PINGPONGPHY_EN" value="false" />
983   <parameter name="REFRESH_BURST_VALIDATION" value="false" />
984   <parameter name="MEM_BL" value="OTF" />
985   <parameter name="MEM_BT" value="Sequential" />
986   <parameter name="MEMASR" value="Manual" />
987   <parameter name="MEM_SRT" value="Normal" />
988   <parameter name="MEMLPD" value="DLL off" />
989   <parameter name="MEM_DRV_STR" value="RZQ/6" />
990   <parameter name="MEM_DLL_EN" value="true" />
991   <parameter name="MEM_RTT_NOM" value="RZQ/6" />
992   <parameter name="MEM_RTT_WR" value="RZQ/4" />
993   <parameter name="MEM_WTCL" value="5" />
994   <parameter name="MEM_ATCL" value="Disabled" />
995   <parameter name="MEM_TCL" value="6" />
996   <parameter name="MEM_AUTO_LEVELING_MODE" value="true" />
997   <parameter name="MEM_USER_LEVELING_MODE" value="Leveling" />
998   <parameter name="MEM_INIT_EN" value="false" />
999   <parameter name="MEM_INIT_FILE" value="" />
1000  <parameter name="DAT_DATA_WIDTH" value="32" />
1001  <parameter name="TIMING_TIS" value="350" />
1002  <parameter name="TIMING_TIH" value="275" />
1003  <parameter name="TIMING_TDS" value="125" />
1004  <parameter name="TIMING_TDH" value="150" />
1005  <parameter name="TIMING_TDQSQ" value="200" />
1006  <parameter name="TIMING_TQH" value="0.38" />
1007  <parameter name="TIMING_TDQSCK" value="400" />
1008  <parameter name="TIMING_TDQSCKDS" value="450" />
1009  <parameter name="TIMING_TDQSCKDM" value="900" />
```

```

1010 <parameter name="TIMING_TDQSCKDL" value="1200" />
1011 <parameter name="TIMING_TDQSS" value="0.25" />
1012 <parameter name="TIMING_TQSH" value="0.38" />
1013 <parameter name="TIMING_TDSH" value="0.2" />
1014 <parameter name="TIMING_TDSS" value="0.2" />
1015 <parameter name="MEM_TINIT_US" value="500" />
1016 <parameter name="MEM_TMRD_CK" value="4" />
1017 <parameter name="MEM_TRAS_NS" value="35.0" />
1018 <parameter name="MEM_TRCD_NS" value="13.75" />
1019 <parameter name="MEM_TRP_NS" value="13.75" />
1020 <parameter name="MEM_TREFI_US" value="7.8" />
1021 <parameter name="MEM_TRFC_NS" value="260.0" />
1022 <parameter name="CFG_TCCD_NS" value="2.5" />
1023 <parameter name="MEM_TWR_NS" value="15.0" />
1024 <parameter name="MEM_TWIR" value="4" />
1025 <parameter name="MEM_TFAW_NS" value="40.0" />
1026 <parameter name="MEM_TRRD_NS" value="7.5" />
1027 <parameter name="MEM_TRTP_NS" value="7.5" />
1028 <parameter name="RATE" value="Full" />
1029 <parameter name="MEM_CLK_FREQ" value="400.0" />
1030 <parameter name="USE_MEM_CLK_FREQ" value="false" />
1031 <parameter name="FORCE_DQS_TRACKING" value="AUTO" />
1032 <parameter name="FORCE_SHADOW_REGS" value="AUTO" />
1033 <parameter name="MRS_MIRROR_PING_PONG_ATSO" value="false" />
1034 <parameter name="SYS_INFO_DEVICE_FAMILY" value="Cyclone V" />
1035 <parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM_VALID" value="false" />
1036 <parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM" value="" />
1037 <parameter name="DEVICE_FAMILY_PARAM" value="" />
1038 <parameter name="SPEED_GRADE" value="7" />
1039 <parameter name="IS_ES_DEVICE" value="false" />
1040 <parameter name="DISABLE_CHILDMESSAGING" value="false" />
1041 <parameter name="HARD_EMIF" value="true" />
1042 <parameter name="HHP_HPS" value="false" />
1043 <parameter name="HHP_HPS_VERIFICATION" value="false" />
1044 <parameter name="HHP_HPS_SIMULATION" value="false" />
1045 <parameter name="HPS_PROTOCOL" value="DEFAULT" />
1046 <parameter name="CUT_NEW_FAMILY_TIMING" value="true" />
1047 <parameter name="POWER_OF_TWO_BUS" value="true" />
1048 <parameter name="SOPC_COMPAT_RESET" value="false" />
1049 <parameter name="AVL_MAX_SIZE" value="4" />
1050 <parameter name="BYTE_ENABLE" value="true" />
1051 <parameter name="ENABLE_CTRL_AVALON_INTERFACE" value="true" />
1052 <parameter name="CTL_DEEP_POWERDN_EN" value="false" />
1053 <parameter name="CTL_SELF_REFRESH_EN" value="false" />
1054 <parameter name="AUTO_POWERDN_EN" value="false" />
1055 <parameter name="AUTO_PD_CYCLES" value="0" />
1056 <parameter name="CTLUSR_REFRESH_EN" value="false" />
1057 <parameter name="CTLAUTOPCH_EN" value="false" />
1058 <parameter name="CTLZQCAL_EN" value="false" />
1059 <parameter name="ADDR_ORDER" value="0" />
1060 <parameter name="CTL_LOOK_AHEAD_DEPTH" value="4" />
1061 <parameter name="CONTROLLER_LATENCY" value="5" />
1062 <parameter name="CFG_REORDER_DATA" value="false" />
1063 <parameter name="STARVE_LIMIT" value="10" />
1064 <parameter name="CTL_CSR_ENABLED" value="false" />
1065 <parameter name="CTL_CSR_CONNECTION" value="INTERNAL_JTAG" />
1066 <parameter name="CTL_ECC_ENABLED" value="false" />
1067 <parameter name="CTL_HRB_ENABLED" value="false" />
1068 <parameter name="CTL_ECC_AUTO_CORRECTION_ENABLED" value="false" />
1069 <parameter name="MULTICAST_EN" value="false" />
1070 <parameter name="CTL_DYNAMIC_BANK_ALLOCATION" value="false" />
1071 <parameter name="CTL_DYNAMIC_BANK_NUM" value="4" />
1072 <parameter name="DEBUG_MODE" value="false" />
1073 <parameter name="ENABLE_BURST_MERGE" value="false" />
1074 <parameter name="CTL_ENABLE_BURST_INTERRUPT" value="false" />
1075 <parameter name="CTL_ENABLE_BURST_TERMINATE" value="false" />
1076 <parameter name="LOCAL_ID_WIDTH" value="8" />
1077 <parameter name="WRBUFFER_ADDR_WIDTH" value="6" />

```

```

1078 <parameter name="MAX_PENDING_WR_CMD" value="8" />
1079 <parameter name="MAX_PENDING_RD_CMD" value="16" />
1080 <parameter name="USE_MM_ADAPTOR" value="true" />
1081 <parameter name="USE_AXI_ADAPTOR" value="false" />
1082 <parameter name="HCX_COMPAT_MODE" value="false" />
1083 <parameter name="CTL_CMD_QUEUE_DEPTH" value="8" />
1084 <parameter name="CTL_CSR_READ_ONLY" value="1" />
1085 <parameter name="CFG_DATA_REORDERING_TYPE" value="INTER_BANK" />
1086 <parameter name="NUM_OF_PORTS" value="1" />
1087 <parameter name="ENABLE_BONDING" value="false" />
1088 <parameter name="ENABLE_USER_ECC" value="false" />
1089 <parameter name="AVL_DATA_WIDTH_PORT" value="256,32,32,32,32,32" />
1090 <parameter name="PRIORITY_PORT" value="1,1,1,1,1,1" />
1091 <parameter name="WEIGHT_PORT" value="0,0,0,0,0,0" />
1092 <parameter name="CPOR_TYPE_PORT">Bidirectional , Bidirectional , Bidirectional ,
1093 Bidirectional , Bidirectional</parameter>
1094 <parameter name="ENABLE_EMIT_BFM_MASTER" value="false" />
1095 <parameter name="FORCE_SEQUENCER_TCL_DEBUG_MODE" value="false" />
1096 <parameter name="ENABLE_SEQUENCER_MARGINING_ON_BY_DEFAULT" value="false" />
1097 <parameter name="REF_CLK_FREQ" value="100.0" />
1098 <parameter name="REF_CLK_FREQ_PARAM_VALID" value="false" />
1099 <parameter name="REF_CLK_FREQ_MIN_PARAM" value="0.0" />
1100 <parameter name="REF_CLK_FREQ_MAX_PARAM" value="0.0" />
1101 <parameter name="PLL_DR_CLK_FREQ_PARAM" value="0.0" />
1102 <parameter name="PLL_DR_CLK_FREQ_SIM_STR_PARAM" value="" />
1103 <parameter name="PLL_DR_CLK_PHASE_PS_PARAM" value="0" />
1104 <parameter name="PLL_DR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1105 <parameter name="PLL_DR_CLK_MULT_PARAM" value="0" />
1106 <parameter name="PLL_DR_CLK_DIV_PARAM" value="0" />
1107 <parameter name="PLL_MEM_CLK_FREQ_PARAM" value="0.0" />
1108 <parameter name="PLL_MEM_CLK_FREQ_SIM_STR_PARAM" value="" />
1109 <parameter name="PLL_MEM_CLK_PHASE_PS_PARAM" value="0" />
1110 <parameter name="PLL_MEM_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1111 <parameter name="PLL_MEM_CLK_MULT_PARAM" value="0" />
1112 <parameter name="PLL_MEM_CLK_DIV_PARAM" value="0" />
1113 <parameter name="PLL_AFI_CLK_FREQ_PARAM" value="0.0" />
1114 <parameter name="PLL_AFI_CLK_FREQ_SIM_STR_PARAM" value="" />
1115 <parameter name="PLL_AFI_CLK_PHASE_PS_PARAM" value="0" />
1116 <parameter name="PLL_AFI_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1117 <parameter name="PLL_AFI_CLK_MULT_PARAM" value="0" />
1118 <parameter name="PLL_AFI_CLK_DIV_PARAM" value="0" />
1119 <parameter name="PLL_WRITE_CLK_FREQ_PARAM" value="0.0" />
1120 <parameter name="PLL_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
1121 <parameter name="PLL_WRITE_CLK_PHASE_PS_PARAM" value="0" />
1122 <parameter name="PLL_WRITE_CLK_MULT_PARAM" value="0" />
1123 <parameter name="PLL_WRITE_CLK_DIV_PARAM" value="0" />
1124 <parameter name="PLL_ADDR_CMD_CLK_FREQ_PARAM" value="0.0" />
1125 <parameter name="PLL_ADDR_CMD_CLK_FREQ_SIM_STR_PARAM" value="" />
1126 <parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_PARAM" value="0" />
1127 <parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1128 <parameter name="PLL_ADDR_CMD_CLK_MULT_PARAM" value="0" />
1129 <parameter name="PLL_ADDR_CMD_CLK_DIV_PARAM" value="0" />
1130 <parameter name="PLL_AFI_HALF_CLK_FREQ_PARAM" value="0.0" />
1131 <parameter name="PLL_AFI_HALF_CLK_FREQ_SIM_STR_PARAM" value="" />
1132 <parameter name="PLL_AFI_HALF_CLK_PHASE_PS_PARAM" value="0" />
1133 <parameter name="PLL_AFI_HALF_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1134 <parameter name="PLL_AFI_HALF_CLK_MULT_PARAM" value="0" />
1135 <parameter name="PLL_AFI_HALF_CLK_DIV_PARAM" value="0" />
1136 <parameter name="PLL_NIOS_CLK_FREQ_PARAM" value="0.0" />
1137 <parameter name="PLL_NIOS_CLK_FREQ_SIM_STR_PARAM" value="" />
1138 <parameter name="PLL_NIOS_CLK_PHASE_PS_PARAM" value="0" />
1139 <parameter name="PLL_NIOS_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1140 <parameter name="PLL_NIOS_CLK_MULT_PARAM" value="0" />
1141 <parameter name="PLL_NIOS_CLK_DIV_PARAM" value="0" />
1142 <parameter name="PLL_CONFIG_CLK_FREQ_PARAM" value="0.0" />
1143 <parameter name="PLL_CONFIG_CLK_FREQ_SIM_STR_PARAM" value="" />
1144 <parameter name="PLL_CONFIG_CLK_PHASE_PS_PARAM" value="0" />

```

```

1145 <parameter name="PLL_CONFIG_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1146 <parameter name="PLL_CONFIG_CLK_MULT_PARAM" value="0" />
1147 <parameter name="PLL_CONFIG_CLK_DIV_PARAM" value="0" />
1148 <parameter name="PLL_P2C_READ_CLK_FREQ_PARAM" value="0.0" />
1149 <parameter name="PLL_P2C_READ_CLK_PHASE_PS_PARAM" value="" />
1150 <parameter name="PLL_P2C_READ_CLK_FREQ_SIM_STR_PARAM" value="0" />
1151 <parameter name="PLL_P2C_READ_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1152 <parameter name="PLL_P2C_READ_CLK_MULT_PARAM" value="0" />
1153 <parameter name="PLL_P2C_READ_CLK_DIV_PARAM" value="0" />
1154 <parameter name="PLL_C2P_WRITE_CLK_FREQ_PARAM" value="0.0" />
1155 <parameter name="PLL_C2P_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
1156 <parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_PARAM" value="0" />
1157 <parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1158 <parameter name="PLL_C2P_WRITE_CLK_MULT_PARAM" value="0" />
1159 <parameter name="PLL_C2P_WRITE_CLK_DIV_PARAM" value="0" />
1160 <parameter name="PLL_HR_CLK_FREQ_PARAM" value="0.0" />
1161 <parameter name="PLL_HR_CLK_FREQ_SIM_STR_PARAM" value="" />
1162 <parameter name="PLL_HR_CLK_PHASE_PS_PARAM" value="0" />
1163 <parameter name="PLL_HR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1164 <parameter name="PLL_HR_CLK_MULT_PARAM" value="0" />
1165 <parameter name="PLL_HR_CLK_DIV_PARAM" value="0" />
1166 <parameter name="PLL_AFI_PHY_CLK_FREQ_PARAM" value="0.0" />
1167 <parameter name="PLL_AFI_PHY_CLK_FREQ_SIM_STR_PARAM" value="" />
1168 <parameter name="PLL_AFI_PHY_CLK_PHASE_PS_PARAM" value="0" />
1169 <parameter name="PLL_AFI_PHY_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
1170 <parameter name="PLL_AFI_PHY_CLK_MULT_PARAM" value="0" />
1171 <parameter name="PLL_AFI_PHY_CLK_DIV_PARAM" value="0" />
1172 <parameter name="PLL_CLK_PARAM_VALID" value="false" />
1173 <parameter name="ENABLE_EXTRA_REPORTING" value="false" />
1174 <parameter name="NUM_EXTRA_REPORT_PATH" value="10" />
1175 <parameter name="ENABLE_ISS_PROBES" value="false" />
1176 <parameter name="CALIB_REG_WIDTH" value="8" />
1177 <parameter name="USE_SEQUENCER_BFM" value="false" />
1178 <parameter name="DEFAULT_FAST_SIM_MODEL" value="true" />
1179 <parameter name="PLL_SHARING_MODE" value="None" />
1180 <parameter name="NUM_PLL_SHARING_INTERFACES" value="1" />
1181 <parameter name="EXPORT_AFI_HALF_CLK" value="false" />
1182 <parameter name="ABSTRACT_REAL_COMPARE_TEST" value="false" />
1183 <parameter name="INCLUDE_BOARD_DELAY_MODEL" value="false" />
1184 <parameter name="INCLUDE_MULTIRANK_BOARD_DELAY_MODEL" value="false" />
1185 <parameter name="USE_FAKE_PHY" value="false" />
1186 <parameter name="FORCE_MAX_LATENCY_COUNT_WIDTH" value="0" />
1187 <parameter name="ENABLE_NON_DESTRUCTIVE_CALIB" value="false" />
1188 <parameter name="TRACKING_ERROR_TEST" value="false" />
1189 <parameter name="TRACKING_WATCH_TEST" value="false" />
1190 <parameter name="MARGIN_VARIATION_TEST" value="false" />
1191 <parameter name="EXTRA_SETTINGS" value="" />
1192 <parameter name="MEM_DEVICE" value="MISSING_MODEL" />
1193 <parameter name="FORCE_SYNTHESIS_LANGUAGE" value="" />
1194 <parameter name="FORCED_NUM_WRITE_FR_CYCLE_SHIFTS" value="0" />
1195 <parameter name="SEQUENCER_TYPE" value="NIOS" />
1196 <parameter name="ADVERTISE_SEQUENCER_SW_BUILD_FILES" value="false" />
1197 <parameter name="FORCED_NON_LDC_ADDR_CMD_MEM_CK_INVERT" value="false" />
1198 <parameter name="PHY_ONLY" value="false" />
1199 <parameter name="SEQ_MODE" value="0" />
1200 <parameter name="ADVANCED_CK_PHASES" value="false" />
1201 <parameter name="COMMAND_PHASE" value="0.0" />
1202 <parameter name="MEM_CK_PHASE" value="0.0" />
1203 <parameter name="P2C_READ_CLOCK_ADD_PHASE" value="0.0" />
1204 <parameter name="C2P_WRITE_CLOCK_ADD_PHASE" value="0.0" />
1205 <parameter name="ACV_PHY_CLK_ADD_FR_PHASE" value="0.0" />
1206 <parameter name="MEM_VOLTAGE" value="1.5V DDR3" />
1207 <parameter name="PLL_LOCATION" value="Top_Bottom" />
1208 <parameter name="SKIP_MEM_INIT" value="true" />
1209 <parameter name="READ_DQ_DQS_CLOCK_SOURCE" value="INVERTED_DQS_BUS" />
1210 <parameter name="DQ_INPUT_REG_USE_CLKN" value="false" />
1211 <parameter name="DQS_DQSN_MODE" value="DIFFERENTIAL" />
1212 <parameter name="AFI_DEBUG_INFO_WIDTH" value="32" />

```

```

1213 <parameter name="CALIBRATION_MODE" value="Skip" />
1214 <parameter name="NIOS_ROM_DATA_WIDTH" value="32" />
1215 <parameter name="READ_FIFO_SIZE" value="8" />
1216 <parameter name="PHY_CSR_ENABLED" value="false" />
1217 <parameter name="PHY_CSR_CONNECTION" value="INTERNAL_JTAG" />
1218 <parameter name="USER_DEBUG_LEVEL" value="1" />
1219 <parameter name="TIMING_BOARD_DERATE_METHOD" value="AUTO" />
1220 <parameter name="TIMING_BOARD_CK_CKN_SLEW_RATE" value="2.0" />
1221 <parameter name="TIMING_BOARD_AC_SLEW_RATE" value="1.0" />
1222 <parameter name="TIMING_BOARD_DQS_DQSN_SLEW_RATE" value="2.0" />
1223 <parameter name="TIMING_BOARD_DQ_SLEW_RATE" value="1.0" />
1224 <parameter name="TIMING_BOARD_TIS" value="0.0" />
1225 <parameter name="TIMING_BOARD_TIH" value="0.0" />
1226 <parameter name="TIMING_BOARD_TDS" value="0.0" />
1227 <parameter name="TIMING_BOARD_TDH" value="0.0" />
1228 <parameter name="TIMING_BOARD_ISI_METHOD" value="AUTO" />
1229 <parameter name="TIMING_BOARD_AC_EYE_REDUCTION_SU" value="0.0" />
1230 <parameter name="TIMING_BOARD_AC_EYE_REDUCTION_H" value="0.0" />
1231 <parameter name="TIMING_BOARD_DQ_EYE_REDUCTION" value="0.0" />
1232 <parameter name="TIMING_BOARD_DELTA_DQS_ARRIVAL_TIME" value="0.0" />
1233 <parameter name="PACKAGE_DESKEW" value="false" />
1234 <parameter name="AC_PACKAGE_DESKEW" value="false" />
1235 <parameter name="TIMING_BOARD_MAX_CK_DELAY" value="0.29132963" />
1236 <parameter name="TIMING_BOARD_MAX_DQS_DELAY" value="0.286640405" />
1237 <parameter name="TIMING_BOARD_SKEW_CKDQS_DIMM_MIN" value="-0.005828107" />
1238 <parameter name="TIMING_BOARD_SKEW_CKDQS_DIMM_MAX" value="0.048251827" />
1239 <parameter name="TIMING_BOARD_SKEW_BETWEEN_DIMMS" value="0.05" />
1240 <parameter name="TIMING_BOARD_SKEW_WITHIN_DQS" value="0.001782034" />
1241 <parameter name="TIMING_BOARD_SKEW_BETWEEN_DQS" value="0.029609881" />
1242 <parameter name="TIMING_BOARD_DQ_TO_DQS_SKEW" value="0.0" />
1243 <parameter name="TIMING_BOARD_AC_SKEW" value="0.096664741" />
1244 <parameter name="TIMING_BOARD_AC_TO_CK_SKEW" value="0.026243846" />
1245 <parameter name="ENABLE_EXPORT_SEQ_DEBUG_BRIDGE" value="false" />
1246 <parameter name="CORE_DEBUG_CONNECTION" value="EXPORT" />
1247 <parameter name="ADD_EXTERNAL_SEQ_DEBUG_NIOS" value="false" />
1248 <parameter name="ED_EXPORT_SEQ_DEBUG" value="false" />
1249 <parameter name="ADD EFFICIENCY_MONITOR" value="false" />
1250 <parameter name="ENABLE_ABS_RAM_MEM_INIT" value="false" />
1251 <parameter name="ABS_RAM_MEM_INIT_FILENAME" value="meminit" />
1252 <parameter name="DLL_SHARING_MODE" value="None" />
1253 <parameter name="NUM_DLL_SHARING_INTERFACES" value="1" />
1254 <parameter name="OCT_SHARING_MODE" value="None" />
1255 <parameter name="NUM_OCT_SHARING_INTERFACES" value="1" />
1256 <parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
1257 </module>
1258 <module kind="clock_source" version="13.0" enabled="1" name="refclk_100">
1259 <parameter name="clockFrequency" value="100000000" />
1260 <parameter name="clockFrequencyKnown" value="true" />
1261 <parameter name="inputClockFrequency" value="0" />
1262 <parameter name="resetSynchronousEdges" value="NONE" />
1263 </module>
1264 <module kind="altera_pll" version="13.0" enabled="1" name="pll_0">
1265 <parameter name="device_family" value="Cyclone V" />
1266 <parameter name="gui_device_speed_grade" value="7" />
1267 <parameter name="gui_pll_mode" value="Integer-N PLL" />
1268 <parameter name="gui_reference_clock_frequency" value="400.0" />
1269 <parameter name="gui_channel_spacing" value="0.0" />
1270 <parameter name="gui_operation_mode" value="normal" />
1271 <parameter name="gui_feedback_clock" value="Global Clock" />
1272 <parameter name="gui_fractional_cout" value="32" />
1273 <parameter name="gui_dsm_out_sel" value="1st_order" />
1274 <parameter name="gui_use_locked" value="false" />
1275 <parameter name="gui_en_adv_params" value="false" />
1276 <parameter name="gui_number_of_clocks" value="1" />
1277 <parameter name="gui_multiply_factor" value="1" />
1278 <parameter name="gui_frac_multiply_factor" value="1" />
1279 <parameter name="gui_divide_factor_n" value="1" />
1280 <parameter name="gui_output_clock_frequency0" value="100.0" />

```

```

1281 <parameter name="gui_divide_factor_c0" value="1" />
1282 <parameter name="gui_actual_output_clock_frequency0" value="0 MHz" />
1283 <parameter name="gui_ps_units0" value="ps" />
1284 <parameter name="gui_phase_shift0" value="0" />
1285 <parameter name="gui_phase_shift_deg0" value="0" />
1286 <parameter name="gui_actual_phase_shift0" value="0" />
1287 <parameter name="gui_duty_cycle0" value="50" />
1288 <parameter name="gui_output_clock_frequency1" value="100.0" />
1289 <parameter name="gui_divide_factor_c1" value="1" />
1290 <parameter name="gui_actual_output_clock_frequency1" value="0 MHz" />
1291 <parameter name="gui_ps_units1" value="ps" />
1292 <parameter name="gui_phase_shift1" value="0" />
1293 <parameter name="gui_phase_shift_deg1" value="0" />
1294 <parameter name="gui_actual_phase_shift1" value="0" />
1295 <parameter name="gui_duty_cycle1" value="50" />
1296 <parameter name="gui_output_clock_frequency2" value="100.0" />
1297 <parameter name="gui_divide_factor_c2" value="1" />
1298 <parameter name="gui_actual_output_clock_frequency2" value="0 MHz" />
1299 <parameter name="gui_ps_units2" value="ps" />
1300 <parameter name="gui_phase_shift2" value="0" />
1301 <parameter name="gui_phase_shift_deg2" value="0" />
1302 <parameter name="gui_actual_phase_shift2" value="0" />
1303 <parameter name="gui_duty_cycle2" value="50" />
1304 <parameter name="gui_output_clock_frequency3" value="100.0" />
1305 <parameter name="gui_divide_factor_c3" value="1" />
1306 <parameter name="gui_actual_output_clock_frequency3" value="0 MHz" />
1307 <parameter name="gui_ps_units3" value="ps" />
1308 <parameter name="gui_phase_shift3" value="0" />
1309 <parameter name="gui_phase_shift_deg3" value="0" />
1310 <parameter name="gui_actual_phase_shift3" value="0" />
1311 <parameter name="gui_duty_cycle3" value="50" />
1312 <parameter name="gui_output_clock_frequency4" value="100.0" />
1313 <parameter name="gui_divide_factor_c4" value="1" />
1314 <parameter name="gui_actual_output_clock_frequency4" value="0 MHz" />
1315 <parameter name="gui_ps_units4" value="ps" />
1316 <parameter name="gui_phase_shift4" value="0" />
1317 <parameter name="gui_phase_shift_deg4" value="0" />
1318 <parameter name="gui_actual_phase_shift4" value="0" />
1319 <parameter name="gui_duty_cycle4" value="50" />
1320 <parameter name="gui_output_clock_frequency5" value="100.0" />
1321 <parameter name="gui_divide_factor_c5" value="1" />
1322 <parameter name="gui_actual_output_clock_frequency5" value="0 MHz" />
1323 <parameter name="gui_ps_units5" value="ps" />
1324 <parameter name="gui_phase_shift5" value="0" />
1325 <parameter name="gui_phase_shift_deg5" value="0" />
1326 <parameter name="gui_actual_phase_shift5" value="0" />
1327 <parameter name="gui_duty_cycle5" value="50" />
1328 <parameter name="gui_output_clock_frequency6" value="100.0" />
1329 <parameter name="gui_divide_factor_c6" value="1" />
1330 <parameter name="gui_actual_output_clock_frequency6" value="0 MHz" />
1331 <parameter name="gui_ps_units6" value="ps" />
1332 <parameter name="gui_phase_shift6" value="0" />
1333 <parameter name="gui_phase_shift_deg6" value="0" />
1334 <parameter name="gui_actual_phase_shift6" value="0" />
1335 <parameter name="gui_duty_cycle6" value="50" />
1336 <parameter name="gui_output_clock_frequency7" value="100.0" />
1337 <parameter name="gui_divide_factor_c7" value="1" />
1338 <parameter name="gui_actual_output_clock_frequency7" value="0 MHz" />
1339 <parameter name="gui_ps_units7" value="ps" />
1340 <parameter name="gui_phase_shift7" value="0" />
1341 <parameter name="gui_phase_shift_deg7" value="0" />
1342 <parameter name="gui_actual_phase_shift7" value="0" />
1343 <parameter name="gui_duty_cycle7" value="50" />
1344 <parameter name="gui_output_clock_frequency8" value="100.0" />
1345 <parameter name="gui_divide_factor_c8" value="1" />
1346 <parameter name="gui_actual_output_clock_frequency8" value="0 MHz" />
1347 <parameter name="gui_ps_units8" value="ps" />
1348 <parameter name="gui_phase_shift8" value="0" />

```

```

1349 <parameter name="gui_phase_shift_deg8" value="0" />
1350 <parameter name="gui_actual_phase_shift8" value="0" />
1351 <parameter name="gui_duty_cycle8" value="50" />
1352 <parameter name="gui_output_clock_frequency9" value="100.0" />
1353 <parameter name="gui_divide_factor_c9" value="1" />
1354 <parameter name="gui_actual_output_clock_frequency9" value="0 MHz" />
1355 <parameter name="gui_ps_units9" value="ps" />
1356 <parameter name="gui_phase_shift9" value="0" />
1357 <parameter name="gui_phase_shift_deg9" value="0" />
1358 <parameter name="gui_actual_phase_shift9" value="0" />
1359 <parameter name="gui_duty_cycle9" value="50" />
1360 <parameter name="gui_output_clock_frequency10" value="100.0" />
1361 <parameter name="gui_divide_factor_c10" value="1" />
1362 <parameter name="gui_actual_output_clock_frequency10" value="0 MHz" />
1363 <parameter name="gui_ps_units10" value="ps" />
1364 <parameter name="gui_phase_shift10" value="0" />
1365 <parameter name="gui_phase_shift_deg10" value="0" />
1366 <parameter name="gui_actual_phase_shift10" value="0" />
1367 <parameter name="gui_duty_cycle10" value="50" />
1368 <parameter name="gui_output_clock_frequency11" value="100.0" />
1369 <parameter name="gui_divide_factor_c11" value="1" />
1370 <parameter name="gui_actual_output_clock_frequency11" value="0 MHz" />
1371 <parameter name="gui_ps_units11" value="ps" />
1372 <parameter name="gui_phase_shift11" value="0" />
1373 <parameter name="gui_phase_shift_deg11" value="0" />
1374 <parameter name="gui_actual_phase_shift11" value="0" />
1375 <parameter name="gui_duty_cycle11" value="50" />
1376 <parameter name="gui_output_clock_frequency12" value="100.0" />
1377 <parameter name="gui_divide_factor_c12" value="1" />
1378 <parameter name="gui_actual_output_clock_frequency12" value="0 MHz" />
1379 <parameter name="gui_ps_units12" value="ps" />
1380 <parameter name="gui_phase_shift12" value="0" />
1381 <parameter name="gui_phase_shift_deg12" value="0" />
1382 <parameter name="gui_actual_phase_shift12" value="0" />
1383 <parameter name="gui_duty_cycle12" value="50" />
1384 <parameter name="gui_output_clock_frequency13" value="100.0" />
1385 <parameter name="gui_divide_factor_c13" value="1" />
1386 <parameter name="gui_actual_output_clock_frequency13" value="0 MHz" />
1387 <parameter name="gui_ps_units13" value="ps" />
1388 <parameter name="gui_phase_shift13" value="0" />
1389 <parameter name="gui_phase_shift_deg13" value="0" />
1390 <parameter name="gui_actual_phase_shift13" value="0" />
1391 <parameter name="gui_duty_cycle13" value="50" />
1392 <parameter name="gui_output_clock_frequency14" value="100.0" />
1393 <parameter name="gui_divide_factor_c14" value="1" />
1394 <parameter name="gui_actual_output_clock_frequency14" value="0 MHz" />
1395 <parameter name="gui_ps_units14" value="ps" />
1396 <parameter name="gui_phase_shift14" value="0" />
1397 <parameter name="gui_phase_shift_deg14" value="0" />
1398 <parameter name="gui_actual_phase_shift14" value="0" />
1399 <parameter name="gui_duty_cycle14" value="50" />
1400 <parameter name="gui_output_clock_frequency15" value="100.0" />
1401 <parameter name="gui_divide_factor_c15" value="1" />
1402 <parameter name="gui_actual_output_clock_frequency15" value="0 MHz" />
1403 <parameter name="gui_ps_units15" value="ps" />
1404 <parameter name="gui_phase_shift15" value="0" />
1405 <parameter name="gui_phase_shift_deg15" value="0" />
1406 <parameter name="gui_actual_phase_shift15" value="0" />
1407 <parameter name="gui_duty_cycle15" value="50" />
1408 <parameter name="gui_output_clock_frequency16" value="100.0" />
1409 <parameter name="gui_divide_factor_c16" value="1" />
1410 <parameter name="gui_actual_output_clock_frequency16" value="0 MHz" />
1411 <parameter name="gui_ps_units16" value="ps" />
1412 <parameter name="gui_phase_shift16" value="0" />
1413 <parameter name="gui_phase_shift_deg16" value="0" />
1414 <parameter name="gui_actual_phase_shift16" value="0" />
1415 <parameter name="gui_duty_cycle16" value="50" />
1416 <parameter name="gui_output_clock_frequency17" value="100.0" />

```

```

1417 <parameter name="gui_divide_factor_c17" value="1" />
1418 <parameter name="gui_actual_output_clock_frequency17" value="0 MHz" />
1419 <parameter name="gui_ps_units17" value="ps" />
1420 <parameter name="gui_phase_shift17" value="0" />
1421 <parameter name="gui_phase_shift_deg17" value="0" />
1422 <parameter name="gui_actual_phase_shift17" value="0" />
1423 <parameter name="gui_duty_cycle17" value="50" />
1424 <parameter name="gui_pll_auto_reset" value="Off" />
1425 <parameter name="gui_pll_bandwidth_preset" value="Auto" />
1426 <parameter name="gui_en_reconf" value="false" />
1427 <parameter name="gui_en_dps_ports" value="false" />
1428 <parameter name="gui_en_phout_ports" value="false" />
1429 <parameter name="gui_mif_generate" value="false" />
1430 <parameter name="gui_enable_mif_dps" value="false" />
1431 <parameter name="gui_dps_cntr" value="C0" />
1432 <parameter name="gui_dps_num" value="1" />
1433 <parameter name="gui_dps_dir" value="Positive" />
1434 <parameter name="gui_refclk_switch" value="false" />
1435 <parameter name="gui_refclk1_frequency" value="100.0" />
1436 <parameter name="gui_switchover_mode">Automatic Switchover</parameter>
1437 <parameter name="gui_switchover_delay" value="0" />
1438 <parameter name="gui_active_clk" value="false" />
1439 <parameter name="gui_clk_bad" value="false" />
1440 <parameter name="gui_enable_cascade_out" value="false" />
1441 <parameter name="gui_enable_cascade_in" value="false" />
1442 <parameter name="gui_pll_cascading_mode">Create an adjpll signal to connect with an
    upstream PLL</parameter>
1443 <parameter name="AUTO_REFCLK_CLOCK_RATE" value="400000000" />
1444 </module>
1445 <module
1446   kind="altera_avalon_mm_clock_crossing_bridge"
1447   version="13.0"
1448   enabled="1"
1449   name="mm_clock_crossing_bridge_0">
1450   <parameter name="DATA_WIDTH" value="32" />
1451   <parameter name="SYMBOL_WIDTH" value="8" />
1452   <parameter name="ADDRESS_WIDTH" value="28" />
1453   <parameter name="ADDRESS_UNITS" value="WORDS" />
1454   <parameter name="MAX_BURST_SIZE" value="1" />
1455   <parameter name="COMMAND_FIFO_DEPTH" value="4" />
1456   <parameter name="RESPONSE_FIFO_DEPTH" value="4" />
1457   <parameter name="MASTER_SYNC_DEPTH" value="2" />
1458   <parameter name="SLAVE_SYNC_DEPTH" value="2" />
1459   <parameter name="AUTO_M0_CLK_CLOCK_RATE" value="100000000" />
1460   <parameter name="AUTO_S0_CLK_CLOCK_RATE" value="50000000" />
1461   <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
1462 </module>
1463 <module kind="ImgDiff" version="1.0" enabled="0" name="ImgDiff_0">
1464   <parameter name="DATA_WIDTH" value="32" />
1465   <parameter name="ADDRESS_WIDTH" value="32" />
1466   <parameter name="MASTER_WRITE_BURST_CAPABLE" value="0" />
1467   <parameter name="MASTER_WRITE_MAXIMUM_BURST_COUNT" value="2" />
1468   <parameter name="MASTER_WRITE_BURST_COUNT_WIDTH" value="2" />
1469   <parameter name="MASTER_WRITE_FIFO_DEPTH" value="32" />
1470   <parameter name="MASTER_WRITE_FIFO_DEPTH_LOG2" value="5" />
1471   <parameter name="MASTER_WRITE_MEMORY_BASED_FIFO" value="1" />
1472   <parameter name="MASTER_READ_BURST_CAPABLE" value="0" />
1473   <parameter name="MASTER_READ_MAXIMUM_BURST_COUNT" value="2" />
1474   <parameter name="MASTER_READ_BURST_COUNT_WIDTH" value="2" />
1475   <parameter name="MASTER_READ_FIFO_DEPTH" value="32" />
1476   <parameter name="MASTER_READ_FIFO_DEPTH_LOG2" value="5" />
1477   <parameter name="MASTER_READ_MEMORY_BASED_FIFO" value="1" />
1478   <parameter name="IRQ_EN" value="0" />
1479   <parameter name="AUTO_CLK_IN_CLOCK_RATE" value="100000000" />
1480   <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
1481 </module>
1482 <module
1483   kind="altera_avalon_mm_clock_crossing_bridge"

```

```

1484    version="13.0"
1485    enabled="1"
1486    name="mm_clock_crossing_bridge_1">
1487      <parameter name="DATA_WIDTH" value="32" />
1488      <parameter name="SYMBOL_WIDTH" value="8" />
1489      <parameter name="ADDRESS_WIDTH" value="10" />
1490      <parameter name="ADDRESS_UNITS" value="WORDS" />
1491      <parameter name="MAX_BURST_SIZE" value="1" />
1492      <parameter name="COMMAND_FIFO_DEPTH" value="4" />
1493      <parameter name="RESPONSE_FIFO_DEPTH" value="4" />
1494      <parameter name="MASTER_SYNC_DEPTH" value="2" />
1495      <parameter name="SLAVE_SYNC_DEPTH" value="2" />
1496      <parameter name="AUTO_M0_CLK_CLOCK_RATE" value="50000000" />
1497      <parameter name="AUTO_S0_CLK_CLOCK_RATE" value="100000000" />
1498      <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
1499    </module>
1500  <module kind="IPTest" version="1.0" enabled="1" name="IPTest_0">
1501    <parameter name="DATA_WIDTH" value="32" />
1502    <parameter name="ADDRESS_WIDTH" value="32" />
1503    <parameter name="MASTER_WRITE_BURST_CAPABLE" value="0" />
1504    <parameter name="MASTER_WRITE_MAXIMUM_BURST_COUNT" value="2" />
1505    <parameter name="MASTER_WRITE_BURST_COUNT_WIDTH" value="2" />
1506    <parameter name="MASTER_WRITE_FIFO_DEPTH" value="32" />
1507    <parameter name="MASTER_WRITE_FIFO_DEPTH_LOG2" value="5" />
1508    <parameter name="MASTER_WRITE_MEMORY_BASED_FIFO" value="1" />
1509    <parameter name="MASTER_READ_BURST_CAPABLE" value="0" />
1510    <parameter name="MASTER_READ_MAXIMUM_BURST_COUNT" value="2" />
1511    <parameter name="MASTER_READ_BURST_COUNT_WIDTH" value="2" />
1512    <parameter name="MASTER_READ_FIFO_DEPTH" value="32" />
1513    <parameter name="MASTER_READ_FIFO_DEPTH_LOG2" value="5" />
1514    <parameter name="MASTER_READ_MEMORY_BASED_FIFO" value="1" />
1515    <parameter name="IRQ_EN" value="0" />
1516    <parameter name="AUTO_CLK_IN_CLOCK_RATE" value="100000000" />
1517    <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
1518  </module>
1519  <connection
1520    kind="avalon"
1521    version="13.0"
1522    start="hps_0.h2f_lw_axi_master"
1523    end="sysid_qsys.control_slave">
1524    <parameter name="arbitrationPriority" value="1" />
1525    <parameter name="baseAddress" value="0x00010000" />
1526    <parameter name="defaultConnection" value="false" />
1527  </connection>
1528  <connection
1529    kind="avalon"
1530    version="13.0"
1531    start="hps_0.h2f_lw_axi_master"
1532    end="led_pio.s1">
1533    <parameter name="arbitrationPriority" value="1" />
1534    <parameter name="baseAddress" value="0x00010040" />
1535    <parameter name="defaultConnection" value="false" />
1536  </connection>
1537  <connection
1538    kind="avalon"
1539    version="13.0"
1540    start="hps_0.h2f_lw_axi_master"
1541    end="dipsw_pio.s1">
1542    <parameter name="arbitrationPriority" value="1" />
1543    <parameter name="baseAddress" value="0x00010080" />
1544    <parameter name="defaultConnection" value="false" />
1545  </connection>
1546  <connection
1547    kind="avalon"
1548    version="13.0"
1549    start="hps_0.h2f_lw_axi_master"
1550    end="button_pio.s1">
1551    <parameter name="arbitrationPriority" value="1" />

```

```

1552 <parameter name="baseAddress" value="0x000100c0" />
1553 <parameter name="defaultConnection" value="false" />
1554 </connection>
1555 <connection
1556   kind="interrupt"
1557   version="13.0"
1558   start="intr_capturer_0.interrupt_receiver"
1559   end="button_pio.irq">
1560   <parameter name="irqNumber" value="1" />
1561 </connection>
1562 <connection
1563   kind="interrupt"
1564   version="13.0"
1565   start="intr_capturer_0.interrupt_receiver"
1566   end="dipsw_pio.irq">
1567   <parameter name="irqNumber" value="0" />
1568 </connection>
1569 <connection
1570   kind="interrupt"
1571   version="13.0"
1572   start="intr_capturer_0.interrupt_receiver"
1573   end="jtag_uart.irq">
1574   <parameter name="irqNumber" value="2" />
1575 </connection>
1576 <connection
1577   kind="interrupt"
1578   version="13.0"
1579   start="hps_0.f2h_irq0"
1580   end="dipsw_pio.irq">
1581   <parameter name="irqNumber" value="0" />
1582 </connection>
1583 <connection
1584   kind="interrupt"
1585   version="13.0"
1586   start="hps_0.f2h_irq0"
1587   end="button_pio.irq">
1588   <parameter name="irqNumber" value="1" />
1589 </connection>
1590 <connection
1591   kind="interrupt"
1592   version="13.0"
1593   start="hps_0.f2h_irq0"
1594   end="jtag_uart.irq">
1595   <parameter name="irqNumber" value="2" />
1596 </connection>
1597 <connection
1598   kind="avalon"
1599   version="13.0"
1600   start="hps_0.h2f_lw_axi_master"
1601   end="jtag_uart.avalon_jtag_slave">
1602   <parameter name="arbitrationPriority" value="1" />
1603   <parameter name="baseAddress" value="0x00020000" />
1604   <parameter name="defaultConnection" value="false" />
1605 </connection>
1606 <connection
1607   kind="clock"
1608   version="13.0"
1609   start="ext_clk_50.clk"
1610   end="intr_capturer_0.clock" />
1611 <connection
1612   kind="clock"
1613   version="13.0"
1614   start="ext_clk_50.clk"
1615   end="fpga_only_master.clk" />
1616 <connection
1617   kind="clock"
1618   version="13.0"
1619   start="ext_clk_50.clk"

```

```

1620    end="jtag_uart.clk" />
1621  <connection
1622    kind="clock"
1623    version="13.0"
1624    start="ext_clk_50.clk"
1625    end="button_pio.clk" />
1626  <connection
1627    kind="clock"
1628    version="13.0"
1629    start="ext_clk_50.clk"
1630    end="dipsw_pio.clk" />
1631  <connection kind="clock" version="13.0" start="ext_clk_50.clk" end="led_pio.clk" />
1632  <connection
1633    kind="clock"
1634    version="13.0"
1635    start="ext_clk_50.clk"
1636    end="sysid_qsys.clk" />
1637  <connection
1638    kind="clock"
1639    version="13.0"
1640    start="ext_clk_50.clk"
1641    end="hps_0.h2f_lw_axi_clock" />
1642  <connection
1643    kind="clock"
1644    version="13.0"
1645    start="ext_clk_50.clk"
1646    end="hps_0.f2h_axi_clock" />
1647  <connection
1648    kind="avalon"
1649    version="13.0"
1650    start="fpga_only_master.master"
1651    end="jtag_uart.avalon_jtag_slave">
1652    <parameter name="arbitrationPriority" value="1" />
1653    <parameter name="baseAddress" value="0x00020000" />
1654    <parameter name="defaultConnection" value="false" />
1655  </connection>
1656  <connection
1657    kind="avalon"
1658    version="13.0"
1659    start="fpga_only_master.master"
1660    end="button_pio.s1">
1661    <parameter name="arbitrationPriority" value="1" />
1662    <parameter name="baseAddress" value="0x000100c0" />
1663    <parameter name="defaultConnection" value="false" />
1664  </connection>
1665  <connection
1666    kind="avalon"
1667    version="13.0"
1668    start="fpga_only_master.master"
1669    end="dipsw_pio.s1">
1670    <parameter name="arbitrationPriority" value="1" />
1671    <parameter name="baseAddress" value="0x00010080" />
1672    <parameter name="defaultConnection" value="false" />
1673  </connection>
1674  <connection
1675    kind="avalon"
1676    version="13.0"
1677    start="fpga_only_master.master"
1678    end="led_pio.s1">
1679    <parameter name="arbitrationPriority" value="1" />
1680    <parameter name="baseAddress" value="0x00010040" />
1681    <parameter name="defaultConnection" value="false" />
1682  </connection>
1683  <connection
1684    kind="avalon"
1685    version="13.0"
1686    start="fpga_only_master.master"
1687    end="sysid_qsys.control_slave">

```

```

1688 <parameter name="arbitrationPriority" value="1" />
1689 <parameter name="baseAddress" value="0x00010000" />
1690 <parameter name="defaultConnection" value="false" />
1691 </connection>
1692 <connection
1693   kind="avalon"
1694   version="13.0"
1695   start="fpga_only_master.master"
1696   end="intr_capturer_0.avlon_slave_0">
1697   <parameter name="arbitrationPriority" value="1" />
1698   <parameter name="baseAddress" value="0x00030000" />
1699   <parameter name="defaultConnection" value="false" />
1700 </connection>
1701 <connection
1702   kind="clock"
1703   version="13.0"
1704   start="ext_clk_50.clk"
1705   end="hps_only_master.clk" />
1706 <connection
1707   kind="avalon"
1708   version="13.0"
1709   start="hps_only_master.master"
1710   end="hps_0.f2h_axi_slave">
1711   <parameter name="arbitrationPriority" value="1" />
1712   <parameter name="baseAddress" value="0x0000" />
1713   <parameter name="defaultConnection" value="false" />
1714 </connection>
1715 <connection
1716   kind="clock"
1717   version="13.0"
1718   start="ext_clk_50.clk"
1719   end="hps_0.h2f_axi_clock" />
1720 <connection
1721   kind="reset"
1722   version="13.0"
1723   start="ext_clk_50.clk_reset"
1724   end="intr_capturer_0.reset_sink" />
1725 <connection
1726   kind="reset"
1727   version="13.0"
1728   start="ext_clk_50.clk_reset"
1729   end="fpga_only_master.clk_reset" />
1730 <connection
1731   kind="reset"
1732   version="13.0"
1733   start="ext_clk_50.clk_reset"
1734   end="jtag_uart.reset" />
1735 <connection
1736   kind="reset"
1737   version="13.0"
1738   start="ext_clk_50.clk_reset"
1739   end="button_pio.reset" />
1740 <connection
1741   kind="reset"
1742   version="13.0"
1743   start="ext_clk_50.clk_reset"
1744   end="dipsw_pio.reset" />
1745 <connection
1746   kind="reset"
1747   version="13.0"
1748   start="ext_clk_50.clk_reset"
1749   end="led_pio.reset" />
1750 <connection
1751   kind="reset"
1752   version="13.0"
1753   start="ext_clk_50.clk_reset"
1754   end="sysid_qsys.reset" />
1755 <connection

```

```

1756     kind="reset"
1757     version="13.0"
1758     start="ext_clk_50.clk_reset"
1759     end="hps_only_master.clk_reset" />
1760 <connection
1761   kind="clock"
1762   version="13.0"
1763   start="refclk_100.clk"
1764   end="mem_if_ddr3_fpga pll_ref_clk" />
1765 <connection
1766   kind="clock"
1767   version="13.0"
1768   start="mem_if_ddr3_fpga.afi_clk"
1769   end="pll_0.refclk" />
1770 <connection
1771   kind="reset"
1772   version="13.0"
1773   start="ext_clk_50.clk_reset"
1774   end="pll_0.reset" />
1775 <connection
1776   kind="reset"
1777   version="13.0"
1778   start="ext_clk_50.clk_reset"
1779   end="mem_if_ddr3_fpga.mp_cmd_reset_n_0" />
1780 <connection
1781   kind="reset"
1782   version="13.0"
1783   start="ext_clk_50.clk_reset"
1784   end="mem_if_ddr3_fpga.mp_rfifo_reset_n_0" />
1785 <connection
1786   kind="reset"
1787   version="13.0"
1788   start="ext_clk_50.clk_reset"
1789   end="mem_if_ddr3_fpga.mp_wfifo_reset_n_0" />
1790 <connection
1791   kind="reset"
1792   version="13.0"
1793   start="ext_clk_50.clk_reset"
1794   end="mem_if_ddr3_fpga.mp_rfifo_reset_n_1" />
1795 <connection
1796   kind="reset"
1797   version="13.0"
1798   start="ext_clk_50.clk_reset"
1799   end="mem_if_ddr3_fpga.mp_wfifo_reset_n_1" />
1800 <connection
1801   kind="reset"
1802   version="13.0"
1803   start="ext_clk_50.clk_reset"
1804   end="mem_if_ddr3_fpga.mp_rfifo_reset_n_2" />
1805 <connection
1806   kind="reset"
1807   version="13.0"
1808   start="ext_clk_50.clk_reset"
1809   end="mem_if_ddr3_fpga.mp_wfifo_reset_n_2" />
1810 <connection
1811   kind="reset"
1812   version="13.0"
1813   start="ext_clk_50.clk_reset"
1814   end="mem_if_ddr3_fpga.mp_rfifo_reset_n_3" />
1815 <connection
1816   kind="reset"
1817   version="13.0"
1818   start="ext_clk_50.clk_reset"
1819   end="mem_if_ddr3_fpga.mp_wfifo_reset_n_3" />
1820 <connection
1821   kind="clock"
1822   version="13.0"
1823   start="pll_0.outclk0"

```

```

1824    end="mem_if_ddr3_fpga.mp_cmd_clk_0" />
1825  <connection
1826    kind="clock"
1827    version="13.0"
1828    start="pll_0.outclk0"
1829    end="mem_if_ddr3_fpga.mp_rfifo_clk_0" />
1830  <connection
1831    kind="clock"
1832    version="13.0"
1833    start="pll_0.outclk0"
1834    end="mem_if_ddr3_fpga.mp_wfifo_clk_0" />
1835  <connection
1836    kind="clock"
1837    version="13.0"
1838    start="pll_0.outclk0"
1839    end="mem_if_ddr3_fpga.mp_rfifo_clk_1" />
1840  <connection
1841    kind="clock"
1842    version="13.0"
1843    start="pll_0.outclk0"
1844    end="mem_if_ddr3_fpga.mp_rfifo_clk_2" />
1845  <connection
1846    kind="clock"
1847    version="13.0"
1848    start="pll_0.outclk0"
1849    end="mem_if_ddr3_fpga.mp_wfifo_clk_1" />
1850  <connection
1851    kind="clock"
1852    version="13.0"
1853    start="pll_0.outclk0"
1854    end="mem_if_ddr3_fpga.mp_wfifo_clk_2" />
1855  <connection
1856    kind="clock"
1857    version="13.0"
1858    start="pll_0.outclk0"
1859    end="mem_if_ddr3_fpga.mp_rfifo_clk_3" />
1860  <connection
1861    kind="clock"
1862    version="13.0"
1863    start="pll_0.outclk0"
1864    end="mem_if_ddr3_fpga.mp_wfifo_clk_3" />
1865  <connection
1866    kind="reset"
1867    version="13.0"
1868    start="ext_clk_50.clk_reset"
1869    end="refclk_100.clk_in_reset" />
1870  <connection
1871    kind="avalon"
1872    version="13.0"
1873    start="hps_0.h2f_axi_master"
1874    end="mm_clock_crossing_bridge_0.s0">
1875    <parameter name="arbitrationPriority" value="1" />
1876    <parameter name="baseAddress" value="0x0000" />
1877    <parameter name="defaultConnection" value="false" />
1878  </connection>
1879  <connection
1880    kind="avalon"
1881    version="13.0"
1882    start="mm_clock_crossing_bridge_0.m0"
1883    end="mem_if_ddr3_fpga.avl_0">
1884    <parameter name="arbitrationPriority" value="1" />
1885    <parameter name="baseAddress" value="0x0000" />
1886    <parameter name="defaultConnection" value="false" />
1887  </connection>
1888  <connection
1889    kind="clock"
1890    version="13.0"
1891    start="pll_0.outclk0"

```

```

1892    end="mm_clock_crossing_bridge_0.m0_clk" />
1893  <connection
1894    kind="clock"
1895    version="13.0"
1896    start="ext_clk_50.clk"
1897    end="mm_clock_crossing_bridge_0.s0_clk" />
1898  <connection
1899    kind="reset"
1900    version="13.0"
1901    start="ext_clk_50.clk_reset"
1902    end="mm_clock_crossing_bridge_0.s0_reset" />
1903  <connection
1904    kind="reset"
1905    version="13.0"
1906    start="refclk_100.clk_reset"
1907    end="mem_if_ddr3_fpga.soft_reset" />
1908  <connection
1909    kind="reset"
1910    version="13.0"
1911    start="refclk_100.clk_reset"
1912    end="mem_if_ddr3_fpga.global_reset" />
1913  <connection
1914    kind="reset"
1915    version="13.0"
1916    start="refclk_100.clk_reset"
1917    end="mm_clock_crossing_bridge_0.m0_reset" />
1918  <connection
1919    kind="reset"
1920    version="13.0"
1921    start="ext_clk_50.clk_reset"
1922    end="ImgDiff_0.clk_in_reset" />
1923  <connection
1924    kind="clock"
1925    version="13.0"
1926    start="pll_0.outclk0"
1927    end="ImgDiff_0.clk_in" />
1928  <connection
1929    kind="avalon"
1930    version="13.0"
1931    start="ImgDiff_0.master_read"
1932    end="mem_if_ddr3_fpga.avl_0">
1933    <parameter name="arbitrationPriority" value="1" />
1934    <parameter name="baseAddress" value="0x0000" />
1935    <parameter name="defaultConnection" value="false" />
1936  </connection>
1937  <connection
1938    kind="avalon"
1939    version="13.0"
1940    start="ImgDiff_0.master_write"
1941    end="mem_if_ddr3_fpga.avl_0">
1942    <parameter name="arbitrationPriority" value="1" />
1943    <parameter name="baseAddress" value="0x0000" />
1944    <parameter name="defaultConnection" value="false" />
1945  </connection>
1946  <connection
1947    kind="avalon"
1948    version="13.0"
1949    start="hps_0.h2f_lw_axi_master"
1950    end="mm_clock_crossing_bridge_1.s0">
1951    <parameter name="arbitrationPriority" value="1" />
1952    <parameter name="baseAddress" value="0x00030000" />
1953    <parameter name="defaultConnection" value="false" />
1954  </connection>
1955  <connection
1956    kind="avalon"
1957    version="13.0"
1958    start="mm_clock_crossing_bridge_1.m0"
1959    end="ImgDiff_0.cs">
```

```

1960 <parameter name="arbitrationPriority" value="1" />
1961 <parameter name="baseAddress" value="0x0000" />
1962 <parameter name="defaultConnection" value="false" />
1963 </connection>
1964 <connection
1965   kind="reset"
1966   version="13.0"
1967   start="ext_clk_50.clk_reset"
1968   end="mm_clock_crossing_bridge_1.s0_reset" />
1969 <connection
1970   kind="clock"
1971   version="13.0"
1972   start="pll_0.outclk0"
1973   end="mm_clock_crossing_bridge_1.s0_clk" />
1974 <connection
1975   kind="clock"
1976   version="13.0"
1977   start="ext_clk_50.clk"
1978   end="mm_clock_crossing_bridge_1.m0_clk" />
1979 <connection
1980   kind="reset"
1981   version="13.0"
1982   start="ext_clk_50.clk_reset"
1983   end="mm_clock_crossing_bridge_1.m0_reset" />
1984 <connection
1985   kind="avalon"
1986   version="13.0"
1987   start="mm_clock_crossing_bridge_1.m0"
1988   end="IPTTest_0.csr">
1989 <parameter name="arbitrationPriority" value="1" />
1990 <parameter name="baseAddress" value="0x0800" />
1991 <parameter name="defaultConnection" value="false" />
1992 </connection>
1993 <connection
1994   kind="avalon"
1995   version="13.0"
1996   start="IPTTest_0.master_write"
1997   end="mem_if_ddr3_fpga.avl_0">
1998 <parameter name="arbitrationPriority" value="1" />
1999 <parameter name="baseAddress" value="0x0000" />
2000 <parameter name="defaultConnection" value="false" />
2001 </connection>
2002 <connection
2003   kind="avalon"
2004   version="13.0"
2005   start="IPTTest_0.master_read"
2006   end="mem_if_ddr3_fpga.avl_0">
2007 <parameter name="arbitrationPriority" value="1" />
2008 <parameter name="baseAddress" value="0x0000" />
2009 <parameter name="defaultConnection" value="false" />
2010 </connection>
2011 <connection
2012   kind="clock"
2013   version="13.0"
2014   start="pll_0.outclk0"
2015   end="IPTTest_0.clk_in" />
2016 <connection
2017   kind="reset"
2018   version="13.0"
2019   start="ext_clk_50.clk_reset"
2020   end="IPTTest_0.clk_in_reset" />
2021 <interconnectRequirement for="$system" name="qsys_mm.clockCrossingAdapter" value="FIFO" /
2022   >
2023 <interconnectRequirement for="$system" name="qsys_mm.maxAdditionalLatency" value="4" />
2024 <interconnectRequirement for="hps_only_master.master" name="qsys_mm.security" value="SECURE" />
</system>
```

Listing 16.2: Qsys System - Code View (orion_system.qsys)

16.3.4 Top Level Verilog (orion_top.v)

```
1 //-----//  
2 // Title : orion_top.v //  
3 // Rev: Rev 1 //  
4 //-----//  
5 // Description: Senior Design Top Level File (based on golden_top.v ) //  
6 //-----//  
7 // ****ORIGIONAL COMMENTS*****//  
8 //-----//  
9 // Title : golden_top.v //  
10 // Rev: Rev 5 //  
11 // Created: November 30, 2012 //  
12 //-----//  
13 // Description: Cyclone V SX SoC pinout and IO Standard example design //  
14 //-----//  
15 // Revision History: //  
16 // Rev 0: 1st cut //  
17 //-----//  
18 //----- 1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7 ----- 7  
19 //----- 0 ----- 0 ----- 0 ----- 0 ----- 0 ----- 0 ----- 0 ----- 8  
20 //-----//  
21 // Copyright 2012 Altera Corporation. All rights reserved. Altera products  
22 // are protected under numerous U.S. and foreign patents, maskwork rights,  
23 // copyrights and other intellectual property laws.  
24 //  
25 // This reference design file, and your use thereof, is subject to and  
26 // governed by the terms and conditions of the applicable Altera Reference  
27 // Design License Agreement. By using this reference design file, you  
28 // indicate your acceptance of such terms and conditions between you and  
29 // Altera Corporation. In the event that you do not agree with such terms and  
30 // conditions, you may not use the reference design file. Please promptly  
31 // destroy any copies you have made.  
32 //  
33 // This reference design file being provided on an "as-is" basis and as an  
34 // accommodation and therefore all warranties, representations or guarantees  
35 // of any kind (whether express, implied or statutory) including, without  
36 // limitation, warranties of merchantability, non-infringement, or fitness for  
37 // a particular purpose, are specifically disclaimed. By making this  
38 // reference design file available, Altera expressly does not recommend,  
39 // suggest or require that this reference design file be used in combination  
40 // with any other product not provided by Altera.  
41 //  
42 //  
43 //  
44 //define HPS  
45 //`define HPS_GPIO  
46 `define DDR3_FPGA  
47 `define DDR3_HPS  
48 `define HPS_USB  
49 //`define I2C_FPGA  
50 //`define MAX_FPGA  
51 //  
52 //`define SDI_FPGA  
53 `define ENET_FPGA  
54 //`define PCIE_FPGA  
55 //  
56 //`define HSMA_XCVR  
57 //`define HSMA_LVDS  
58 //`define HSMA_PARALLEL  
59 //`define HSMA_CMOS
```

```

61 // 'define PCIE_XCVR
62 // 'define SDI_XCVR
63
64 module orion_top (
65 //-----
66 /*
67 // User-IO-----//X pins //
68   output [3:0] user_led_fpga ,
69
70 //GPLL-CLK-----//X pins
71   input    clk_100m_fpga ,      //2.5V  //100 MHz (2nd copy to max)
72   input    clk_50m_fpga ,      //2.5V  //50MHz (2nd copy to max)
73
74   input    clk_top1 ,          //2.5V  //156.25 MHz adjustable
75   input    clk_bot1 ,          //1.5V  //100 MHz ajustable
76
77   output   ddr3_fpga_csn ,    //SSTL15 //Chip Select
78   input    cpu_resetn ,        //2.5V  //NIOS CPU Reset Pushbutton
79 */
80 //-----
81 //////////////// Hard Processor System (HPS) pins ///////////////////
82 //-----
83
84 `ifdef HPS
85 //HPS-CLK-INPUT-----//2 pins //
86   //input    clk_osc1 ,          //3.3V //Not for instantiation
87   //input    clk_osc2 ,          //3.3V //Not for instantiation
88
89 //HPS-Reset-----//2 pins //
90   //inout   mictor_rstn ,       //3.3V //Cold Reset
91   //input    hps_resetn ,       //3.3V //Warm Reset
92
93 //HPS-UART-----//2 pins //
94   input    uart_rx ,          //3.3V LV //UART Receive
95   output   uart_tx ,          //3.3V //UART Transmit
96
97
98 //HPS-I2C-----//2 pins //
99   inout   i2c_scl_hps ,       //3.3V //HPS I2C Clock output
100  inout   i2c_sda_hps ,       //3.3V //HPS I2C Data Input/Output
101
102 //HPS-SPI-Bus-----//4 pins //
103  output   spi_csn ,          //3.3V //Slave Sel 0 - LTC Analog
104  input    spi_miso ,          //3.3V //Master Input
105  output   spi_mosi ,          //3.3V //Master Output
106  output   spi_sck ,          //3.3V //Clock Output
107
108 //HPS-QSPI-Flash-----//6 pins //
109  output   qspi_clk ,          //3.3V //Clock
110  inout   [3:0] qspi_io ,       //3.3V //Data
111  output   qspi_ss0 ,          //3.3V //Select
112
113 //HPS-SD-Card-Flash-----//7 pins //
114  output   sd_clk ,           //3.3V //
115  inout   sd_cmd ,            //3.3V //
116  //output   sd_pwren ,          //3.3V //
117  inout   [3:0] sd_dat ,       //3.3V //????????????????? should only be (3:0)
118  //inout  [7:0] sd_dat ,       //3.3V //????????????????? should only be (3:0)
119
120
121 //HPS-Ethernet----10/100/1000----//14 pins //
122  output   enet_hps_gtx_clk ,  //3.3V //Gb Ethernet Clock
123  //input   enet_hps_intn ,     //3.3V //Placed on HPS GPIO
124  output   enet_hps_mdc ,      //3.3V //MDIO Clock (TR=0)
125  inout   enet_hps_mdio ,      //3.3V //MDIO Data (TR=0)
126  input    enet_hps_rx_clk ,   //3.3V //Receive Data

```

```

127    input      enet_hps_rx_dv ,      // 3.3V   // Receive Data Valid / Cont
128    input [3:0] enet_hps_rxd ,      // 3.3V   // Receive Data
129    output     enet_hps_tx_en ,      // 3.3V   // Transmit Data Enable / Cont
130    output [3:0] enet_hps_txd ,      // 3.3V   // Transmit Data
131
132 // HPS-CAN-BUS-----//2 pins //
133    input      can_0_rx ,          // 3.3V   //HPS only
134    output     can_0_tx ,          // 3.3V   //HPS only
135
136 // HPS-Trace-----//10 pins //
137    //input      mictror_rstn ,      // 3.3V   //????????????????? not in Qsys - needed?
138    output     trace_clk_mic ,      // 3.3V   //
139    output [7:0] trace_data ,      // 3.3V   //
140
141 'endif
142
143 'ifdef HPS_GPIO
144 // HPS-User-IO-----//X pins //
145    inout [3:0] user_dipsw_hps ,      // 1.5V
146    inout [3:0] user_led_hps ,      // 3.3V
147    inout [1:0] user_pb_hps ,      // 1.5V
148 'endif
149
150 'ifdef HPS_USB
151 // HPS-USB-OTG-----//19 pins //
152    input      usb_clk ,          // 3.3V   // Clock
153    inout [7:0] usb_data ,          // 3.3V   //
154    input      usb_nxt ,          // 3.3V   //
155    input      usb_dir ,          // 3.3V   //
156    output     usb_stp ,          // 3.3V   //
157    //input      usb_empty ,          //
158    //input      usb_full ,          //
159    //output     usb_oen ,          //
160    //output     usb_rdn ,          //
161    //output     usb_resetn ,        //
162    //output     usb_scl ,          //
163    //inout     usb_sda ,          //
164    //output     usb_wrn ,          //
165 'endif
166
167 // HPS-DDR3-400Mx40-----//78 pins //
168 'ifdef DDR3_HPS
169    output [14:0] ddr3_hps_a ,      // SSTL15 // Address
170    output [2:0] ddr3_hps_ba ,      // SSTL15 // Bank Address
171    output      ddr3_hps_casn ,      // SSTL15 // Column Address Strobe
172    output      ddr3_hps_cke ,      // SSTL15 // Clock Enable
173    output      ddr3_hps_clk_n ,      // SSTL15 // Diff Clock - Neg
174    output      ddr3_hps_clk_p ,      // SSTL15 // Diff Clock - Pos
175    output      ddr3_hps_csn ,      // SSTL15 // Chip Select
176    output [4:0] ddr3_hps_dm ,      // SSTL15 // Data Write Mask
177    inout [39:0] ddr3_hps_dq ,      // SSTL15 // Data Bus
178    inout [4:0] ddr3_hps_dqs_n ,      // SSTL15 // Diff Data Strobe - Neg
179    inout [4:0] ddr3_hps_dqs_p ,      // SSTL15 // Diff Data Strobe - Pos
180    output      ddr3_hps_odt ,      // SSTL15 // On-Die Termination Enable
181    output      ddr3_hps_rasn ,      // SSTL15 // Row Address Strobe
182    output      ddr3_hps_resetn ,      // SSTL15 // Reset
183    output      ddr3_hps_wen ,      // SSTL15 // Write Enable
184    input       ddr3_hps_rzq ,      // OCT_rzqin
185    //input      ddr3_rup ,          // OCT_rup
186    //input      ddr3_rdn ,          // OCT_rdn
187 'else
188    //output     ddr3_hps_csn ,      // SSTL15 // Chip Select
189 'endif
190
191
192 // -----
193 // //////////////////////////////////////////////////////////////////// FPGA Pins ////////////////////////////////////////////////////////////////////
194 // -----

```

```

195
196
197 //FPGA-GPLL-CLK-----//X pins
198 input clk_100m_fpga , //2.5V //100 MHz (2nd copy to max)
199 input clk_50m_fpga , //2.5V //50MHz (2nd copy to max)
200 input clk_25m_fpga ,
201 input clk_top1 , //2.5V //156.25 MHz adjustable
202 input clk_bot1 , //1.5V //100 MHz adjustable
203 // input clk_enet_fpga_p , //LVDS //125 MHz fixed
204
205 //FPGA-User-IO-----//11 pins //
206 input cpu_resetn , //2.5V //NIOS CPU Reset Pushbutton
207 input [3:0] user_dipsw_fpga , //
208 output [3:0] user_led_fpga , //
209 input [1:0] user_pb_fpga , //
210
211 //FPGA-DDR3-400Mx32-----//71 pins //
212 'ifdef DDR3_FPGA
213 output [14:0] ddr3_fpga_a , //SSTL15 //Address
214 output [2:0] ddr3_fpga_ba , //SSTL15 //Bank Address
215 output ddr3_fpga_casn , //SSTL15 //Column Address Strobe
216 output ddr3_fpga_cke , //SSTL15 //Clock Enable
217 output ddr3_fpga_clk_n , //SSTL15 //Diff Clock - Neg
218 output ddr3_fpga_clk_p , //SSTL15 //Diff Clock - Pos
219 output ddr3_fpga_csn , //SSTL15 //Chip Select
220 output [3:0] ddr3_fpga_dm , //SSTL15 //Data Write Mask
221 inout [31:0] ddr3_fpga_dq , //SSTL15 //Data Bus
222 inout [3:0] ddr3_fpga_dqs_n , //SSTL15 //Diff Data Strobe - Neg
223 inout [3:0] ddr3_fpga_dqs_p , //SSTL15 //Diff Data Strobe - Pos
224 output ddr3_fpga_odt , //SSTL15 //On-Die Termination Enable
225 input ddr3_fpga_rasn , //SSTL15 //Row Address Strobe
226 input ddr3_fpga_resetn , //SSTL15 //Reset
227 input ddr3_fpga_wen , //SSTL15 //Write Enable
228 input ddr3_fpga_rzq , //OCT_rzqin
229 'else
230 output ddr3_fpga_csn , //SSTL15 //Chip Select
231 'endif
232
233
234 //FPGA-Ethernet1 ---10/100-----//14 pins //
235 'ifdef ENET_FPGA
236 input enet1_rx_clk , //2.5V //Receive Data Clock
237 input [3:0] enet1_rx_d , //2.5V //Receive Data
238 input enet1_rx_dv , //2.5V //Receive Data Valid
239 input enet1_rx_error , //2.5V //Receive Data Error
240 output enet1_tx_clk_fb , //2.5V //Transmit Clock Feedback
241 output [3:0] enet1_tx_d , //2.5V //Transmit Data
242 output enet1_tx_en , //2.5V //Transmit Data Enable
243 input enet1_tx_error , //2.5V //Transmit Data Error
244 output enet_dual_resetn , //2.5V //EtherCat PHY Reset
245 input enet2_rx_clk , //2.5V //Receive Data Clock
246 input [3:0] enet2_rx_d , //2.5V //Receive Data
247 input enet2_rx_dv , //2.5V //Receive Data Valid
248 input enet2_rx_error , //2.5V //Receive Data Error
249 output enet2_tx_clk_fb , //2.5V //Transmit Clock Feedback
250 output [3:0] enet2_tx_d , //2.5V //Transmit Data
251 output enet2_tx_en , //2.5V //Transmit Data Enable
252 input enet2_tx_error , //2.5V //Transmit Data Error
253 input enet_fpga_mdc ,
254 input enet_fpga_mdio
255 'endif
256
257 //SDI-XCVR-Video-----//X pins //
258 'ifdef use_SDI_XCVR_refclk_p
259 input clk_148_p , //LVDS //148.5MHz Prog. SDI VCXO
260 'endif
261 'ifdef SDI_XCVR
262 input gxb_rx_14_p , //PCML //SDI Receiver (or SMA)

```

```

263     output      gxb_tx_14_p ,           //PCML    //SDI Transmitter (or SMA)
264 'endif
265
266 'ifdef SDI_FPGA
267 // 148.5M programmable VCXO in refclk section above
268     output      sdi_clk148_dn ,        // 2.5V   //VCXO pump up
269     output      sdi_clk148_up ,        // 2.5V   //VCXO pump down
270     input       sdi_fault ,          // 2.5V   //SDI Cable Driver Fault
271     output      sdi_rsti ,           // 2.5V   //SDI Cable Driver Reset
272     output      sdi_rx_bypass ,       // 2.5V   //SDI Equalizer Bypass
273     output      sdi_rx_en ,          // 2.5V   //SDI Equalizer Enable
274     output      sdi_tx_en ,           // 2.5V   //SDI Cable Driver Enable
275     output      sdi_tx_sd_hdn ,       // 2.5V   //SDI Cable Driver High-Def
276 'endif
277
278 'ifdef I2C_FPGA
279     inout      i2c_sda_fpga ,
280     inout      i2c_scl_fpga ,
281 'endif
282
283
284 'ifdef MAX_FPGA
285     output      max_fpga_miso ,
286     input       max_fpga_mosi ,
287     inout      max_fpga_sck ,
288     inout      max_fpga_sscl ,
289 'endif
290
291
292 //HSMC-Port-A-----//107pins //
293 'ifdef use_HSMA_XCVR_refclk_p
294     input       refclk_q12_p ,         //LVDS   //HSMA Transceiver Refclk -reqs OCT
295 'endif
296 'ifdef HSMA_XCVR
297     input [3:0]  hsma_rx_p ,          //PCML?  //HSMA Receive Data -reqs OCT
298     output [3:0]  hsma_tx_p ,          //PCML?  //HSMA Transmit Data
299 'endif
300
301 'ifdef HSMA_CMOS
302 // Enable below for CMOS HSMC
303 //inout [79:0] hsma_d ,             //2.5V   //HSMA CMOS Data Bus
304 // Enable below for LVDS HSMC
305     input       hsma_clk_in0 ,        //2.5V   //Primary single-ended CLKIN
306     input       hsma_clk_in_p1 ,       // LVCMS signal
307     input       hsma_clk_in_n1 ,       // LVCMS signal
308     input       hsma_clk_in_p2 ,       //LVDS   //Primary Source-Sync CLKIN
309     output      hsma_clk_out0 ,        //2.5V   //Primary single-ended CLKOUT
310     output      hsma_clk_out_p1 ,       // LVCMS signal
311     output      hsma_clk_out_n1 ,       // LVCMS signal
312     output      hsma_clk_out_p2 ,       //LVDS   //Primary Source-Sync CLKOUT
313     inout      [3:0]  hsma_d ,          //2.5V   //Dedicated CMOS IO
314     input      [16:0] hsma_rx_d_p ,       //LVDS   //LVDS Source-Sync Input
315     output     [16:0] hsma_tx_d_p ,       //LVDS   //LVDS Source-Sync Output
316     input       hsma_prsntn ,          //2.5V   //HSMC Presence Detect Input
317     output      hsma_scl ,            //2.5V   //SMBus Clock
318     inout      hsma_sda ,            //2.5V   //SMBus Data
319 'endif
320
321 // PCI-Express -----//X pins //
322 'ifdef use_PCIE_XCVR_refclk_p
323     input       pcie_refclk_p ,        //LVDS   //PCIe Refclk -reqs OCT
324 'endif
325 'ifdef PCIE_XCVR
326     input [3:0]  pcie_rx_p ,          //PCML   //PCIe Receive Data-req'sOCT
327     output [3:0]  pcie_tx_p ,          //PCML   //PCIe Transmit Data
328 'endif
329 //---Found multiple nets with the same name
330 'ifdef PCIE_FPGA

```

```

331     input      pcie_perstn_in ,    // Pin W21 instance 0
332     output     pcie_perstn_out ,   // Pin AG6 instance 1
333     input      pcie_prsnt2_x1 ,
334     input      pcie_prsnt2_x4 ,
335     inout     pcie_smbclk ,
336     inout     pcie_smbdat ,
337     inout     pcie_waken
338 `endif
339 );
340 );
341
342 // Rev A board Ethernet reset can interrupt device configuration, to fix
343 // enet_dual_reset tristate as input or drive as output to 1
344 assign enet_dual_resetn = 1'b1;
345 // FPGA user dipswitches drive user LEDs
346 // assign user_led_fpga = ~user_dipsw_fpga;
347 // Disable FPGA DDR3
348 // assign ddr3_fpga_csn = 1'b1;
349
350
351
352
353
354 //*****From golden_top by Altera Corp.*****
355 // internal wires and registers declaration
356 wire [1:0] fpga_debounced_buttons;
357 wire [3:0] fpga_led_internal;
358 wire      hps_fpga_reset_n;
359
360 // connection of internal logics
361 // assign fpga_led_internal = user_led_fpga;
362
363
364 // Debounce logic to clean out glitches within 1ms
365 debounce debounce_inst (
366   .clk                      (clk_50m_fpga),
367   .reset_n                  (hps_fpga_reset_n),
368   .data_in                  (user_pb_fpga[0]),
369   .data_out                 (fpga_debounced_buttons)
370 );
371 defparam debounce_inst.WIDTH = 2;
372 defparam debounce_inst.POLARITY = "LOW";
373 defparam debounce_inst.TIMEOUT = 50000;           // at 50Mhz this is a debounce
374           time of 1ms
375 defparam debounce_inst.TIMEOUT_WIDTH = 16;        // ceil(log2(TIMEOUT))
376
377 // Due to PLLs the clock pins need to be buffered when connected internal clock network
378 // wire clk_50m_net;
379 // wire clk_100m_net;
380
381 // clk_cntrl clk_cntrl_50m (
382 //   .inclk ( clk_50m_fpga ),
383 //   .outclk ( clk_50m_net )
384 // );
385 // clk_cntrl clk_cntrl_100m (
386 //   .inclk ( clk_100m_fpga ),
387 //   .outclk ( clk_100m_net )
388 // );
389
390 // **** IMPORTANT NOTE ****
391 // Need to use clk_bott1 (100 Mhz (adjustable) to drive PLL due to fitter constraint
392 // DO NOT use clk_100m_fpga as it is in the wrong IO block for the FPGA
393
394 orion_system u0 (
395   .memory_mem_a              (ddr3_hps_a),
396   //                                memory.mem_a
397   .memory_mem_ba             (ddr3_hps_ba),
398   //                                .mem.ba

```

```

396     .memory_mem_ck           // (ddr3_hps_clk_p), .mem_ck
397     .memory_mem_ck_n         // (ddr3_hps_clk_n), .mem_ck_n
398     .memory_mem_cke          // .mem_cke
399     .memory_mem_csn          // (ddr3_hps_csn), .mem_csn
400     .memory_mem_rasn          // (ddr3_hps_rasn), .mem_rasn
401     .memory_mem_casn          // (ddr3_hps_casn), .mem_casn
402     .memory_mem_wen          // (ddr3_hps_wen), .mem_wen
403     .memory_mem_resetn        // (ddr3_hps_resetn), .mem_resetn
404     .memory_mem_dq           // (ddr3_hps_dq), .mem_dq
405     .memory_mem_dqs          // (ddr3_hps_dqs_p), .mem_dqs
406     .memory_mem_dqs_n         // (ddr3_hps_dqs_n), .mem_dqs_n
407     .memory_mem_odt          // (ddr3_hps_odt), .mem_odt
408     .memory_mem_dm           // (ddr3_hps_dm), .mem_dm
409     .memory_oct_rzqin        // (ddr3_hps_rzq), .oct_rzqin
410     .hps_0_hps_io_hps_io_emac1_inst_TX_CLK (enet_hps_gtx_clk), // hps_0_hps_io.hps_io_emac1_inst_TX_CLK
411     .hps_0_hps_io_hps_io_emac1_inst_RXDO (enet_hps_txd[0]), // .hps_io_emac1_inst_RXDO
412     .hps_0_hps_io_hps_io_emac1_inst_TXD1 (enet_hps_txd[1]), // .hps_io_emac1_inst_TXD1
413     .hps_0_hps_io_hps_io_emac1_inst_TXD2 (enet_hps_txd[2]), // .hps_io_emac1_inst_TXD2
414     .hps_0_hps_io_hps_io_emac1_inst_TXD3 (enet_hps_txd[3]), // .hps_io_emac1_inst_TXD3
415     .hps_0_hps_io_hps_io_emac1_inst_RXDO (enet_hps_rxd[0]), // .hps_io_emac1_inst_RXDO
416     .hps_0_hps_io_hps_io_emac1_inst_MDIO (enet_hps_mdio), // .hps_io_emac1_inst_MDIO
417     .hps_0_hps_io_hps_io_emac1_inst_MDC (enet_hps_mdc), // .hps_io_emac1_inst_MDC
418     .hps_0_hps_io_hps_io_emac1_inst_RX_CTL (enet_hps_rx_dv), // .hps_io_emac1_inst_RX_CTL
419     .hps_0_hps_io_hps_io_emac1_inst_TX_CTL (enet_hps_tx_en), // .hps_io_emac1_inst_TX_CTL
420     .hps_0_hps_io_hps_io_emac1_inst_RX_CLK (enet_hps_rx_clk), // .hps_io_emac1_inst_RX_CLK
421     .hps_0_hps_io_hps_io_emac1_inst_RXD1 (enet_hps_rxd[1]), // .hps_io_emac1_inst_RXD1
422     .hps_0_hps_io_hps_io_emac1_inst_RXD2 (enet_hps_rxd[2]), // .hps_io_emac1_inst_RXD2
423     .hps_0_hps_io_hps_io_emac1_inst_RXD3 (enet_hps_rxd[3]), // .hps_io_emac1_inst_RXD3
424     .hps_0_hps_io_hps_io_qspi_inst_IO0 (qspi.io[0]), // .hps_io_qspi_inst_IO0
425     .hps_0_hps_io_hps_io_qspi_inst_IO1 (qspi.io[1]), // .hps_io_qspi_inst_IO1
426     .hps_0_hps_io_hps_io_qspi_inst_IO2 (qspi.io[2]), // .hps_io_qspi_inst_IO2
427     .hps_0_hps_io_hps_io_qspi_inst_IO3 (qspi.io[3]), // .hps_io_qspi_inst_IO3
428     .hps_0_hps_io_hps_io_qspi_inst_SS0 (qspi_ss0), // .hps_io_qspi_inst_SS0
429     .hps_0_hps_io_hps_io_qspi_inst_CLK (qspi.clk), // .hps_io_qspi_inst_CLK

```

```

430 .hps_0.hps_io.hps_io_sdio_inst_CMD      (sd_cmd),           //  

431 // .hps_0.hps_io.hps_io_sdio_inst_PWREN   (sd_pwren),          //  

432 .hps_0.hps_io.hps_io_sdio_inst_D0       (sd_dat[0]),          //  

433 .hps_0.hps_io.hps_io_sdio_inst_D1       (sd_dat[1]),          //  

434 .hps_0.hps_io.hps_io_sdio_inst_CLK      (sd_clk),             //  

435 .hps_0.hps_io.hps_io_sdio_inst_D2       (sd_dat[2]),          //  

436 .hps_0.hps_io.hps_io_sdio_inst_D3       (sd_dat[3]),          //  

437 .hps_0.hps_io.hps_io_usb1_inst_D0       (usb_data[0]),         //  

438 .hps_0.hps_io.hps_io_usb1_inst_D1       (usb_data[1]),         //  

439 .hps_0.hps_io.hps_io_usb1_inst_D2       (usb_data[2]),         //  

440 .hps_0.hps_io.hps_io_usb1_inst_D3       (usb_data[3]),         //  

441 .hps_0.hps_io.hps_io_usb1_inst_D4       (usb_data[4]),         //  

442 .hps_0.hps_io.hps_io_usb1_inst_D5       (usb_data[5]),         //  

443 .hps_0.hps_io.hps_io_usb1_inst_D6       (usb_data[6]),         //  

444 .hps_0.hps_io.hps_io_usb1_inst_D7       (usb_data[7]),         //  

445 .hps_0.hps_io.hps_io_usb1_inst_CLK      (usb_clk),            //  

446 .hps_0.hps_io.hps_io_usb1_inst_STP      (usb_stp),             //  

447 .hps_0.hps_io.hps_io_usb1_inst_DIR      (usb_dir),             //  

448 .hps_0.hps_io.hps_io_usb1_inst_NXT      (usb_nxt),             //  

449 .hps_0.hps_io.hps_io_spim0_inst_CLK     (spi_sck),            //  

450 .hps_0.hps_io.hps_io_spim0_inst_MOSI    (spi_mosi),           //  

451 .hps_0.hps_io.hps_io_spim0_inst_MISO    (spi_miso),           //  

452 .hps_0.hps_io.hps_io_spim0_inst_SS0     (spi_csn),            //  

453 .hps_0.hps_io.hps_io_uart0_inst_RX      (uart_rx),             //  

454 .hps_0.hps_io.hps_io_uart0_inst_TX      (uart_tx),             //  

455 .hps_0.hps_io.hps_io_i2c0_inst_SDA     (i2c_sda_hps),          //  

456 .hps_0.hps_io.hps_io_i2c0_inst_SCL     (i2c_scl_hps),          //  

457 .hps_0.hps_io.hps_io_can0_inst_RX      (can_0_rx),             //  

458 .hps_0.hps_io.hps_io_can0_inst_TX      (can_0_tx),             //  

459 .hps_0.hps_io.hps_io_trace_inst_CLK     (trace_clk_mic),        //  

460 .hps_0.hps_io.hps_io_trace_inst_D0      (trace_data[0]),         //  

461 .hps_0.hps_io.hps_io_trace_inst_D1      (trace_data[1]),         //  

462 .hps_0.hps_io.hps_io_trace_inst_D2      (trace_data[2]),         //  

463 .hps_0.hps_io.hps_io_trace_inst_D3      (trace_data[3]),         //

```

```

464     .hps_0.hps_io.hps_io_trace_inst_D4      (trace_data[4]),          //  

465     .hps_0.hps_io.hps_io_trace_inst_D5      (trace_data[5]),          //  

466     .hps_0.hps_io.hps_io_trace_inst_D6      (trace_data[6]),          //  

467     .hps_0.hps_io.hps_io_trace_inst_D7      (trace_data[7]),          //  

468 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO04    (user_dipsw_hps[0]),      //  

469 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO05    (user_dipsw_hps[1]),      //  

470 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO06    (user_dipsw_hps[2]),      //  

471 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO07    (user_dipsw_hps[3]),      //  

472 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO08    (user_pb_hps[0]),        //  

473 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO09    (user_pb_hps[1]),        //  

474 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO41    (user_led_hps[3]),       //  

475 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO42    (user_led_hps[2]),       //  

476 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO43    (user_led_hps[1]),       //  

477 //     .hps_0.hps_io.hps_io_gpio_inst_GPIO44    (user_led_hps[0]),       //  

478     .clk_50_clk                           (clk_50m_fpga),           clk_50.clk  

479     .led_pio_external_connection_in_port   (user_led_fpga),           //  

480     .led_pio_external_connection_out_port  (user_led_fpga),           //  

481     .dipsw_pio_external_connection_export (user_dipsw_fpga),           //  

482     .button_pio_external_connection_export (fpga_debounced_buttons), //  

483     .hps_0.h2f_reset.reset_n              (hps_fpga_reset_n),         //  

484     .fpga_memory.mem_a                  (ddr3_fpga_a),             //  

485     .fpga_memory.mem_ba                (ddr3_fpga_ba),             //  

486     .fpga_memory.mem_ck                (ddr3_fpga_clk_p),           mem_ck  

487     .fpga_memory.mem_ck_n              (ddr3_fpga_clk_n),           mem_ck_n  

488     .fpga_memory.mem_cke               (ddr3_fpga_cke),             mem_cke  

489     .fpga_memory.mem_csn               (ddr3_fpga_csn),             mem_csn  

490     .fpga_memory.mem_dm               (ddr3_fpga_dm),             mem_dm  

491     .fpga_memory.mem_ras_n            (ddr3_fpga_rasn),            mem_ras_n  

492     .fpga_memory.mem_cas_n            (ddr3_fpga_casn),            mem_cas_n  

493     .fpga_memory.mem_we_n             (ddr3_fpga_wen),             mem_we_n  

494     .fpga_memory.mem_reset_n          (ddr3_fpga_resetn),           mem_reset_n  

495     .fpga_memory.mem_dq               (ddr3_fpga_dq),              mem_dq  

496     .fpga_memory.mem_dqs              (ddr3_fpga_dqs_p),            mem_dqs  

497     .fpga_memory.mem_dqs_n            (ddr3_fpga_dqs_n),            mem_dqs_n

```

```

498     .fpga_memory_mem_odt          (ddr3_fpga_odt),           //
499     .fpga_memory_oct_rzqin      .mem_odt                  (ddr3_fpga_rzq),           //
500     .refclk_100_clk             fpga_memory_oct.rzqin      (clk_bot1),           //
501     .reset_50_reset_n          refclk_100.clk            (hps_fpga_reset_n),       //
502     .mem_if_ddr3_fpga_status_local_init_done //               reset_50.reset_n        //
503     .mem_if_ddr3_fpga_status_local_init_done mem_if_ddr3_fpga_status.local_init_done( ),           //
504     .mem_if_ddr3_fpga_status_local_cal_success //             local_cal_success       //
505     .mem_if_ddr3_fpga_status_local_cal_fail   .local_cal_success( ),           //
506     .mem_if_ddr3_fpga_pll_sharing_pll_mem_clk mem_if_ddr3_fpga_pll_sharing pll_mem_clk( ),           //
507     .mem_if_ddr3_fpga_pll_sharing_pll_write_clk .pll_write_clk( ),           //
508     .mem_if_ddr3_fpga_pll_sharing_pll_addr_cmd_clk .pll_write_clk.pre_phy_clk( ),           //
509     .mem_if_ddr3_fpga_pll_sharing_pll_locked    .pll_addr_cmd_clk( ),           //
510     .mem_if_ddr3_fpga_pll_sharing_pll_locked    .pll_locked( ),           //
511     .mem_if_ddr3_fpga_pll_sharing_pll_avl_clk   .pll_locked( ),           //
512     .mem_if_ddr3_fpga_pll_sharing_pll_config_clk .pll_avl_clk( ),           //
513     .mem_if_ddr3_fpga_pll_sharing_pll_mem_phy_clk .pll_config_clk( ),           //
514     .mem_if_ddr3_fpga_pll_sharing_afi_phy_clk   .pll_mem_phy_clk( ),           //
515     .mem_if_ddr3_fpga_pll_sharing_pll_avl_phy_clk .afi_phy_clk( ),           //
516     .mem_if_ddr3_fpga_pll_sharing_pll_avl_phy_clk .pll_avl_phy_clk( ),           //
517
518 );
519 endmodule

```

Listing 16.3: Top Level Verilog (orion_top.v)

16.3.5 Timing Constraint File (timing.base.sdc)

```

1 #Timing constraint commands from DDR3 Demo from Board Test System by Altera Corporation
2 #Some Modifications by Christopher Yarp
3
4 derive_pll_clocks
5 derive_clock_uncertainty
6
7 #set_false_path -from * -to {sld_signaltap:auto_signaltap_0[*]}
8 #set_false_path -from {sld_signaltap:auto_signaltap_0[*]} -to *
9
10 #set sys_clock_hmc {u0|pll_0|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk}
11
12 #set_false_path -from [get_ports hps_fpga_reset_n] -to *
13
14 # Timing file copied from Holden Hardware Reference Design (GHRD)
15 #modified for ports in use
16
17 # 50MHz board input clock
18 create_clock -period 20 [get_ports clk_50m_fpga]
19 create_clock -period 10 [get_ports clk_100m_fpga]
20
21 # for enhancing USB BlasterII to be reliable, 25MHz
22 create_clock -name {altera_reserved_tck} -period 40 {altera_reserved_tck}

```

```

23 set_input_delay -clock altera_reserved_tck -clock_fall 3 [get_ports altera_reserved_tdi]
24 set_input_delay -clock altera_reserved_tck -clock_fall 3 [get_ports altera_reserved_tms]
25 set_output_delay -clock altera_reserved_tck -clock_fall 3 [get_ports altera_reserved_tdo]
26
27 # FPGA IO port constraints
28 set_false_path -from [get_ports {fpga_button_pio[0]}] -to *
29 set_false_path -from [get_ports {fpga_button_pio[1]}] -to *
30 set_false_path -from [get_ports {fpga_dipsw_pio[0]}] -to *
31 set_false_path -from [get_ports {fpga_dipsw_pio[1]}] -to *
32 set_false_path -from [get_ports {fpga_dipsw_pio[2]}] -to *
33 set_false_path -from [get_ports {fpga_dipsw_pio[3]}] -to *
34 set_false_path -from * -to [get_ports {fpga_led_pio[0]}]
35 set_false_path -from * -to [get_ports {fpga_led_pio[1]}]
36 set_false_path -from * -to [get_ports {fpga_led_pio[2]}]
37 set_false_path -from * -to [get_ports {fpga_led_pio[3]}]
38
39 # HPS peripherals port false path setting to workaround the unconstrained path (setting
40 # false-path for hps_0 ports will not affect the routing as it is hard silicon)
41 # **** outputs ****
42 set_false_path -from * -to [get_ports {uart_tx}]
43
44 set_false_path -from * -to [get_ports {spi_csn}]
45 set_false_path -from * -to [get_ports {spi_mosi}]
46 set_false_path -from * -to [get_ports {spi_sck}]
47
48 set_false_path -from * -to [get_ports {qspi_clk}]
49 set_false_path -from * -to [get_ports {qspi_ss0}]
50
51 set_false_path -from * -to [get_ports {sd_clk}]
52 set_false_path -from * -to [get_ports {sd_pwren}]
53
54 set_false_path -from * -to [get_ports {enet_hps_gtx_clk}]
55 set_false_path -from * -to [get_ports {enet_hps_mdc}]
56 set_false_path -from * -to [get_ports {enet_hps_tx_en}]
57 set_false_path -from * -to [get_ports {enet_hps_txd[0]}]
58 set_false_path -from * -to [get_ports {enet_hps_txd[1]}]
59 set_false_path -from * -to [get_ports {enet_hps_txd[2]}]
60 set_false_path -from * -to [get_ports {enet_hps_txd[3]}]
61
62 set_false_path -from * -to [get_ports {can_0_tx}]
63
64 set_false_path -from * -to [get_ports {trace_clk_mic}]
65 set_false_path -from * -to [get_ports {trace_data[0]}]
66 set_false_path -from * -to [get_ports {trace_data[1]}]
67 set_false_path -from * -to [get_ports {trace_data[2]}]
68 set_false_path -from * -to [get_ports {trace_data[3]}]
69 set_false_path -from * -to [get_ports {trace_data[4]}]
70 set_false_path -from * -to [get_ports {trace_data[5]}]
71 set_false_path -from * -to [get_ports {trace_data[6]}]
72 set_false_path -from * -to [get_ports {trace_data[7]}]
73
74 # **** inputs ****
75 set_false_path -from [get_ports {uart_rx}] -to *
76 set_false_path -from [get_ports {spi_miso}] -to *
77
78 set_false_path -from [get_ports {enet_hps_rx_clk}] -to *
79 set_false_path -from [get_ports {enet_hps_rx_dv}] -to *
80 set_false_path -from [get_ports {enet_hps_rxd[0]}] -to *
81 set_false_path -from [get_ports {enet_hps_rxd[1]}] -to *
82 set_false_path -from [get_ports {enet_hps_rxd[2]}] -to *
83 set_false_path -from [get_ports {enet_hps_rxd[3]}] -to *
84
85 set_false_path -from [get_ports {can_0_rx}] -to *
86
87 # **** inout ****
88 set_false_path -from * -to [get_ports {i2c_scl_hps}]
89 set_false_path -from [get_ports {i2c_scl_hps}] -to *
90 set_false_path -from * -to [get_ports {i2c_sda_hps}]

```

```

90| set_false_path -from [get_ports {i2c_sda_hps}] -to *
91|
92| set_false_path -from * -to [get_ports {qspi_io[0]}]
93| set_false_path -from [get_ports {qspi_io[0]}] -to *
94| set_false_path -from * -to [get_ports {qspi_io[1]}]
95| set_false_path -from [get_ports {qspi_io[1]}] -to *
96| set_false_path -from * -to [get_ports {qspi_io[2]}]
97| set_false_path -from [get_ports {qspi_io[2]}] -to *
98| set_false_path -from * -to [get_ports {qspi_io[3]}]
99| set_false_path -from [get_ports {qspi_io[3]}] -to *
100|
101| set_false_path -from * -to [get_ports {sd_cmd}]
102| set_false_path -from [get_ports {sd_cmd}] -to *
103| set_false_path -from * -to [get_ports {sd_dat[0]}]
104| set_false_path -from [get_ports {sd_dat[0]}] -to *
105| set_false_path -from * -to [get_ports {sd_dat[1]}]
106| set_false_path -from [get_ports {sd_dat[1]}] -to *
107| set_false_path -from * -to [get_ports {sd_dat[2]}]
108| set_false_path -from [get_ports {sd_dat[2]}] -to *
109| set_false_path -from * -to [get_ports {sd_dat[3]}]
110| set_false_path -from [get_ports {sd_dat[3]}] -to *
111|
112| set_false_path -from * -to [get_ports {enet_hps_mdio}]
113| set_false_path -from [get_ports {enet_hps_mdio}] -to *
114|
115#-----Orig-----
116|
117## Copyright (C) 1991–2013 Altera Corporation
118## Your use of Altera Corporation's design tools, logic functions
119## and other software and tools, and its AMPP partner logic
120## functions, and any output files from any of the foregoing
121## (including device programming or simulation files), and any
122## associated documentation or information are expressly subject
123## to the terms and conditions of the Altera Program License
124## Subscription Agreement, Altera MegaCore Function License
125## Agreement, or other applicable license agreement, including,
126## without limitation, that your use is for the sole purpose of
127## programming logic devices manufactured by Altera and sold by
128## Altera or its authorized distributors. Please refer to the
129## applicable agreement for further details.
130|
131#set_false_path -from [get_keepers {*user_dipsw*}]
132#set_false_path -from [get_keepers {*user_pb*}]
133#set_false_path -to [get_keepers {*user_led*}]
134|
135|
136#*****#
137# Set Multicycle Path
138#*****#
139|
140|
141|
142#*****#
143# Set Maximum Delay
144#*****#
145|
146|
147|
148#*****#
149# Set Minimum Delay
150#*****#
151|
152|
153|
154#*****#
155# Set Input Transition
156#*****#

```

Listing 16.4: Timing Constraint File (timing.base.sdc)

16.3.6 Quartus II Project File (orion_system.qpf)

```
1 # ----- #
2 #
3 # Copyright (C) 1991–2012 Altera Corporation
4 # Your use of Altera Corporation's design tools, logic functions
5 # and other software and tools, and its AMPP partner logic
6 # functions, and any output files from any of the foregoing
7 # (including device programming or simulation files), and any
8 # associated documentation or information are expressly subject
9 # to the terms and conditions of the Altera Program License
10 # Subscription Agreement, Altera MegaCore Function License
11 # Agreement, or other applicable license agreement, including,
12 # without limitation, that your use is for the sole purpose of
13 # programming logic devices manufactured by Altera and sold by
14 # Altera or its authorized distributors. Please refer to the
15 # applicable agreement for further details.
16 #
17 # -----
18 #
19 # Quartus II 64-Bit
20 # Version 13.0 Internal Build 69 11/14/2012 SJ Full Version
21 # Date created = 18:45:45 November 15, 2012
22 #
23 #
24 QUARTUS_VERSION = "13.0"
25 DATE = "18:45:45 November 15, 2012"
26 #
27 #
28 # Revisions
29 #
30 PROJECT_REVISION = "orion_system"
```

Listing 16.5: Quartus II Project File (orion_system.qpf)

16.3.7 Quartus II Settings File (orion_system.qsf)

```
1 # -----
2 #
3 # Copyright (C) 1991–2012 Altera Corporation
4 # Your use of Altera Corporation's design tools, logic functions
5 # and other software and tools, and its AMPP partner logic
6 # functions, and any output files from any of the foregoing
7 # (including device programming or simulation files), and any
8 # associated documentation or information are expressly subject
9 # to the terms and conditions of the Altera Program License
10 # Subscription Agreement, Altera MegaCore Function License
11 # Agreement, or other applicable license agreement, including,
12 # without limitation, that your use is for the sole purpose of
13 # programming logic devices manufactured by Altera and sold by
14 # Altera or its authorized distributors. Please refer to the
15 # applicable agreement for further details.
16 #
17 # -----
18 #
19 # Quartus II 64-Bit
20 # Version 13.0 Internal Build 69 11/14/2012 SJ Full Version
21 # Date created = 18:45:45 November 15, 2012
22 #
23 # -----
```

```

24 #
25 # Notes:
26 #
27 # 1) The default values for assignments are stored in the file:
28 #     bts_xcvr_assignment_defaults.qdf
29 #     If this file doesn't exist, see file:
30 #     assignment_defaults.qdf
31 #
32 # 2) Altera recommends that you do not modify this file. This
33 #     file is updated automatically by the Quartus II software
34 #     and any changes you make may be lost or overwritten.
35 #
36 # -----
37
38 set_global_assignment -name FAMILY "Cyclone V"
39 set_global_assignment -name DEVICE 5CSXFC6D6F31C8ES
40 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 12.1
41 set_global_assignment -name PROJECT_CREATION_TIME_DATE "22:55:46 OCTOBER 03, 2012"
42 set_global_assignment -name LAST_QUARTUS_VERSION 13.0
43 set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files_sof_rpt
44 set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 256
45 set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim-Altera (Verilog)"
46 set_global_assignment -name EDA_TIME_SCALE "1 ps" -section_id eda_simulation
47 set_global_assignment -name EDA_OUTPUT_DATA_FORMAT "VERILOG HDL" -section_id
    eda_simulation
48 set_global_assignment -name TOP_LEVEL_ENTITY orion_top
49 set_global_assignment -name POWER_PRESET_COOLING_SOLUTION "23 MM HEAT SINK WITH 200 LFPM
    AIRFLOW"
50 set_global_assignment -name POWER_BOARD_THERMAL_MODEL "NONE (CONSERVATIVE)"
51 set_global_assignment -name UNIPHYSSEQUENCER.DQS_CONFIG_ENABLE ON
52 set_global_assignment -name OPTIMIZE_MULTI_CORNER_TIMING ON
53 set_global_assignment -name ECO_REGENERATE_REPORT ON
54 set_global_assignment -name STRATIX_DEVICE_IO_STANDARD "2.5 V"
55 set_global_assignment -name STRATIXV_CONFIGURATION_SCHEME "PASSIVE PARALLEL X16"
56 set_global_assignment -name USE_CONFIGURATION_DEVICE OFF
57 set_global_assignment -name GENERATE_RBF_FILE ON
58 set_global_assignment -name GENERATE_HEX_FILE ON
59 set_global_assignment -name CRC_ERROR_OPEN_DRAIN ON
60 set_global_assignment -name ON_CHIP_BITSTREAM_DECOMPRESSION OFF
61 set_global_assignment -name RESERVE_DATA15_THROUGH_DATA8_AFTER_CONFIGURATION "AS INPUT
    TRI-STATE"
62 set_global_assignment -name RESERVE_DATA7_THROUGH_DATA5_AFTER_CONFIGURATION "AS INPUT
    TRI-STATE"
63 set_global_assignment -name OUTPUT_IO_TIMING_NEAR_END_VMEAS "HALF VCCIO" -rise
64 set_global_assignment -name OUTPUT_IO_TIMING_NEAR_END_VMEAS "HALF VCCIO" -fall
65 set_global_assignment -name OUTPUT_IO_TIMING_FAR-END_VMEAS "HALF SIGNAL SWING" -rise
66 set_global_assignment -name OUTPUT_IO_TIMING_FAR-END_VMEAS "HALF SIGNAL SWING" -fall
67 set_location_assignment PIN_D25 -to clk_osc1
68 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to clk_osc1
69 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_rzq -tag
    __hps_sdram_p0
70 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[0] -tag
    __hps_sdram_p0
71 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[1] -tag
    __hps_sdram_p0
72 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[2] -tag
    __hps_sdram_p0
73 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[3] -tag
    __hps_sdram_p0
74 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[4] -tag
    __hps_sdram_p0
75 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[5] -tag
    __hps_sdram_p0
76 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[6] -tag
    __hps_sdram_p0
77 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[7] -tag
    __hps_sdram_p0
78 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[8] -tag
    __hps_sdram_p0

```

```

79 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[8] -tag
    --hps_sdram_p0
80 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[9] -tag
    --hps_sdram_p0
81 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[10] -tag
    --hps_sdram_p0
82 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[11] -tag
    --hps_sdram_p0
83 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[12] -tag
    --hps_sdram_p0
84 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[13] -tag
    --hps_sdram_p0
85 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[14] -tag
    --hps_sdram_p0
86 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[15] -tag
    --hps_sdram_p0
87 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[16] -tag
    --hps_sdram_p0
88 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[17] -tag
    --hps_sdram_p0
89 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[18] -tag
    --hps_sdram_p0
90 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[19] -tag
    --hps_sdram_p0
91 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[20] -tag
    --hps_sdram_p0
92 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[21] -tag
    --hps_sdram_p0
93 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[22] -tag
    --hps_sdram_p0
94 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[23] -tag
    --hps_sdram_p0
95 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[24] -tag
    --hps_sdram_p0
96 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[25] -tag
    --hps_sdram_p0
97 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[26] -tag
    --hps_sdram_p0
98 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[27] -tag
    --hps_sdram_p0
99 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[28] -tag
    --hps_sdram_p0
100 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[29] -tag
    --hps_sdram_p0
101 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[30] -tag
    --hps_sdram_p0
102 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[31] -tag
    --hps_sdram_p0
103 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[32] -tag
    --hps_sdram_p0
104 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[33] -tag
    --hps_sdram_p0
105 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[34] -tag
    --hps_sdram_p0
106 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[35] -tag
    --hps_sdram_p0
107 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[36] -tag
    --hps_sdram_p0
108 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[37] -tag
    --hps_sdram_p0
109 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[38] -tag
    --hps_sdram_p0
110 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[39] -tag
    --hps_sdram_p0
111 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
    ddr3_hps_dqs_p[0] -tag --hps_sdram_p0
112 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
    ddr3_hps_dqs_p[1] -tag --hps_sdram_p0

```

```

113 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
114   ddr3_hps_dqs_p[2] -tag __hps_sdram_p0
115 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
116   ddr3_hps_dqs_p[3] -tag __hps_sdram_p0
117 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
118   ddr3_hps_dqs_p[4] -tag __hps_sdram_p0
119 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
120   ddr3_hps_dqs_n[0] -tag __hps_sdram_p0
121 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
122   ddr3_hps_dqs_n[1] -tag __hps_sdram_p0
123 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
124   ddr3_hps_dqs_n[2] -tag __hps_sdram_p0
125 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
126   ddr3_hps_dqs_n[3] -tag __hps_sdram_p0
127 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
128   ddr3_hps_dqs_n[4] -tag __hps_sdram_p0
129 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
130   ddr3_hps_clk_p -tag __hps_sdram_p0
131 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
132   ddr3_hps_clk_n -tag __hps_sdram_p0
133 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[0] -tag
134   __hps_sdram_p0
135 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[10] -tag
136   __hps_sdram_p0
137 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[11] -tag
138   __hps_sdram_p0
139 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[12] -tag
140   __hps_sdram_p0
141 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[13] -tag
142   __hps_sdram_p0
143 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[14] -tag
144   __hps_sdram_p0
145 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[1] -tag
146   __hps_sdram_p0
147 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[2] -tag
148   __hps_sdram_p0
149 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[3] -tag
150   __hps_sdram_p0
151 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[4] -tag
152   __hps_sdram_p0
153 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[5] -tag
154   __hps_sdram_p0
155 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[6] -tag
156   __hps_sdram_p0
157 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[7] -tag
158   __hps_sdram_p0
159 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[8] -tag
160   __hps_sdram_p0
161 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[9] -tag
162   __hps_sdram_p0
163 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_ba[0] -tag
164   __hps_sdram_p0
165 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_ba[1] -tag
166   __hps_sdram_p0
167 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_ba[2] -tag
168   __hps_sdram_p0
169 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_casn -tag
170   __hps_sdram_p0
171 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_cke -tag
172   __hps_sdram_p0
173 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_csn -tag
174   __hps_sdram_p0
175 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_odt -tag
176   __hps_sdram_p0
177 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_rasn -tag
178   __hps_sdram_p0
179 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_wen -tag
180   __hps_sdram_p0

```

```

147 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_resetn -tag
148   --hps_sdram_p0
149 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[0] -tag
150   --hps_sdram_p0
151 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[1] -tag
152   --hps_sdram_p0
153 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[2] -tag
154   --hps_sdram_p0
155 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[3] -tag
156   --hps_sdram_p0
157 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[4] -tag
158   --hps_sdram_p0
159 set_location_assignment PIN_F26 -to ddr3_hps_a[0]
160 set_location_assignment PIN_G30 -to ddr3_hps_a[1]
161 set_location_assignment PIN_F28 -to ddr3_hps_a[2]
162 set_location_assignment PIN_F30 -to ddr3_hps_a[3]
163 set_location_assignment PIN_J25 -to ddr3_hps_a[4]
164 set_location_assignment PIN_J27 -to ddr3_hps_a[5]
165 set_location_assignment PIN_F29 -to ddr3_hps_a[6]
166 set_location_assignment PIN_E28 -to ddr3_hps_a[7]
167 set_location_assignment PIN_H27 -to ddr3_hps_a[8]
168 set_location_assignment PIN_G26 -to ddr3_hps_a[9]
169 set_location_assignment PIN_D29 -to ddr3_hps_a[10]
170 set_location_assignment PIN_C30 -to ddr3_hps_a[11]
171 set_location_assignment PIN_B30 -to ddr3_hps_a[12]
172 set_location_assignment PIN_C29 -to ddr3_hps_a[13]
173 set_location_assignment PIN_H25 -to ddr3_hps_a[14]
174 set_location_assignment PIN_E29 -to ddr3_hps_ba[0]
175 set_location_assignment PIN_J24 -to ddr3_hps_ba[1]
176 set_location_assignment PIN_J23 -to ddr3_hps_ba[2]
177 set_location_assignment PIN_K28 -to ddr3_hps_dm[0]
178 set_location_assignment PIN_M28 -to ddr3_hps_dm[1]
179 set_location_assignment PIN_R28 -to ddr3_hps_dm[2]
180 set_location_assignment PIN_W30 -to ddr3_hps_dm[3]
181 set_location_assignment PIN_W27 -to ddr3_hps_dm[4]
182 set_location_assignment PIN_K23 -to ddr3_hps_dq[0]
183 set_location_assignment PIN_K22 -to ddr3_hps_dq[1]
184 set_location_assignment PIN_H30 -to ddr3_hps_dq[2]
185 set_location_assignment PIN_G28 -to ddr3_hps_dq[3]
186 set_location_assignment PIN_L25 -to ddr3_hps_dq[4]
187 set_location_assignment PIN_L24 -to ddr3_hps_dq[5]
188 set_location_assignment PIN_J30 -to ddr3_hps_dq[6]
189 set_location_assignment PIN_J29 -to ddr3_hps_dq[7]
190 set_location_assignment PIN_K26 -to ddr3_hps_dq[8]
191 set_location_assignment PIN_L26 -to ddr3_hps_dq[9]
192 set_location_assignment PIN_K29 -to ddr3_hps_dq[10]
193 set_location_assignment PIN_K27 -to ddr3_hps_dq[11]
194 set_location_assignment PIN_M26 -to ddr3_hps_dq[12]
195 set_location_assignment PIN_M27 -to ddr3_hps_dq[13]
196 set_location_assignment PIN_L28 -to ddr3_hps_dq[14]
197 set_location_assignment PIN_M30 -to ddr3_hps_dq[15]
198 set_location_assignment PIN_U26 -to ddr3_hps_dq[16]
199 set_location_assignment PIN_T26 -to ddr3_hps_dq[17]
200 set_location_assignment PIN_N29 -to ddr3_hps_dq[18]
201 set_location_assignment PIN_N28 -to ddr3_hps_dq[19]
202 set_location_assignment PIN_P26 -to ddr3_hps_dq[20]
203 set_location_assignment PIN_P27 -to ddr3_hps_dq[21]
204 set_location_assignment PIN_N27 -to ddr3_hps_dq[22]
205 set_location_assignment PIN_R29 -to ddr3_hps_dq[23]
206 set_location_assignment PIN_P24 -to ddr3_hps_dq[24]
207 set_location_assignment PIN_P25 -to ddr3_hps_dq[25]
208 set_location_assignment PIN_T29 -to ddr3_hps_dq[26]
209 set_location_assignment PIN_T28 -to ddr3_hps_dq[27]
210 set_location_assignment PIN_R27 -to ddr3_hps_dq[28]
211 set_location_assignment PIN_R26 -to ddr3_hps_dq[29]
212 set_location_assignment PIN_V30 -to ddr3_hps_dq[30]
213 set_location_assignment PIN_W29 -to ddr3_hps_dq[31]
214 set_location_assignment PIN_W26 -to ddr3_hps_dq[32]

```

```

209 | set_location_assignment PIN_R24 -to ddr3_hps_dq[33]
210 | set_location_assignment PIN_U27 -to ddr3_hps_dq[34]
211 | set_location_assignment PIN_V28 -to ddr3_hps_dq[35]
212 | set_location_assignment PIN_T25 -to ddr3_hps_dq[36]
213 | set_location_assignment PIN_U25 -to ddr3_hps_dq[37]
214 | set_location_assignment PIN_V27 -to ddr3_hps_dq[38]
215 | set_location_assignment PIN_Y29 -to ddr3_hps_dq[39]
216 | set_location_assignment PIN_M19 -to ddr3_hps_dqs_n[0]
217 | set_location_assignment PIN_N24 -to ddr3_hps_dqs_n[1]
218 | set_location_assignment PIN_R18 -to ddr3_hps_dqs_n[2]
219 | set_location_assignment PIN_R21 -to ddr3_hps_dqs_n[3]
220 | set_location_assignment PIN_T23 -to ddr3_hps_dqs_n[4]
221 | set_location_assignment PIN_N18 -to ddr3_hps_dqs_p[0]
222 | set_location_assignment PIN_N25 -to ddr3_hps_dqs_p[1]
223 | set_location_assignment PIN_R19 -to ddr3_hps_dqs_p[2]
224 | set_location_assignment PIN_R22 -to ddr3_hps_dqs_p[3]
225 | set_location_assignment PIN_T24 -to ddr3_hps_dqs_p[4]
226 | set_location_assignment PIN_E27 -to ddr3_hps_casn
227 | set_location_assignment PIN_L29 -to ddr3_hps_cke
228 | set_location_assignment PIN_L23 -to ddr3_hps_clk_n
229 | set_location_assignment PIN_M23 -to ddr3_hps_clk_p
230 | set_location_assignment PIN_H24 -to ddr3_hps_csn
231 | set_location_assignment PIN_H28 -to ddr3_hps_odt
232 | set_location_assignment PIN_D30 -to ddr3_hps_rasn
233 | set_location_assignment PIN_P30 -to ddr3_hps_resetn
234 | set_location_assignment PIN_C28 -to ddr3_hps_wen
235 | set_location_assignment PIN_D27 -to ddr3_hps_rzq
236 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to clk_osc2
237 | set_location_assignment PIN_F25 -to clk_osc2
238 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to mictor_rstn
239 | set_location_assignment PIN_C27 -to mictor_rstn
240 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to hps_resetn
241 | set_location_assignment PIN_F23 -to hps_resetn
242 | set_instance_assignment -name IO_STANDARD "1.5 V" -to user_dipsw_hps
243 | set_location_assignment PIN_N30 -to user_dipsw_hps[0]
244 | set_location_assignment PIN_P29 -to user_dipsw_hps[1]
245 | set_location_assignment PIN_P22 -to user_dipsw_hps[2]
246 | set_location_assignment PIN_V20 -to user_dipsw_hps[3]
247 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to user_led_hps
248 | set_location_assignment PIN_E17 -to user_led_hps[0]
249 | set_location_assignment PIN_E18 -to user_led_hps[1]
250 | set_location_assignment PIN_G17 -to user_led_hps[2]
251 | set_location_assignment PIN_C18 -to user_led_hps[3]
252 | set_instance_assignment -name IO_STANDARD "1.5 V" -to user_pb_hps
253 | set_location_assignment PIN_T30 -to user_pb_hps[0]
254 | set_location_assignment PIN_U28 -to user_pb_hps[1]
255 | set_location_assignment PIN_T21 -to user_pb_hps[2]
256 | set_location_assignment PIN_U20 -to user_pb_hps[3]
257 | set_location_assignment PIN_E24 -to uart_rx
258 | set_location_assignment PIN_D24 -to uart_tx
259 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to uart_rx
260 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to uart_tx
261 | set_location_assignment PIN_D22 -to i2c_scl_hps
262 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to i2c_scl_hps
263 | set_location_assignment PIN_C23 -to i2c_sda_hps
264 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to i2c_sda_hps
265 | set_location_assignment PIN_H20 -to spi_csn
266 | set_location_assignment PIN_B23 -to spi_miso
267 | set_location_assignment PIN_C22 -to spi_mosi
268 | set_location_assignment PIN_A23 -to spi_sck
269 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to spi_csn
270 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to spi_miso
271 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to spi_mosi
272 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to spi_sck
273 | set_location_assignment PIN_D19 -to qspi_clk
274 | set_location_assignment PIN_C20 -to qspi_io[0]
275 | set_location_assignment PIN_H18 -to qspi_io[1]
276 | set_location_assignment PIN_A19 -to qspi_io[2]

```

```

277 | set_location_assignment PIN_E19 -to qspi_io[3]
278 | set_location_assignment PIN_A18 -to qspi_ss0
279 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to qspi_clk
280 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to qspi_io
281 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to qspi_ss0
282 | set_location_assignment PIN_B16 -to sd_dat[3]
283 | set_location_assignment PIN_A16 -to sd_clk
284 | set_location_assignment PIN_F18 -to sd_cmd
285 | set_location_assignment PIN_G18 -to sd_dat[0]
286 | set_location_assignment PIN_C17 -to sd_dat[1]
287 | set_location_assignment PIN_D17 -to sd_dat[2]
288 | set_location_assignment PIN_B17 -to sd_pwren
289 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_clk
290 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_cmd
291 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_pwren
292 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_dat
293 | set_location_assignment PIN_N16 -to usb_clk
294 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_clk
295 | set_location_assignment PIN_E16 -to usb_data[0]
296 | set_location_assignment PIN_G16 -to usb_data[1]
297 | set_location_assignment PIN_D16 -to usb_data[2]
298 | set_location_assignment PIN_D14 -to usb_data[3]
299 | set_location_assignment PIN_A15 -to usb_data[4]
300 | set_location_assignment PIN_C14 -to usb_data[5]
301 | set_location_assignment PIN_D15 -to usb_data[6]
302 | set_location_assignment PIN_M17 -to usb_data[7]
303 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_data
304 | set_location_assignment PIN_A14 -to usb_nxt
305 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_nxt
306 | set_location_assignment PIN_E14 -to usb_dir
307 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_stp
308 | set_location_assignment PIN_C15 -to usb_stp
309 | set_location_assignment PIN_H19 -to enet_hps_gtx_clk
310 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_gtx_clk
311 | set_location_assignment PIN_B21 -to enet_hps_mdc
312 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_mdc
313 | set_location_assignment PIN_E21 -to enet_hps_mdio
314 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_mdio
315 | set_location_assignment PIN_G20 -to enet_hps_rx_clk
316 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_rx_clk
317 | set_location_assignment PIN_K17 -to enet_hps_rx_dv
318 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_rx_dv
319 | set_location_assignment PIN_A21 -to enet_hps_rxd[0]
320 | set_location_assignment PIN_B20 -to enet_hps_rxd[1]
321 | set_location_assignment PIN_B18 -to enet_hps_rxd[2]
322 | set_location_assignment PIN_D21 -to enet_hps_rxd[3]
323 | set_location_assignment PIN_A20 -to enet_hps_tx_en
324 | set_location_assignment PIN_F20 -to enet_hps_txd[0]
325 | set_location_assignment PIN_J19 -to enet_hps_txd[1]
326 | set_location_assignment PIN_F21 -to enet_hps_txd[2]
327 | set_location_assignment PIN_F19 -to enet_hps_txd[3]
328 | set_instance_assignment -name IO_STANDARD "2.5 V" -to enet_hps_rxd
329 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_tx_en
330 | set_instance_assignment -name IO_STANDARD "2.5 V" -to enet_hps_txd
331 | set_location_assignment PIN_C19 -to enet_hps_intn
332 | set_location_assignment PIN_B22 -to can_0_rx
333 | set_location_assignment PIN_G22 -to can_0_tx
334 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to can_0_rx
335 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to can_0_tx
336 | set_location_assignment PIN_B26 -to trace_clk_mic
337 | set_location_assignment PIN_B25 -to trace_data[0]
338 | set_location_assignment PIN_C25 -to trace_data[1]
339 | set_location_assignment PIN_A25 -to trace_data[2]
340 | set_location_assignment PIN_H23 -to trace_data[3]
341 | set_location_assignment PIN_A24 -to trace_data[4]
342 | set_location_assignment PIN_G21 -to trace_data[5]
343 | set_location_assignment PIN_C24 -to trace_data[6]
344 | set_location_assignment PIN_E23 -to trace_data[7]

```

```

345 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to trace_clk_mic
346 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to trace_data
347 | set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_dir
348 | set_location_assignment PIN_AB27 -to clk_100m_fpga
349 | set_instance_assignment -name IO_STANDARD "2.5 V" -to clk_100m_fpga
350 | set_location_assignment PIN_AC18 -to clk_50m_fpga
351 | set_instance_assignment -name IO_STANDARD "1.5 V" -to clk_50m_fpga
352 | set_location_assignment PIN_AF14 -to clk_bot1
353 | set_instance_assignment -name IO_STANDARD "1.5 V" -to clk_bot1
354 | set_location_assignment PIN_Y26 -to clk_enet_fpga_p
355 | set_instance_assignment -name IO_STANDARD LVDS -to clk_enet_fpga_p
356 | set_location_assignment PIN_AA26 -to clk_top1
357 | set_instance_assignment -name IO_STANDARD "2.5 V" -to clk_top1
358 | set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a
359 | set_location_assignment PIN_AJ14 -to ddr3_fpga_a[0]
360 | set_location_assignment PIN_AK14 -to ddr3_fpga_a[1]
361 | set_location_assignment PIN_AH12 -to ddr3_fpga_a[2]
362 | set_location_assignment PIN_AJ12 -to ddr3_fpga_a[3]
363 | set_location_assignment PIN_AG15 -to ddr3_fpga_a[4]
364 | set_location_assignment PIN_AH15 -to ddr3_fpga_a[5]
365 | set_location_assignment PIN_AK12 -to ddr3_fpga_a[6]
366 | set_location_assignment PIN_AK13 -to ddr3_fpga_a[7]
367 | set_location_assignment PIN_AH13 -to ddr3_fpga_a[8]
368 | set_location_assignment PIN_AH14 -to ddr3_fpga_a[9]
369 | set_location_assignment PIN_AJ9 -to ddr3_fpga_a[10]
370 | set_location_assignment PIN_AK9 -to ddr3_fpga_a[11]
371 | set_location_assignment PIN_AK7 -to ddr3_fpga_a[12]
372 | set_location_assignment PIN_AK8 -to ddr3_fpga_a[13]
373 | set_location_assignment PIN_AG12 -to ddr3_fpga_a[14]
374 | set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_ba
375 | set_location_assignment PIN_AH10 -to ddr3_fpga_ba[0]
376 | set_location_assignment PIN_AJ11 -to ddr3_fpga_ba[1]
377 | set_location_assignment PIN_AK11 -to ddr3_fpga_ba[2]
378 | set_location_assignment PIN_AH7 -to ddr3_fpga_casn
379 | set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_casn -tag
    --orion_system_mem_if_ddr3_fpga_p0
380 | set_location_assignment PIN_AJ21 -to ddr3_fpga_cke
381 | set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_cke -tag
    --orion_system_mem_if_ddr3_fpga_p0
382 | set_location_assignment PIN_AA15 -to ddr3_fpga_clk_n
383 | set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
    ddr3_fpga_clk_n -tag --orion_system_mem_if_ddr3_fpga_p0
384 | set_location_assignment PIN_AA14 -to ddr3_fpga_clk_p
385 | set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
    ddr3_fpga_clk_p -tag --orion_system_mem_if_ddr3_fpga_p0
386 | set_location_assignment PIN_AB15 -to ddr3_fpga_csn
387 | set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_csn -tag
    --orion_system_mem_if_ddr3_fpga_p0
388 | set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dm
389 | set_location_assignment PIN_AH17 -to ddr3_fpga_dm[0]
390 | set_location_assignment PIN_AG23 -to ddr3_fpga_dm[1]
391 | set_location_assignment PIN_AK23 -to ddr3_fpga_dm[2]
392 | set_location_assignment PIN_AJ27 -to ddr3_fpga_dm[3]
393 | set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq
394 | set_location_assignment PIN_AF18 -to ddr3_fpga_dq[0]
395 | set_location_assignment PIN_AE17 -to ddr3_fpga_dq[1]
396 | set_location_assignment PIN_AG16 -to ddr3_fpga_dq[2]
397 | set_location_assignment PIN_AF16 -to ddr3_fpga_dq[3]
398 | set_location_assignment PIN_AH20 -to ddr3_fpga_dq[4]
399 | set_location_assignment PIN_AG21 -to ddr3_fpga_dq[5]
400 | set_location_assignment PIN_AJ16 -to ddr3_fpga_dq[6]
401 | set_location_assignment PIN_AH18 -to ddr3_fpga_dq[7]
402 | set_location_assignment PIN_AK18 -to ddr3_fpga_dq[8]
403 | set_location_assignment PIN_AJ17 -to ddr3_fpga_dq[9]
404 | set_location_assignment PIN_AG18 -to ddr3_fpga_dq[10]
405 | set_location_assignment PIN_AK19 -to ddr3_fpga_dq[11]
406 | set_location_assignment PIN_AG20 -to ddr3_fpga_dq[12]
407 | set_location_assignment PIN_AF19 -to ddr3_fpga_dq[13]

```

```

408 set_location_assignment PIN_AJ20 -to ddr3_fpga_dq[14]
409 set_location_assignment PIN_AH24 -to ddr3_fpga_dq[15]
410 set_location_assignment PIN_AE19 -to ddr3_fpga_dq[16]
411 set_location_assignment PIN_AE18 -to ddr3_fpga_dq[17]
412 set_location_assignment PIN_AG22 -to ddr3_fpga_dq[18]
413 set_location_assignment PIN_AK22 -to ddr3_fpga_dq[19]
414 set_location_assignment PIN_AF21 -to ddr3_fpga_dq[20]
415 set_location_assignment PIN_AF20 -to ddr3_fpga_dq[21]
416 set_location_assignment PIN_AH23 -to ddr3_fpga_dq[22]
417 set_location_assignment PIN_AK24 -to ddr3_fpga_dq[23]
418 set_location_assignment PIN_AF24 -to ddr3_fpga_dq[24]
419 set_location_assignment PIN_AF23 -to ddr3_fpga_dq[25]
420 set_location_assignment PIN_AJ24 -to ddr3_fpga_dq[26]
421 set_location_assignment PIN_AK26 -to ddr3_fpga_dq[27]
422 set_location_assignment PIN_AE23 -to ddr3_fpga_dq[28]
423 set_location_assignment PIN_AE22 -to ddr3_fpga_dq[29]
424 set_location_assignment PIN_AG25 -to ddr3_fpga_dq[30]
425 set_location_assignment PIN_AK27 -to ddr3_fpga_dq[31]
426 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
   ddr3_fpga_dqs_n
427 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
   ddr3_fpga_dqs_p
428 set_location_assignment PIN_AE16 -to ddr3_fpga_0dt
429 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_0dt -tag
   __orion_system_mem_if_ddr3_fpga_p0
430 set_location_assignment PIN_AH8 -to ddr3_fpga_rasn
431 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_rasn
432 set_location_assignment PIN_AK21 -to ddr3_fpga_resetn
433 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_resetn
434 set_location_assignment PIN_AJ6 -to ddr3_fpga_wen
435 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_wen
436 set_location_assignment PIN_C12 -to hsma_tx_d_p[16]
437 set_location_assignment PIN_B13 -to hsma_tx_d_p[15]
438 set_location_assignment PIN_C10 -to hsma_tx_d_p[14]
439 set_location_assignment PIN_C8 -to hsma_tx_d_p[13]
440 set_location_assignment PIN_A9 -to hsma_tx_d_p[12]
441 set_location_assignment PIN_C7 -to hsma_tx_d_p[11]
442 set_location_assignment PIN_A6 -to hsma_tx_d_p[10]
443 set_location_assignment PIN_D5 -to hsma_tx_d_p[9]
444 set_location_assignment PIN_A4 -to hsma_tx_d_p[8]
445 set_location_assignment PIN_C3 -to hsma_tx_d_p[7]
446 set_location_assignment PIN_B2 -to hsma_tx_d_p[6]
447 set_location_assignment PIN_D2 -to hsma_tx_d_p[5]
448 set_location_assignment PIN_E1 -to hsma_tx_d_p[4]
449 set_location_assignment PIN_E3 -to hsma_tx_d_p[3]
450 set_location_assignment PIN_E4 -to hsma_tx_d_p[2]
451 set_location_assignment PIN_D6 -to hsma_tx_d_p[1]
452 set_location_assignment PIN_E8 -to hsma_tx_d_p[0]
453 set_instance_assignment -name IO_STANDARD LVDS -to hsma_tx_d_p
454 set_location_assignment PIN_B11 -to hsma_tx_d_n[16]
455 set_location_assignment PIN_A13 -to hsma_tx_d_n[15]
456 set_location_assignment PIN_C9 -to hsma_tx_d_n[14]
457 set_location_assignment PIN_B8 -to hsma_tx_d_n[13]
458 set_location_assignment PIN_A8 -to hsma_tx_d_n[12]
459 set_location_assignment PIN_B7 -to hsma_tx_d_n[11]
460 set_location_assignment PIN_A5 -to hsma_tx_d_n[10]
461 set_location_assignment PIN_C4 -to hsma_tx_d_n[9]
462 set_location_assignment PIN_A3 -to hsma_tx_d_n[8]
463 set_location_assignment PIN_B3 -to hsma_tx_d_n[7]
464 set_location_assignment PIN_B1 -to hsma_tx_d_n[6]
465 set_location_assignment PIN_C2 -to hsma_tx_d_n[5]
466 set_location_assignment PIN_D1 -to hsma_tx_d_n[4]
467 set_location_assignment PIN_E2 -to hsma_tx_d_n[3]
468 set_location_assignment PIN_D4 -to hsma_tx_d_n[2]
469 set_location_assignment PIN_C5 -to hsma_tx_d_n[1]
470 set_location_assignment PIN_D7 -to hsma_tx_d_n[0]
471 set_location_assignment PIN_AH2 -to hsma_sda
472 set_location_assignment PIN_AA12 -to hsma_scl

```

```

473 set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_scl
474 set_location_assignment PIN_J2 -to hsma_rx_p[3]
475 set_location_assignment PIN_L2 -to hsma_rx_p[2]
476 set_location_assignment PIN_N2 -to hsma_rx_p[1]
477 set_location_assignment PIN_R2 -to hsma_rx_p[0]
478 set_location_assignment PIN_J1 -to hsma_rx_n[3]
479 set_location_assignment PIN_L1 -to hsma_rx_n[2]
480 set_location_assignment PIN_N1 -to hsma_rx_n[1]
481 set_location_assignment PIN_R1 -to hsma_rx_n[0]
482 set_location_assignment PIN_F15 -to hsma_rx_d_p[16]
483 set_location_assignment PIN_C13 -to hsma_rx_d_p[15]
484 set_location_assignment PIN_F13 -to hsma_rx_d_p[14]
485 set_location_assignment PIN_E12 -to hsma_rx_d_p[13]
486 set_location_assignment PIN_D11 -to hsma_rx_d_p[12]
487 set_location_assignment PIN_E9 -to hsma_rx_d_p[11]
488 set_location_assignment PIN_B6 -to hsma_rx_d_p[10]
489 set_location_assignment PIN_F11 -to hsma_rx_d_p[9]
490 set_location_assignment PIN_F9 -to hsma_rx_d_p[8]
491 set_location_assignment PIN_G10 -to hsma_rx_d_p[7]
492 set_location_assignment PIN_H8 -to hsma_rx_d_p[6]
493 set_location_assignment PIN_J7 -to hsma_rx_d_p[5]
494 set_location_assignment PIN_G12 -to hsma_rx_d_p[4]
495 set_location_assignment PIN_K7 -to hsma_rx_d_p[3]
496 set_location_assignment PIN_J10 -to hsma_rx_d_p[2]
497 set_location_assignment PIN_K12 -to hsma_rx_d_p[1]
498 set_location_assignment PIN_H14 -to hsma_rx_d_p[0]
499 set_instance_assignment -name IO_STANDARD LVDS -to hsma_rx_d_p
500 set_location_assignment PIN_F14 -to hsma_rx_d_n[16]
501 set_location_assignment PIN_B12 -to hsma_rx_d_n[15]
502 set_location_assignment PIN_E13 -to hsma_rx_d_n[14]
503 set_location_assignment PIN_D12 -to hsma_rx_d_n[13]
504 set_location_assignment PIN_D10 -to hsma_rx_d_n[12]
505 set_location_assignment PIN_D9 -to hsma_rx_d_n[11]
506 set_location_assignment PIN_B5 -to hsma_rx_d_n[10]
507 set_location_assignment PIN_E11 -to hsma_rx_d_n[9]
508 set_location_assignment PIN_F8 -to hsma_rx_d_n[8]
509 set_location_assignment PIN_F10 -to hsma_rx_d_n[7]
510 set_location_assignment PIN_G8 -to hsma_rx_d_n[6]
511 set_location_assignment PIN_H7 -to hsma_rx_d_n[5]
512 set_location_assignment PIN_G11 -to hsma_rx_d_n[4]
513 set_location_assignment PIN_K8 -to hsma_rx_d_n[3]
514 set_location_assignment PIN_J9 -to hsma_rx_d_n[2]
515 set_location_assignment PIN_J12 -to hsma_rx_d_n[1]
516 set_location_assignment PIN_G13 -to hsma_rx_d_n[0]
517 set_location_assignment PIN_AD12 -to hsma_prsntn
518 set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_prsntn
519 set_location_assignment PIN_AG1 -to hsma_d[3]
520 set_location_assignment PIN_AG7 -to hsma_d[2]
521 set_location_assignment PIN_AF8 -to hsma_d[1]
522 set_location_assignment PIN_AF9 -to hsma_d[0]
523 set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_d
524 set_location_assignment PIN_A10 -to hsma_clk_out0
525 set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_clk_out0
526 set_location_assignment PIN_E7 -to hsma_clk_out_p2
527 set_instance_assignment -name IO_STANDARD LVDS -to hsma_clk_out_p2
528 set_location_assignment PIN_E6 -to hsma_clk_out_n2
529 set_location_assignment PIN_K14 -to hsma_clk_in0
530 set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_clk_in0
531 set_location_assignment PIN_H15 -to hsma_clk_in_p2
532 set_instance_assignment -name IO_STANDARD LVDS -to hsma_clk_in_p2
533 set_location_assignment PIN_G15 -to hsma_clk_in_n2
534 set_location_assignment PIN_AB22 -to enet1_tx_en
535 set_location_assignment PIN_AB26 -to enet1_tx_d[3]
536 set_location_assignment PIN_AA25 -to enet1_tx_d[2]
537 set_location_assignment PIN_Y21 -to enet1_tx_d[1]
538 set_location_assignment PIN_W20 -to enet1_tx_d[0]
539 set_location_assignment PIN_W25 -to enet1_tx_clk_fb
540 set_location_assignment PIN_AE28 -to enet1_rx_error

```

```

541 set_location_assignment PIN_Y23 -to enet1_rx_dv
542 set_location_assignment PIN_AE27 -to enet1_rx_d[3]
543 set_location_assignment PIN_AB25 -to enet1_rx_d[2]
544 set_location_assignment PIN_AA24 -to enet1_rx_d[1]
545 set_location_assignment PIN_AB23 -to enet1_rx_d[0]
546 set_location_assignment PIN_Y24 -to enet1_rx_clk
547 set_location_assignment PIN_W24 -to enet2_tx_en
548 set_location_assignment PIN_V23 -to enet2_tx_d[3]
549 set_location_assignment PIN_AF28 -to enet2_tx_d[2]
550 set_location_assignment PIN_AG28 -to enet2_tx_d[1]
551 set_location_assignment PIN_AG27 -to enet2_tx_d[0]
552 set_location_assignment PIN_AG30 -to enet2_tx_clk_fb
553 set_location_assignment PIN_V25 -to enet2_rx_error
554 set_location_assignment PIN_AC28 -to enet2_rx_dv
555 set_location_assignment PIN_AC27 -to enet2_rx_d[3]
556 set_location_assignment PIN_AD26 -to enet2_rx_d[2]
557 set_location_assignment PIN_AF30 -to enet2_rx_d[1]
558 set_location_assignment PIN_AF29 -to enet2_rx_d[0]
559 set_location_assignment PIN_AH30 -to enet2_rx_clk
560 set_location_assignment PIN_AC29 -to sdi_tx_sd_hdn
561 set_location_assignment PIN_AA30 -to sdi_tx_en
562 set_location_assignment PIN_AA28 -to sdi_rx_en
563 set_location_assignment PIN_AB28 -to sdi_rx_bypass
564 set_location_assignment PIN_AB30 -to sdi_rsti
565 set_location_assignment PIN_AD30 -to sdi_fault
566 set_location_assignment PIN_AB12 -to sdi_clk148_up
567 set_location_assignment PIN_AF6 -to sdi_clk148_dn
568 set_location_assignment PIN_AG10 -to user_dipsw_fpga[0]
569 set_location_assignment PIN_AH9 -to user_dipsw_fpga[1]
570 set_location_assignment PIN_AF11 -to user_dipsw_fpga[2]
571 set_location_assignment PIN_AG11 -to user_dipsw_fpga[3]
572 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_dipsw_fpga[0]
573 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_dipsw_fpga[1]
574 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_dipsw_fpga[2]
575 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_dipsw_fpga[3]
576 set_location_assignment PIN_AK2 -to user_led_fpga[0]
577 set_location_assignment PIN_Y16 -to user_led_fpga[1]
578 set_location_assignment PIN_W15 -to user_led_fpga[2]
579 set_location_assignment PIN_AB17 -to user_led_fpga[3]
580 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_led_fpga[0]
581 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_led_fpga[1]
582 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_led_fpga[2]
583 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_led_fpga[3]
584 set_location_assignment PIN_AA13 -to user_pb_fpga[0]
585 set_location_assignment PIN_AB13 -to user_pb_fpga[1]
586 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_pb_fpga[0]
587 set_instance_assignment -name IO_STANDARD "1.5 V" -to user_pb_fpga[1]
588 set_location_assignment PIN_AG17 -to ddr3_fpga_rzq
589 set_instance_assignment -name IO_STANDARD "SSTL-15" -to ddr3_fpga_rzq -tag
    _orion_system_mem_if_ddr3_fpga_p0
590 #---Found multiple nets with the same name
591 set_location_assignment PIN_W21 -to pcie_perstn_in
592 set_location_assignment PIN_AG6 -to pcie_perstn_out
593 set_location_assignment PIN_W7 -to pcie_refclk_n
594 set_location_assignment PIN_W8 -to pcie_refclk_p
595 set_location_assignment PIN_AE1 -to pcie_rx_n[0]
596 set_location_assignment PIN_AC1 -to pcie_rx_n[1]
597 set_location_assignment PIN_AA1 -to pcie_rx_n[2]
598 set_location_assignment PIN_W1 -to pcie_rx_n[3]
599 set_location_assignment PIN_AE2 -to pcie_rx_p[0]
600 set_location_assignment PIN_AC2 -to pcie_rx_p[1]
601 set_location_assignment PIN_AA2 -to pcie_rx_p[2]
602 set_location_assignment PIN_W2 -to pcie_rx_p[3]
603 set_location_assignment PIN_AD3 -to pcie_tx_n[0]
604 set_location_assignment PIN_AB3 -to pcie_tx_n[1]
605 set_location_assignment PIN_Y3 -to pcie_tx_n[2]
606 set_location_assignment PIN_V3 -to pcie_tx_n[3]
607 set_location_assignment PIN_AD4 -to pcie_tx_p[0]

```

```

608 set_location_assignment PIN_AB4 -to pcie_tx_p[1]
609 set_location_assignment PIN_Y4 -to pcie_tx_p[2]
610 set_location_assignment PIN_V4 -to pcie_tx_p[3]
611 set_location_assignment PIN_AF13 -to usb_b2_clk
612 set_location_assignment PIN_AK28 -to usb_b2_data[0]
613 set_location_assignment PIN_AD20 -to usb_b2_data[1]
614 set_location_assignment PIN_AD21 -to usb_b2_data[2]
615 set_location_assignment PIN_Y19 -to usb_b2_data[3]
616 set_location_assignment PIN_AA20 -to usb_b2_data[4]
617 set_location_assignment PIN_AH27 -to usb_b2_data[5]
618 set_location_assignment PIN_AF25 -to usb_b2_data[6]
619 set_location_assignment PIN_AC22 -to usb_b2_data[7]
620 set_instance_assignment -name IO_STANDARD "1.5 V" -to usb_b2_clk
621 set_instance_assignment -name IO_STANDARD "1.5 V" -to usb_b2_data
622 set_location_assignment PIN_AD27 -to cpu_resetn
623 set_location_assignment PIN_AG5 -to enet1_tx_error
624 set_location_assignment PIN_AH5 -to enet2_tx_error
625 set_location_assignment PIN_AJ1 -to enet_dual_resetn
626 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to pcie_tx_p
627 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to pcie_rx_p
628 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_p
629 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_rx_p
630 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to gxb_rx_14_p
631 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to gxb_tx_14_p
632 set_instance_assignment -name IO_STANDARD LVDS -to clk_148_p
633 set_instance_assignment -name IO_STANDARD LVDS -to pcie_refclk_p
634 set_location_assignment PIN_T9 -to clk_148_p
635 set_global_assignment -name ENABLE_SIGNALTAP OFF
636 set_global_assignment -name USE_SIGNALTAP_FILE output_files_sof_rpt/stp1.stp
637 set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
638 set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
639 set_location_assignment PIN_G7 -to i2c_scl_fpga
640 set_location_assignment PIN_F6 -to i2c_sda_fpga
641 set_location_assignment PIN_AJ4 -to usb_empty
642 set_location_assignment PIN_AK3 -to usb_full
643 set_location_assignment PIN_AE14 -to usb_oen
644 set_location_assignment PIN_AJ5 -to usb_rdn
645 set_location_assignment PIN_AD14 -to usb_resetn
646 set_location_assignment PIN_AK4 -to usb_scl
647 set_location_assignment PIN_AE13 -to usb_sda
648 set_location_assignment PIN_AK6 -to usb_wrn
649 set_location_assignment PIN_H12 -to enet_fpga_mdc
650 set_location_assignment PIN_H13 -to enet_fpga_mdio
651 set_location_assignment PIN_T8 -to clk_148_n
652 set_location_assignment PIN_Y27 -to clk_enet_fpga_n
653 set_location_assignment PIN_U1 -to gxb_rx_14_n
654 set_location_assignment PIN_U2 -to gxb_rx_14_p
655 set_location_assignment PIN_T3 -to gxb_tx_14_n
656 set_location_assignment PIN_T4 -to gxb_tx_14_p
657 set_location_assignment PIN_P8 -to refclk_q12_n
658 set_location_assignment PIN_P9 -to refclk_q12_p
659 set_location_assignment PIN_AE29 -to pcie_smbclk
660 set_location_assignment PIN_J14 -to pcie_smbdat
661 set_location_assignment PIN_W22 -to pcie_waken
662 set_location_assignment PIN_AD25 -to max_fpga_miso
663 set_location_assignment PIN_AE26 -to max_fpga_mosi
664 set_location_assignment PIN_AJ29 -to max_fpga_sck
665 set_location_assignment PIN_AC25 -to max_fpga_ssel
666 set_location_assignment PIN_AD29 -to pcie_prsnt2_x1
667 set_location_assignment PIN_A11 -to pcie_prsnt2_x4
668 set_instance_assignment -name IO_STANDARD LVDS -to refclk_q12_p
669 set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
670 set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT_AND_ROUTING
   -section_id Top
671 set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
672 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
   ddr3_hps_dq[0] -tag __hps_sdram_p0

```



```

741 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
742     ddr3_hps_dq[34] -tag __hps_sdram_p0
743 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
744     ddr3_hps_dq[35] -tag __hps_sdram_p0
745 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
746     ddr3_hps_dq[35] -tag __hps_sdram_p0
747 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
748     ddr3_hps_dq[36] -tag __hps_sdram_p0
749 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
750     ddr3_hps_dq[36] -tag __hps_sdram_p0
751 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
752     ddr3_hps_dq[37] -tag __hps_sdram_p0
753 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
754     ddr3_hps_dq[37] -tag __hps_sdram_p0
755 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
756     ddr3_hps_dq[38] -tag __hps_sdram_p0
757 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
758     ddr3_hps_dq[38] -tag __hps_sdram_p0
759 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
760     ddr3_hps_dq[39] -tag __hps_sdram_p0
761 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
762     ddr3_hps_dq[39] -tag __hps_sdram_p0
763 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
764     ddr3_hps_dqs_p[0] -tag __hps_sdram_p0
765 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
766     ddr3_hps_dqs_p[0] -tag __hps_sdram_p0
767 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
768     ddr3_hps_dqs_p[1] -tag __hps_sdram_p0
769 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
770     ddr3_hps_dqs_p[1] -tag __hps_sdram_p0
771 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
772     ddr3_hps_dqs_p[2] -tag __hps_sdram_p0
773 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
774     ddr3_hps_dqs_p[2] -tag __hps_sdram_p0
775 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
776     ddr3_hps_dqs_p[3] -tag __hps_sdram_p0
777 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
778     ddr3_hps_dqs_p[3] -tag __hps_sdram_p0
779 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
780     ddr3_hps_dqs_p[4] -tag __hps_sdram_p0
781 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
782     ddr3_hps_dqs_p[4] -tag __hps_sdram_p0
783 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
784     ddr3_hps_dqs_n[0] -tag __hps_sdram_p0
785 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
786     ddr3_hps_dqs_n[0] -tag __hps_sdram_p0
787 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
788     ddr3_hps_dqs_n[1] -tag __hps_sdram_p0
789 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
790     ddr3_hps_dqs_n[1] -tag __hps_sdram_p0
791 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
792     ddr3_hps_dqs_n[2] -tag __hps_sdram_p0
793 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
794     ddr3_hps_dqs_n[2] -tag __hps_sdram_p0
795 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
796     ddr3_hps_dqs_n[3] -tag __hps_sdram_p0
797 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
798     ddr3_hps_dqs_n[3] -tag __hps_sdram_p0
799 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
800     ddr3_hps_dqs_n[4] -tag __hps_sdram_p0
801 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
802     ddr3_hps_dqs_n[4] -tag __hps_sdram_p0
803 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITHOUT CALIBRATION" -to
804     ddr3_hps_clk_p -tag __hps_sdram_p0
805 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITHOUT CALIBRATION" -to
806     ddr3_hps_clk_n -tag __hps_sdram_p0
807 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[0]
808     -tag __hps_sdram_p0

```

```

775 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[10]
    -tag __hps_sdram_p0
776 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[11]
    -tag __hps_sdram_p0
777 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[12]
    -tag __hps_sdram_p0
778 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[13]
    -tag __hps_sdram_p0
779 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[14]
    -tag __hps_sdram_p0
780 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[1]
    -tag __hps_sdram_p0
781 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[2]
    -tag __hps_sdram_p0
782 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[3]
    -tag __hps_sdram_p0
783 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[4]
    -tag __hps_sdram_p0
784 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[5]
    -tag __hps_sdram_p0
785 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[6]
    -tag __hps_sdram_p0
786 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[7]
    -tag __hps_sdram_p0
787 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[8]
    -tag __hps_sdram_p0
788 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[9]
    -tag __hps_sdram_p0
789 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_ba[0]
    -tag __hps_sdram_p0
790 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_ba[1]
    -tag __hps_sdram_p0
791 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_ba[2]
    -tag __hps_sdram_p0
792 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_casn
    -tag __hps_sdram_p0
793 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_cke -tag
    __hps_sdram_p0
794 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_csn -tag
    __hps_sdram_p0
795 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_odt -tag
    __hps_sdram_p0
796 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_rasn
    -tag __hps_sdram_p0
797 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_wen -tag
    __hps_sdram_p0
798 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_resetn
    -tag __hps_sdram_p0
799 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_hps_dm[0] -tag __hps_sdram_p0
800 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_hps_dm[1] -tag __hps_sdram_p0
801 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_hps_dm[2] -tag __hps_sdram_p0
802 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_hps_dm[3] -tag __hps_sdram_p0
803 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_hps_dm[4] -tag __hps_sdram_p0
804 set_instance_assignment -name ENABLE_BENEFICIAL_SKew_OPTIMIZATION_FOR_NON_GLOBAL_CLOCKS_ON
    -to u0|hps_0|hps_io|border|hps_sdram_inst -tag __hps_sdram_p0
805 set_instance_assignment -name INPUT_TERMINATION DIFFERENTIAL -to hsma_rx_d_p
806 set_instance_assignment -name INPUT_TERMINATION DIFFERENTIAL -to hsma_clk_in_p2
807 set_instance_assignment -name INPUT_TERMINATION "OCT 100 OHMS" -to clk_148_p
808 set_instance_assignment -name INPUT_TERMINATION "OCT 100 OHMS" -to pcie_refclk_p
809 set_instance_assignment -name INPUT_TERMINATION "OCT 100 OHMS" -to refclk_q12_p
810 set_location_assignment PIN_AA16 -to clk_25m_fpga
811 set_instance_assignment -name IO_STANDARD "1.5 V" -to clk_25m_fpga
812 set_location_assignment PIN_AJ2 -to hsma_clk_out_p1

```

```

813 set_location_assignment PIN_AC12 -to hsma_clk_out_n1
814 set_location_assignment PIN_AG2 -to hsma_clk_in_p1
815 set_location_assignment PIN_AH3 -to hsma_clk_in_n1
816
817 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[0] -tag
818   __orion_system_mem_if_ddr3_fpga_p0
819 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[1] -tag
820   __orion_system_mem_if_ddr3_fpga_p0
821 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[2] -tag
822   __orion_system_mem_if_ddr3_fpga_p0
823 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[3] -tag
824   __orion_system_mem_if_ddr3_fpga_p0
825 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[5] -tag
826   __orion_system_mem_if_ddr3_fpga_p0
827 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[4] -tag
828   __orion_system_mem_if_ddr3_fpga_p0
829 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[6] -tag
830   __orion_system_mem_if_ddr3_fpga_p0
831 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[7] -tag
832   __orion_system_mem_if_ddr3_fpga_p0
833 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[8] -tag
834   __orion_system_mem_if_ddr3_fpga_p0
835 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[9] -tag
836   __orion_system_mem_if_ddr3_fpga_p0
837 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[10] -tag
838   __orion_system_mem_if_ddr3_fpga_p0
839 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[12] -tag
840   __orion_system_mem_if_ddr3_fpga_p0
841 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[11] -tag
842   __orion_system_mem_if_ddr3_fpga_p0
843 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[13] -tag
844   __orion_system_mem_if_ddr3_fpga_p0
845 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a[14] -tag
846   __orion_system_mem_if_ddr3_fpga_p0
847 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_ba[2] -tag
848   __orion_system_mem_if_ddr3_fpga_p0
849 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_ba[0] -tag
850   __orion_system_mem_if_ddr3_fpga_p0
851 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_ba[1] -tag
852   __orion_system_mem_if_ddr3_fpga_p0
853 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dm[0] -tag
854   __orion_system_mem_if_ddr3_fpga_p0
855 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dm[1] -tag
856   __orion_system_mem_if_ddr3_fpga_p0
857 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dm[2] -tag
858   __orion_system_mem_if_ddr3_fpga_p0
859 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dm[3] -tag
860   __orion_system_mem_if_ddr3_fpga_p0
861 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[31] -tag
862   __orion_system_mem_if_ddr3_fpga_p0
863 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[0] -tag
864   __orion_system_mem_if_ddr3_fpga_p0
865 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[1] -tag
866   __orion_system_mem_if_ddr3_fpga_p0
867 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[2] -tag
868   __orion_system_mem_if_ddr3_fpga_p0
869 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[3] -tag
870   __orion_system_mem_if_ddr3_fpga_p0
871 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[4] -tag
872   __orion_system_mem_if_ddr3_fpga_p0
873 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[5] -tag
874   __orion_system_mem_if_ddr3_fpga_p0
875 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[6] -tag
876   __orion_system_mem_if_ddr3_fpga_p0
877 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[7] -tag
878   __orion_system_mem_if_ddr3_fpga_p0
879 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[8] -tag
880   __orion_system_mem_if_ddr3_fpga_p0

```

```

849 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[9] -tag
850   __orion_system_mem_if_ddr3_fpga_p0
851 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[10] -tag
852   __orion_system_mem_if_ddr3_fpga_p0
853 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[11] -tag
854   __orion_system_mem_if_ddr3_fpga_p0
855 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[12] -tag
856   __orion_system_mem_if_ddr3_fpga_p0
857 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[13] -tag
858   __orion_system_mem_if_ddr3_fpga_p0
859 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[14] -tag
860   __orion_system_mem_if_ddr3_fpga_p0
861 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[15] -tag
862   __orion_system_mem_if_ddr3_fpga_p0
863 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[16] -tag
864   __orion_system_mem_if_ddr3_fpga_p0
865 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[17] -tag
866   __orion_system_mem_if_ddr3_fpga_p0
867 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[18] -tag
868   __orion_system_mem_if_ddr3_fpga_p0
869 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[19] -tag
870   __orion_system_mem_if_ddr3_fpga_p0
871 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[20] -tag
872   __orion_system_mem_if_ddr3_fpga_p0
873 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[21] -tag
874   __orion_system_mem_if_ddr3_fpga_p0
875 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[22] -tag
876   __orion_system_mem_if_ddr3_fpga_p0
877 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[23] -tag
878   __orion_system_mem_if_ddr3_fpga_p0
879 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[24] -tag
880   __orion_system_mem_if_ddr3_fpga_p0
881 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[25] -tag
882   __orion_system_mem_if_ddr3_fpga_p0
883 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[26] -tag
884   __orion_system_mem_if_ddr3_fpga_p0
885 set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[27] -tag
886   __orion_system_mem_if_ddr3_fpga_p0
887 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
888   ddr3_fpga_dqs_n[0] -tag __orion_system_mem_if_ddr3_fpga_p0
889 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
890   ddr3_fpga_dqs_n[1] -tag __orion_system_mem_if_ddr3_fpga_p0
891 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
892   ddr3_fpga_dqs_n[2] -tag __orion_system_mem_if_ddr3_fpga_p0
893 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
894   ddr3_fpga_dqs_n[3] -tag __orion_system_mem_if_ddr3_fpga_p0
895 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
896   ddr3_fpga_dqs_p[0] -tag __orion_system_mem_if_ddr3_fpga_p0
897 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
898   ddr3_fpga_dqs_p[1] -tag __orion_system_mem_if_ddr3_fpga_p0
899 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
900   ddr3_fpga_dqs_p[2] -tag __orion_system_mem_if_ddr3_fpga_p0
901 set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
902   ddr3_fpga_dqs_p[3] -tag __orion_system_mem_if_ddr3_fpga_p0
903 set_instance_assignment -name IO_STANDARD LVDS -to pcie_refclk_n
904 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_rxd[0]
905 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_rxd[1]
906 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_rxd[2]
907 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_rxd[3]
908 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_txd[3]
909 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_txd[0]
910 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_txd[1]

```

```

887 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to enet_hps_txd[2]
888 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_p[3]
889 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_p[2]
890 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_p[1]
891 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_p[0]
892 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_n[3]
893 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_n[2]
894 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_n[1]
895 set_instance_assignment -name IO_STANDARD "1.5-V PCML" -to hsma_tx_n[0]
896 set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_sda
897 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to qspi_io[0]
898 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to qspi_io[1]
899 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to qspi_io[2]
900 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to qspi_io[3]
901 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_dat[0]
902 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_dat[1]
903 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_dat[2]
904 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to sd_dat[3]
905 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_pb_hps[0]
906 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_pb_hps[1]
907 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_dipsw_hps[0]
908 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_dipsw_hps[1]
909 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_dipsw_hps[2]
910 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_dipsw_hps[3]
911 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_pb_hps[2]
912 set_instance_assignment -name IO_STANDARD "1.8 V" -to user_pb_hps[3]
913 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_wrn
914 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_sda
915 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_scl
916 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_resetn
917 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_rdn
918 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_oen
919 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_full
920 set_instance_assignment -name IO_STANDARD "3.3-V LVC MOS" -to usb_empty
921 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[0] -tag __orion_system_mem_if_ddr3_fpga_p0
922 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[0] -tag __orion_system_mem_if_ddr3_fpga_p0
923 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[1] -tag __orion_system_mem_if_ddr3_fpga_p0
924 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[1] -tag __orion_system_mem_if_ddr3_fpga_p0
925 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[2] -tag __orion_system_mem_if_ddr3_fpga_p0
926 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[2] -tag __orion_system_mem_if_ddr3_fpga_p0
927 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[3] -tag __orion_system_mem_if_ddr3_fpga_p0
928 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[3] -tag __orion_system_mem_if_ddr3_fpga_p0
929 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[4] -tag __orion_system_mem_if_ddr3_fpga_p0
930 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[4] -tag __orion_system_mem_if_ddr3_fpga_p0
931 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[5] -tag __orion_system_mem_if_ddr3_fpga_p0
932 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[5] -tag __orion_system_mem_if_ddr3_fpga_p0
933 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[6] -tag __orion_system_mem_if_ddr3_fpga_p0
934 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[6] -tag __orion_system_mem_if_ddr3_fpga_p0
935 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[7] -tag __orion_system_mem_if_ddr3_fpga_p0
936 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[7] -tag __orion_system_mem_if_ddr3_fpga_p0
937 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to
    ddr3_fpga_dq[8] -tag __orion_system_mem_if_ddr3_fpga_p0

```



```
972 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
973 ddr3_fpga_dq[25] -tag __orion_system_mem_if_ddr3_fpga_p0  
974 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
975 ddr3_fpga_dq[26] -tag __orion_system_mem_if_ddr3_fpga_p0  
976 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
977 ddr3_fpga_dq[26] -tag __orion_system_mem_if_ddr3_fpga_p0  
978 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
979 ddr3_fpga_dq[27] -tag __orion_system_mem_if_ddr3_fpga_p0  
980 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
981 ddr3_fpga_dq[27] -tag __orion_system_mem_if_ddr3_fpga_p0  
982 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
983 ddr3_fpga_dq[28] -tag __orion_system_mem_if_ddr3_fpga_p0  
984 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
985 ddr3_fpga_dq[28] -tag __orion_system_mem_if_ddr3_fpga_p0  
986 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
987 ddr3_fpga_dq[29] -tag __orion_system_mem_if_ddr3_fpga_p0  
988 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
989 ddr3_fpga_dq[29] -tag __orion_system_mem_if_ddr3_fpga_p0  
990 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
991 ddr3_fpga_dqs_p[0] -tag __orion_system_mem_if_ddr3_fpga_p0  
992 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
993 ddr3_fpga_dqs_p[0] -tag __orion_system_mem_if_ddr3_fpga_p0  
994 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
995 ddr3_fpga_dqs_p[1] -tag __orion_system_mem_if_ddr3_fpga_p0  
996 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
997 ddr3_fpga_dqs_p[1] -tag __orion_system_mem_if_ddr3_fpga_p0  
998 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
999 ddr3_fpga_dqs_p[2] -tag __orion_system_mem_if_ddr3_fpga_p0  
1000 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
1001 ddr3_fpga_dqs_p[2] -tag __orion_system_mem_if_ddr3_fpga_p0  
1002 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
1003 ddr3_fpga_dqs_p[3] -tag __orion_system_mem_if_ddr3_fpga_p0  
1004 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
1005 ddr3_fpga_dqs_p[3] -tag __orion_system_mem_if_ddr3_fpga_p0  
1006 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
1007 ddr3_fpga_dqs_n[0] -tag __orion_system_mem_if_ddr3_fpga_p0  
1008 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
1009 ddr3_fpga_dqs_n[0] -tag __orion_system_mem_if_ddr3_fpga_p0  
1010 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
1011 ddr3_fpga_dqs_n[1] -tag __orion_system_mem_if_ddr3_fpga_p0  
1012 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
1013 ddr3_fpga_dqs_n[1] -tag __orion_system_mem_if_ddr3_fpga_p0  
1014 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
1015 ddr3_fpga_dqs_n[2] -tag __orion_system_mem_if_ddr3_fpga_p0  
1016 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
1017 ddr3_fpga_dqs_n[2] -tag __orion_system_mem_if_ddr3_fpga_p0  
1018 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH CALIBRATION" -to  
1019 ddr3_fpga_dqs_n[3] -tag __orion_system_mem_if_ddr3_fpga_p0  
1020 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to  
1021 ddr3_fpga_dqs_n[3] -tag __orion_system_mem_if_ddr3_fpga_p0  
1022 set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITHOUT CALIBRATION" -to  
1023 ddr3_fpga_clk_p -tag __orion_system_mem_if_ddr3_fpga_p0  
1024 set_instance_assignment -name D5_DELAY 2 -to ddr3_fpga_clk_p -tag  
1025 __orion_system_mem_if_ddr3_fpga_p0  
1026 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITHOUT CALIBRATION" -to  
1027 ddr3_fpga_clk_n -tag __orion_system_mem_if_ddr3_fpga_p0  
1028 set_instance_assignment -name D5_DELAY 2 -to ddr3_fpga_clk_n -tag  
1029 __orion_system_mem_if_ddr3_fpga_p0  
1030 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[0]  
1031 -tag __orion_system_mem_if_ddr3_fpga_p0
```

```

1006 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[10]
1007     -tag __orion_system_mem_if_ddr3_fpga_p0
1008 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[11]
1009     -tag __orion_system_mem_if_ddr3_fpga_p0
1010 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[12]
1011     -tag __orion_system_mem_if_ddr3_fpga_p0
1012 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[13]
1013     -tag __orion_system_mem_if_ddr3_fpga_p0
1014 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[14]
1015     -tag __orion_system_mem_if_ddr3_fpga_p0
1016 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[1]
1017     -tag __orion_system_mem_if_ddr3_fpga_p0
1018 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[2]
1019     -tag __orion_system_mem_if_ddr3_fpga_p0
1020 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[3]
1021     -tag __orion_system_mem_if_ddr3_fpga_p0
1022 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[4]
1023     -tag __orion_system_mem_if_ddr3_fpga_p0
1024 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[5]
1025     -tag __orion_system_mem_if_ddr3_fpga_p0
1026 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[6]
1027     -tag __orion_system_mem_if_ddr3_fpga_p0
1028 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[7]
1029     -tag __orion_system_mem_if_ddr3_fpga_p0
1030 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[8]
1031     -tag __orion_system_mem_if_ddr3_fpga_p0
1032 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[9]
1033     -tag __orion_system_mem_if_ddr3_fpga_p0
1034 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_ba[0]
1035     -tag __orion_system_mem_if_ddr3_fpga_p0
1036 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_ba[1]
1037     -tag __orion_system_mem_if_ddr3_fpga_p0
1038 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_ba[2]
1039     -tag __orion_system_mem_if_ddr3_fpga_p0
1040 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_casn
1041     -tag __orion_system_mem_if_ddr3_fpga_p0
1042 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_cke
1043     -tag __orion_system_mem_if_ddr3_fpga_p0
1044 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_csn
1045     -tag __orion_system_mem_if_ddr3_fpga_p0
1046 set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_odt
1047     -tag __orion_system_mem_if_ddr3_fpga_p0
1048 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
1049     ddr3_fpga_dm[0] -tag __orion_system_mem_if_ddr3_fpga_p0
1050 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
1051     ddr3_fpga_dm[1] -tag __orion_system_mem_if_ddr3_fpga_p0
1052 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
1053     ddr3_fpga_dm[2] -tag __orion_system_mem_if_ddr3_fpga_p0
1054 set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH CALIBRATION" -to
1055     ddr3_fpga_dm[3] -tag __orion_system_mem_if_ddr3_fpga_p0
1056 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[0] -tag
1057     __orion_system_mem_if_ddr3_fpga_p0
1058 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[1] -tag
1059     __orion_system_mem_if_ddr3_fpga_p0
1060 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[2] -tag
1061     __orion_system_mem_if_ddr3_fpga_p0
1062 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[3] -tag
1063     __orion_system_mem_if_ddr3_fpga_p0
1064 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[4] -tag
1065     __orion_system_mem_if_ddr3_fpga_p0
1066 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[5] -tag
1067     __orion_system_mem_if_ddr3_fpga_p0
1068 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[6] -tag
1069     __orion_system_mem_if_ddr3_fpga_p0
1070 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[7] -tag
1071     __orion_system_mem_if_ddr3_fpga_p0
1072 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dq[8] -tag
1073     __orion_system_mem_if_ddr3_fpga_p0

```



```

1074 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_dqs_n[3] -tag
1075   __orion_system_mem_if_ddr3_fpga_p0
1076 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[0] -tag
1077   __orion_system_mem_if_ddr3_fpga_p0
1078 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[10] -tag
1079   __orion_system_mem_if_ddr3_fpga_p0
1080 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[11] -tag
1081   __orion_system_mem_if_ddr3_fpga_p0
1082 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[12] -tag
1083   __orion_system_mem_if_ddr3_fpga_p0
1084 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[13] -tag
1085   __orion_system_mem_if_ddr3_fpga_p0
1086 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[14] -tag
1087   __orion_system_mem_if_ddr3_fpga_p0
1088 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[1] -tag
1089   __orion_system_mem_if_ddr3_fpga_p0
1090 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[2] -tag
1091   __orion_system_mem_if_ddr3_fpga_p0
1092 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[3] -tag
1093   __orion_system_mem_if_ddr3_fpga_p0
1094 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[4] -tag
1095   __orion_system_mem_if_ddr3_fpga_p0
1096 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[5] -tag
1097   __orion_system_mem_if_ddr3_fpga_p0
1098 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[6] -tag
1099   __orion_system_mem_if_ddr3_fpga_p0
1100 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[7] -tag
1101   __orion_system_mem_if_ddr3_fpga_p0
1102 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[8] -tag
1103   __orion_system_mem_if_ddr3_fpga_p0
1104 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_a[9] -tag
1105   __orion_system_mem_if_ddr3_fpga_p0
1106 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_ba[0] -tag
1107   __orion_system_mem_if_ddr3_fpga_p0
1108 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_ba[1] -tag
1109   __orion_system_mem_if_ddr3_fpga_p0
1110 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_ba[2] -tag
1111   __orion_system_mem_if_ddr3_fpga_p0
1112 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_casn -tag
1113   __orion_system_mem_if_ddr3_fpga_p0
1114 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_cke -tag
1115   __orion_system_mem_if_ddr3_fpga_p0
1116 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_csn -tag
1117   __orion_system_mem_if_ddr3_fpga_p0
1118 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_odt -tag
1119   __orion_system_mem_if_ddr3_fpga_p0
1120 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_clk_p -tag
1121   __orion_system_mem_if_ddr3_fpga_p0
1122 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_fpga_clk_n -tag
1123   __orion_system_mem_if_ddr3_fpga_p0
1124 set_instance_assignment -name GLOBAL_SIGNAL "DUAL-REGIONAL CLOCK" -to u0|mem_if_ddr3_fpga|
1125   p110|pll_lv1_clk -tag __orion_system_mem_if_ddr3_fpga_p0
1126 set_instance_assignment -name GLOBAL_SIGNAL "DUAL-REGIONAL CLOCK" -to u0|mem_if_ddr3_fpga|
1127   p110|pll_config_clk -tag __orion_system_mem_if_ddr3_fpga_p0
1128 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|ureset|
1129   phy_reset_mem_stable_n -tag __orion_system_mem_if_ddr3_fpga_p0
1130 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|ureset|
1131   phy_reset_n -tag __orion_system_mem_if_ddr3_fpga_p0
1132 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|s0|
1133   sequencer_rw_mgr_inst|rw_mgr_inst|rw_mgr_core_inst|rw_soft_reset_n -tag
1134   __orion_system_mem_if_ddr3_fpga_p0
1135 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
1136   uiio_pads|dq_ddio[0].read_capture_clk_buffer -tag __orion_system_mem_if_ddr3_fpga_p0
1137 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
1138   uread_datapath|reset_n_fifo_write_side[0] -tag __orion_system_mem_if_ddr3_fpga_p0
1139 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
1140   uread_datapath|reset_n_fifo_wraddress[0] -tag __orion_system_mem_if_ddr3_fpga_p0

```

```

1107 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uio_pads|dq_ddio[1].read_capture_clk_buffer -tag __orion_system_mem_if_ddr3_fpga_p0
1108 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uread_datapath|reset_n_fifo_write_side[1] -tag __orion_system_mem_if_ddr3_fpga_p0
1109 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uread_datapath|reset_n_fifo_wraddress[1] -tag __orion_system_mem_if_ddr3_fpga_p0
1110 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uio_pads|dq_ddio[2].read_capture_clk_buffer -tag __orion_system_mem_if_ddr3_fpga_p0
1111 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uread_datapath|reset_n_fifo_write_side[2] -tag __orion_system_mem_if_ddr3_fpga_p0
1112 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uread_datapath|reset_n_fifo_wraddress[2] -tag __orion_system_mem_if_ddr3_fpga_p0
1113 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uio_pads|dq_ddio[3].read_capture_clk_buffer -tag __orion_system_mem_if_ddr3_fpga_p0
1114 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uread_datapath|reset_n_fifo_write_side[3] -tag __orion_system_mem_if_ddr3_fpga_p0
1115 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|mem_if_ddr3_fpga|p0|umemphy|
    uread_datapath|reset_n_fifo_wraddress[3] -tag __orion_system_mem_if_ddr3_fpga_p0
1116 set_instance_assignment -name ENABLE_BENEFICIAL_SKEW_OPTIMIZATION_FOR_NON_GLOBAL_CLOCKS ON
    -to u0|mem_if_ddr3_fpga -tag __orion_system_mem_if_ddr3_fpga_p0
1117 set_instance_assignment -name PLL_COMPENSATION_MODE DIRECT -to u0|mem_if_ddr3_fpga|p110|
    fbout -tag __orion_system_mem_if_ddr3_fpga_p0
1118 set_instance_assignment -name D5_DELAY 2 -to ddr3_hps_clk_p -tag __hps_sdram_p0
1119 set_instance_assignment -name D5_DELAY 2 -to ddr3_hps_clk_n -tag __hps_sdram_p0
1120 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[0] -tag
    __hps_sdram_p0
1121 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[1] -tag
    __hps_sdram_p0
1122 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[2] -tag
    __hps_sdram_p0
1123 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[3] -tag
    __hps_sdram_p0
1124 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[4] -tag
    __hps_sdram_p0
1125 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[5] -tag
    __hps_sdram_p0
1126 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[6] -tag
    __hps_sdram_p0
1127 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[7] -tag
    __hps_sdram_p0
1128 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[8] -tag
    __hps_sdram_p0
1129 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[9] -tag
    __hps_sdram_p0
1130 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[10] -tag
    __hps_sdram_p0
1131 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[11] -tag
    __hps_sdram_p0
1132 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[12] -tag
    __hps_sdram_p0
1133 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[13] -tag
    __hps_sdram_p0
1134 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[14] -tag
    __hps_sdram_p0
1135 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[15] -tag
    __hps_sdram_p0
1136 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[16] -tag
    __hps_sdram_p0
1137 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[17] -tag
    __hps_sdram_p0
1138 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[18] -tag
    __hps_sdram_p0
1139 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[19] -tag
    __hps_sdram_p0
1140 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[20] -tag
    __hps_sdram_p0
1141 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[21] -tag
    __hps_sdram_p0

```

```

1142 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[22] -tag
1143   --hps_sdram_p0
1144 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[23] -tag
1145   --hps_sdram_p0
1146 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[24] -tag
1147   --hps_sdram_p0
1148 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[25] -tag
1149   --hps_sdram_p0
1150 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[26] -tag
1151   --hps_sdram_p0
1152 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[27] -tag
1153   --hps_sdram_p0
1154 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[28] -tag
1155   --hps_sdram_p0
1156 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[29] -tag
1157   --hps_sdram_p0
1158 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[30] -tag
1159   --hps_sdram_p0
1160 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[31] -tag
1161   --hps_sdram_p0
1162 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[32] -tag
1163   --hps_sdram_p0
1164 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[33] -tag
1165   --hps_sdram_p0
1166 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[34] -tag
1167   --hps_sdram_p0
1168 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[35] -tag
1169   --hps_sdram_p0
1170 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[36] -tag
1171   --hps_sdram_p0
1172 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[37] -tag
1173   --hps_sdram_p0
1174 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[38] -tag
1175   --hps_sdram_p0
1176 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[0] -tag
1177   --hps_sdram_p0
1178 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[1] -tag
1179   --hps_sdram_p0
1180 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[2] -tag
1181   --hps_sdram_p0
1182 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[3] -tag
1183   --hps_sdram_p0
1184 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[4] -tag
1185   --hps_sdram_p0
1186 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_p[0] -tag
1187   --hps_sdram_p0
1188 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_p[1] -tag
1189   --hps_sdram_p0
1190 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_p[2] -tag
1191   --hps_sdram_p0
1192 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_p[3] -tag
1193   --hps_sdram_p0
1194 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_p[4] -tag
1195   --hps_sdram_p0
1196 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_n[0] -tag
1197   --hps_sdram_p0
1198 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_n[1] -tag
1199   --hps_sdram_p0
1200 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_n[2] -tag
1201   --hps_sdram_p0
1202 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_n[3] -tag
1203   --hps_sdram_p0
1204 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs_n[4] -tag
1205   --hps_sdram_p0
1206 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[0] -tag
1207   --hps_sdram_p0

```

```

1176 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[10] -tag
1177   __hps_sdram_p0
1178 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[11] -tag
1179   __hps_sdram_p0
1180 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[12] -tag
1181   __hps_sdram_p0
1182 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[13] -tag
1183   __hps_sdram_p0
1184 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[14] -tag
1185   __hps_sdram_p0
1186 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[1] -tag
1187   __hps_sdram_p0
1188 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[2] -tag
1189   __hps_sdram_p0
1190 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[3] -tag
1191   __hps_sdram_p0
1192 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[4] -tag
1193   __hps_sdram_p0
1194 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[5] -tag
1195   __hps_sdram_p0
1196 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[6] -tag
1197   __hps_sdram_p0
1198 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[7] -tag
1199   __hps_sdram_p0
1200 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[8] -tag
1201   __hps_sdram_p0
1202 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_a[9] -tag
1203   __hps_sdram_p0
1204 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ba[0] -tag
1205   __hps_sdram_p0
1206 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ba[1] -tag
1207   __hps_sdram_p0
1208 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ba[2] -tag
1209   __hps_sdram_p0
1210 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_casn -tag
1211   __hps_sdram_p0
1212 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_cke -tag
1213   __hps_sdram_p0
1214 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_csn -tag
1215   __hps_sdram_p0
1216 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_odt -tag
1217   __hps_sdram_p0
1218 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_rasn -tag
1219   __hps_sdram_p0
1220 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_wen -tag
1221   __hps_sdram_p0
1222 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_resetn -tag
1223   __hps_sdram_p0
1224 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_clk_p -tag
1225   __hps_sdram_p0
1226 set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_clk_n -tag
1227   __hps_sdram_p0
1228 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1229   p0|umemphy|ureset|phy_reset_mem_stable_n -tag __hps_sdram_p0
1230 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1231   p0|umemphy|ureset|phy_reset_n -tag __hps_sdram_p0
1232 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1233   p0|umemphy|uiio_pads|dq_ddio[0].read_capture_clk_buffer -tag __hps_sdram_p0
1234 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1235   p0|umemphy|uread_datapath|reset_n_fifo_write_side[0] -tag __hps_sdram_p0
1236 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1237   p0|umemphy|uread_datapath|reset_n_fifo_wraddress[0] -tag __hps_sdram_p0
1238 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1239   p0|umemphy|uiio_pads|dq_ddio[1].read_capture_clk_buffer -tag __hps_sdram_p0
1240 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1241   p0|umemphy|uread_datapath|reset_n_fifo_write_side[1] -tag __hps_sdram_p0
1242 set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1243   p0|umemphy|uread_datapath|reset_n_fifo_wraddress[1] -tag __hps_sdram_p0

```

```

1210 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1211 |   p0|umemphy|ui0_pads|dq_ddio[2].read_capture_clk_buffer -tag __hps_sdram_p0
1212 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1213 |   p0|umemphy|uread_datapath|reset_n_fifo_write_side[2] -tag __hps_sdram_p0
1214 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1215 |   p0|umemphy|uread_datapath|reset_n_fifo_wraddress[2] -tag __hps_sdram_p0
1216 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1217 |   p0|umemphy|ui0_pads|dq_ddio[3].read_capture_clk_buffer -tag __hps_sdram_p0
1218 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1219 |   p0|umemphy|uread_datapath|reset_n_fifo_write_side[3] -tag __hps_sdram_p0
1220 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1221 |   p0|umemphy|uread_datapath|reset_n_fifo_wraddress[3] -tag __hps_sdram_p0
1222 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1223 |   p0|umemphy|ui0_pads|dq_ddio[4].read_capture_clk_buffer -tag __hps_sdram_p0
1224 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1225 |   p0|umemphy|uread_datapath|reset_n_fifo_write_side[4] -tag __hps_sdram_p0
1226 | set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst|
1227 |   p0|umemphy|uread_datapath|reset_n_fifo_wraddress[4] -tag __hps_sdram_p0
1228 | set_instance_assignment -name PLL_COMPENSATION_MODE DIRECT -to u0|hps_0|hps_io|border|
1229 |   hps_sdram_inst|pll10|fbout -tag __hps_sdram_p0
1230 | set_location_assignment PIN_W16 -to ddr3_fpga_dqs_n[0]
1231 | set_location_assignment PIN_W17 -to ddr3_fpga_dqs_n[1]
1232 | set_location_assignment PIN_AA18 -to ddr3_fpga_dqs_n[2]
1233 | set_location_assignment PIN_AD19 -to ddr3_fpga_dqs_n[3]
1234 | set_location_assignment PIN_V16 -to ddr3_fpga_dqs_p[0]
1235 | set_location_assignment PIN_V17 -to ddr3_fpga_dqs_p[1]
1236 | set_location_assignment PIN_Y17 -to ddr3_fpga_dqs_p[2]
1237 | set_location_assignment PIN_AC20 -to ddr3_fpga_dqs_p[3]
1238 | set_location_assignment PIN_H3 -to hsma_tx_p[3]
1239 | set_location_assignment PIN_K3 -to hsma_tx_p[2]
1240 | set_location_assignment PIN_M3 -to hsma_tx_p[1]
1241 | set_location_assignment PIN_P3 -to hsma_tx_p[0]
1242 | set_location_assignment PIN_H4 -to hsma_tx_n[3]
1243 | set_location_assignment PIN_K4 -to hsma_tx_n[2]
1244 | set_location_assignment PIN_M4 -to hsma_tx_n[1]
1245 | set_location_assignment PIN_P4 -to hsma_tx_n[0]
1246 | set_global_assignment -name PHYSICAL_SYNTHESIS_COMBO_LOGIC ON
1247 | set_global_assignment -name PHYSICAL_SYNTHESIS_REGISTER_RETIMING ON
1248 | set_global_assignment -name PHYSICAL_SYNTHESIS_ASYNCNOMOUS_SIGNAL_PIPELINING ON
1249 | set_global_assignment -name PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION ON
1250 | set_global_assignment -name PHYSICAL_SYNTHESIS_COMBO_LOGIC_FOR_AREA ON
1251 | set_global_assignment -name PHYSICAL_SYNTHESIS_EFFORT EXTRA
1252 | set_global_assignment -name OPTIMIZE_HOLD_TIMING "ALL PATHS"
1253 | set_global_assignment -name OPTIMIZE_POWER_DURING_FITTING OFF
1254 | set_global_assignment -name FITTER EFFORT "STANDARD FIT"
1255 | set_global_assignment -name ROUTER_TIMING_OPTIMIZATION_LEVEL MAXIMUM
1256 | set_global_assignment -name ROUTER_CLOCKING_TOPOLOGY_ANALYSIS ON
1257 | set_global_assignment -name AUTO_PACKED_REGISTERS_STRATIXII NORMAL
1258 | set_global_assignment -name ROUTER_LCELL_INSERTION_AND_LOGIC_DUPLICATION ON
1259 | set_instance_assignment -name PLL_AUTO_RESET OFF -to "*q_sys_pll_0*|"
1260 |   altera_pll:altera_pll_i_*|*"
1261 | set_instance_assignment -name PLL_BANDWIDTH_PRESET AUTO -to "*q_sys_pll_0*|"
1262 |   altera_pll:altera_pll_i_*|*"
1263 | set_instance_assignment -name PLL_COMPENSATION_MODE NORMAL -to "*q_sys_pll_0*|"
1264 |   altera_pll:altera_pll_i_*|*"
1265 | set_global_assignment -name SEED 100
1266 | set_global_assignment -name SDC_FILE timing.base.sdc
1267 | set_global_assignment -name VERILOGFILE orion_top.v
1268 | set_global_assignment -name QIPFILE orion_system/synthesis/orion_system.qip
1269 | set_global_assignment -name VERILOGFILE ip/debounce/debounce.v
1270 | set_global_assignment -name QIPFILE clk_cntrl.qip
1271 | set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top

```

Listing 16.6: Quartus II Settings File (orion_system.qsf)

16.3.8 Demo IP Top Verilog File (test.ip_top.v)

```

1  /*
2  Christopher Yarp
3
4  This file was created based on the Altera custom_master template.
5  This file supports multiple ports and connects to the DSP logic
6  directly without requiring an export from qsys and attachment
7  in Quartus.
8 */
9
10 module test_ip_top (
11   clk ,
12   reset ,
13
14   //=====BEGIN WRITE MASTER=====
15   // master inputs and outputs
16   master_write_address ,
17   // master_write_read ,
18   master_write_write ,
19   master_write_bytelenable ,
20   // master_write_readdata ,
21   // master_write_readdatavalid ,
22   master_write_writedata ,
23   master_write_burstcount ,
24   master_write_waitrequest ,
25   //=====END WRITE MASTER=====
26
27   //=====BEGIN READ MASTER=====
28   // master inputs and outputs
29   master_read_address ,
30   master_read_read ,
31   // master_read_write ,
32   master_read_bytelenable ,
33   master_read_readdata ,
34   master_read_readdatavalid ,
35   // master_read_writedata ,
36   master_read_burstcount ,
37   master_read_waitrequest ,
38   //=====END READ MASTER=====
39
40   //=====BEGIN SLAVE=====
41   // signals to connect to an Avalon-MM slave interface
42   slave_address ,
43   slave_read ,
44   slave_write ,
45   slave_readdata ,
46   slave_writedata ,
47   slave_bytelenable ,
48
49   // interrupt signals
50   slave_irq
51   //=====END SLAVE=====
52 );
53
54 parameter DATA_WIDTH = 32;
55 parameter ADDRESS_WIDTH = 32;
56
57 //=====BEGIN WRITE MASTER PARMS=====
58 parameter MASTER_WRITE_MEMORY_BASED_FIFO = 1;           // 0 for LE/ALUT FIFOs , 1 for
      memory FIFOs (highly recommend 1)
59 parameter MASTER_WRITE_FIFO_DEPTH = 32;
60 parameter MASTER_WRITE_FIFO_DEPTH_LOG2 = 5;
61 parameter MASTER_WRITE_BURST_CAPABLE = 0;               // 1 to enable burst , 0 to disable
      it
62 parameter MASTER_WRITE_MAXIMUM_BURST_COUNT = 2;
63 parameter MASTER_WRITE_BURST_COUNT_WIDTH = 2;
64 //=====END WRITE MASTER PARMS=====
65

```

```

66 //=====BEGIN READ MASTER PARMS=====
67 parameter MASTER_READ_MEMORY_BASED_FIFO = 1; // 0 for LE/ALUT FIFOs , 1 for
68   memory FIFOs ( highly recommend 1)
69 parameter MASTER_READ_FIFO_DEPTH = 32;
70 parameter MASTER_READ_FIFO_DEPTH_LOG2 = 5;
71 parameter MASTER_READ_BURST_CAPABLE = 0; // 1 to enable burst , 0 to disable
72   it
73 parameter MASTER_READ_MAXIMUM_BURST_COUNT = 2;
74 parameter MASTER_READ_BURST_COUNT_WIDTH = 2;
75 //=====END READ MASTER PARMS=====

76 //=====BEGIN SLAVE PARMS=====
77 parameter ENABLE_SYNC_SIGNALS = 0; // only used by the component .tcl file , 1 to expose
78   user_chipselect/write/read , 0 to stub them
79 parameter IRQ_EN = 0; // 0 = Enable interrupt , 1 = Disable interrupt
80 //=====END SLAVE PARMS=====

81 input clk;
82 input reset;

83 //=====BEGIN WRITE MASTER WIRES=====
84 // control inputs and outputs
85 wire master_write_control_fixed_location;
86 //wire [ADDRESS_WIDTH-1:0] master_write_control_read_base; // for read master
87 //wire [ADDRESS_WIDTH-1:0] master_write_control_read_length; // for read master
88 wire [ADDRESS_WIDTH-1:0] master_write_control_write_base; // for write master
89 wire [ADDRESS_WIDTH-1:0] master_write_control_write_length; // for write master
90 wire master_write_control_go;
91 /*out*/wire master_write_control_done;
92 /*out*/wire master_write_control_early_done; // for read master

93 // user logic inputs and outputs
94 //wire master_write_user_read_buffer; // for read master
95 wire master_write_user_write_buffer; // for write master
96 wire [DATA_WIDTH-1:0] master_write_user_buffer_input_data; // for write master
97 /*out*/wire [DATA_WIDTH-1:0] master_write_user_buffer_output_data; // for read master
98 /*out*/wire master_write_user_data_available; // for read master
99 /*out*/wire master_write_user_buffer_full; // for write master

100 // master inputs and outputs
101 output wire [ADDRESS_WIDTH-1:0] master_write_address;
102 //output wire master_write_read; // for read master
103 output wire master_write_write; // for write master
104 output wire [(DATA_WIDTH/8)-1:0] master_write_byteenable;
105 //input [DATA_WIDTH-1:0] master_write_readdatavalid; // for read master
106 //input master_write_readdatavalid; // for read master
107 output wire [DATA_WIDTH-1:0] master_write_writedata; // for write master
108 output wire [MASTER_WRITE_BURST_COUNT_WIDTH-1:0] master_write_burstcount; // for
109   bursting read and write masters
110 input master_write_waitrequest;
111 //=====END WRITE MASTER WIRES=====

112 //=====BEGIN READ MASTER WIRES=====
113 // control inputs and outputs
114 wire master_read_control_fixed_location;
115 //wire [ADDRESS_WIDTH-1:0] master_read_control_read_base; // for read master
116 //wire [ADDRESS_WIDTH-1:0] master_read_control_read_length; // for read master
117 //wire [ADDRESS_WIDTH-1:0] master_read_control_write_base; // for write master
118 //wire [ADDRESS_WIDTH-1:0] master_read_control_write_length; // for write master
119 wire master_read_control_go;
120 /*out*/wire master_read_control_done;
121 /*out*/wire master_read_control_early_done; // for read master

122 // user logic inputs and outputs
123 wire master_read_user_read_buffer; // for read master
124 //wire master_read_user_write_buffer; // for write master
125 //wire [DATA_WIDTH-1:0] master_read_user_buffer_input_data; // for write master
126 /*out*/wire [DATA_WIDTH-1:0] master_read_user_buffer_output_data; // for read master

```

```

130  /* out */ wire master_read_user_data_available;           // for read master
131  ///* out */ wire master_read_user_buffer_full;             // for write master
132
133 // master inputs and outputs
134 output wire [ADDRESS_WIDTH-1:0] master_read_address;
135 output wire master_read_read;                            // for read master
136 //output wire master_read_write;                         // for write master
137 output wire [(DATA_WIDTH/8)-1:0] master_read_byteenable;
138 input [DATA_WIDTH-1:0] master_read_readdata;            // for read master
139 input master_read_readdatavalid;                       // for read master
140 //output wire [DATA_WIDTH-1:0] master_read_writedata;    // for write master
141 output wire [MASTER_READ_BURST_COUNT_WIDTH-1:0] master_read_burstcount; // for bursting
142     read and write masters
143 input master_read_waitrequest;
144 //=====END READ MASTER WIRES=====
145
146 //=====BEGIN SLAVE WIRES=====
147 input [8:0] slave_address;
148 input slave_read;
149 input slave_write;
150 output wire [DATA_WIDTH-1:0] slave_readdata;
151 input [DATA_WIDTH-1:0] slave_writedata;
152 input [(DATA_WIDTH/8)-1:0] slave_byteenable;
153 output wire slave_irq;
154
155 // user interface
156 /*output*/ wire [DATA_WIDTH-1:0] user_dataout_0;
157 /*output*/ wire [DATA_WIDTH-1:0] user_dataout_1;
158 /*output*/ wire [DATA_WIDTH-1:0] user_dataout_2;
159 /*output*/ wire [DATA_WIDTH-1:0] user_dataout_3;
160 /*output*/ wire [DATA_WIDTH-1:0] user_dataout_4;
161 /*output*/ wire [DATA_WIDTH-1:0] user_dataout_5;
162 wire [DATA_WIDTH-1:0] user_datain_0;
163 wire [DATA_WIDTH-1:0] user_datain_1;
164 wire [DATA_WIDTH-1:0] user_datain_2;
165 wire [DATA_WIDTH-1:0] user_datain_3;
166 wire [DATA_WIDTH-1:0] user_datain_4;
167 wire [DATA_WIDTH-1:0] user_datain_5;
168 /*output*/ wire [15:0] user_chipselect;
169 /*output*/ wire [(DATA_WIDTH/8)-1:0] user_byteenable;
170 /*output*/ wire user_write;
171 /*output*/ wire user_read;
172
173 //=====BEGIN WRITE MASTER MODULE=====
174 custom_master write_master(
175     .clk(clk),
176     .reset(reset),
177
178 // control inputs and outputs
179     .control_fixed_location(master_write_control_fixed_location),
180     .control_read_base(),
181     .control_read_length(),
182     .control_write_base(master_write_control_write_base),
183     .control_write_length(master_write_control_write_length),
184     .control_go(master_write_control_go),
185     .control_done(master_write_control_done),
186     .control_early_done(),
187
188 // user logic inputs and outputs
189     .user_read_buffer(),
190     .user_write_buffer(master_write_user_write_buffer),
191     .user_buffer_input_data(master_write_user_buffer_input_data),
192     .user_buffer_output_data(),
193     .user_data_available(),
194     .user_buffer_full(master_write_user_buffer_full),
195
196 // master inputs and outputs

```

```

197     .master_address(master_write_address),
198     .master_read(),
199     .master_write(master_write_write),
200     .master_byteenable(master_write_byteenable),
201     .master_readdata(),
202     .master_readdatavalid(),
203     .master_writedata(master_write_writedata),
204     .master_burstcount(master_write_burstcount),
205     .master_waitrequest(master_write_waitrequest)
206   );
207
208   defparam write_master.MASTER_DIRECTION = 1; // 0 for read master, 1 for
209   // write master
210   defparam write_master.DATA_WIDTH = DATA_WIDTH;
211   defparam write_master.MEMORY_BASED.FIFO = MASTER_WRITE_MEMORY_BASED_FIFO; // 0
212   // for LE/ALUT FIFOs, 1 for memory FIFOs (highly recommend 1)
213   defparam write_master.FIFO_DEPTH = MASTER_WRITE_FIFO_DEPTH;
214   defparam write_master.FIFO_DEPTH_LOG2 = MASTER_WRITE_FIFO_DEPTH_LOG2;
215   defparam write_master.ADDRESS_WIDTH = ADDRESS_WIDTH;
216   defparam write_master.BURST_CAPABLE = MASTER_WRITE_BURST_CAPABLE; // 1 to
217   // enable burst, 0 to disable it
218   defparam write_master.MAXIMUM_BURST_COUNT = MASTER_WRITE_MAXIMUM_BURST_COUNT;
219   defparam write_master.BURST_COUNT_WIDTH = MASTER_WRITE_BURST_COUNT_WIDTH;
220   //=====END WRITE MASTER MODULE=====
221
222   //=====BEGIN READ MASTER MODULE=====
223   custom_master read_master(
224     .clk(clk),
225     .reset(reset),
226
227     // control inputs and outputs
228     .control_fixed_location(master_read_control_fixed_location),
229     .control_read_base(master_read_control_read_base),
230     .control_read_length(master_read_control_read_length),
231     .control_write_base(),
232     .control_write_length(),
233     .control_go(master_read_control_go),
234     .control_done(master_read_control_done),
235     .control_early_done(master_read_control_early_done),
236
237     // user logic inputs and outputs
238     .user_read_buffer(master_read_user_read_buffer),
239     .user_write_buffer(),
240     .user_buffer_input_data(),
241     .user_buffer_output_data(master_read_user_buffer_output_data),
242     .user_data_available(master_read_user_data_available),
243     .user_buffer_full(),
244
245     // master inputs and outputs
246     .master_address(master_read_address),
247     .master_read(master_read_read),
248     .master_write(),
249     .master_byteenable(master_read_byteenable),
250     .master_readdata(master_read_readdata),
251     .master_readdatavalid(master_read_readdatavalid),
252     .master_writedata(),
253     .master_burstcount(master_read_burstcount),
254     .master_waitrequest(master_read_waitrequest)
255   );
256
257   defparam read_master.MASTER_DIRECTION = 0; // 0 for read master, 1 for
258   // write master
259   defparam read_master.DATA_WIDTH = DATA_WIDTH;
260   defparam read_master.MEMORY_BASED.FIFO = MASTER_READ_MEMORY_BASED_FIFO; // 0
261   // for LE/ALUT FIFOs, 1 for memory FIFOs (highly recommend 1)
262   defparam read_master.FIFO_DEPTH = MASTER_READ_FIFO_DEPTH;
263   defparam read_master.FIFO_DEPTH_LOG2 = MASTER_READ_FIFO_DEPTH_LOG2;

```

```

260 defparam read_master.ADDRESS.WIDTH = ADDRESS_WIDTH;
261 defparam read_master.BURST.CAPABLE = MASTER.READ.BURST.CAPABLE; // 1 to
262     enable burst, 0 to disable it
263 defparam read_master.MAXIMUMBURST.COUNT = MASTER.READ.MAXIMUMBURST.COUNT;
264 defparam read_master.BURST.COUNT.WIDTH = MASTER.READ.BURST.COUNT.WIDTH;
265 // =====END READ MASTER MODULE=====
266
267 //=====BEGIN SLAVE MODULE=====
268 slave_template csr(
269 // signals to connect to an Avalon clock source interface
270 .clk(clk),
271 .reset(reset),
272
273 // signals to connect to an Avalon-MM slave interface
274 .slave_address(slave_address),
275 .slave_read(slave_read),
276 .slave_write(slave_write),
277 .slave_readdata(slave_readdata),
278 .slave_writedata(slave_writedata),
279 .slave_byteenable(slave_byteenable),
280
281 // interrupt signals
282 .slave_irq(slave_irq),
283
284 // signals to connect to custom user logic (up to 16 input and output pairs)
285 .user_dataout_0(user_dataout_0), //address
286 .user_dataout_1(user_dataout_1), //mode
287 .user_dataout_2(user_dataout_2), //data0
288 .user_dataout_3(user_dataout_3), //data1
289 .user_dataout_4(user_dataout_4), //data2
290 .user_dataout_5(user_dataout_5), //data3
291 .user_dataout_6(),
292 .user_dataout_7(),
293 .user_dataout_8(),
294 .user_dataout_9(),
295 .user_dataout_10(),
296 .user_dataout_11(),
297 .user_dataout_12(),
298 .user_dataout_13(),
299 .user_dataout_14(),
300 .user_dataout_15(),
301 .user_datain_0(user_datain_0), //readReady
302 .user_datain_1(user_datain_1), //writeReady
303 .user_datain_2(user_datain_2), //data0
304 .user_datain_3(user_datain_3), //data1
305 .user_datain_4(user_datain_4), //data2
306 .user_datain_5(user_datain_5), //data3
307 .user_datain_6(),
308 .user_datain_7(),
309 .user_datain_8(),
310 .user_datain_9(),
311 .user_datain_10(),
312 .user_datain_11(),
313 .user_datain_12(),
314 .user_datain_13(),
315 .user_datain_14(),
316 .user_datain_15(),
317
318 // optional signals so that your external logic knows what location is being accessed
319 .user_chipselect(user_chipselect),
320 .user_byteenable(user_byteenable),
321 .user_write(user_write),
322 .user_read(user_read)
323 );
324 defparam csr.DATA.WIDTH = DATA.WIDTH; // word size of each input and output
325     register
326 defparam csr.ENABLE_SYNC_SIGNALS = ENABLE_SYNC_SIGNALS; // only used by the component .
327     tcl file, 1 to expose user_chipselect/write/read , 0 to stub them

```

```

325 defparam csr.MODE_0 = 2; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
326     = Output with loopback , 4 = Disabled
327 defparam csr.MODE_1 = 2; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
328     = Output with loopback , 4 = Disabled
329 defparam csr.MODE_2 = 2; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
330     = Output with loopback , 4 = Disabled
331 defparam csr.MODE_3 = 2; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
332     = Output with loopback , 4 = Disabled
333 defparam csr.MODE_4 = 2; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
334     = Output with loopback , 4 = Disabled
335 defparam csr.MODE_5 = 2; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
336     = Output with loopback , 4 = Disabled
337 defparam csr.MODE_6 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
338     = Output with loopback , 4 = Disabled
339 defparam csr.MODE_7 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
340     = Output with loopback , 4 = Disabled
341 defparam csr.MODE_8 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
342     = Output with loopback , 4 = Disabled
343 defparam csr.MODE_9 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
344     = Output with loopback , 4 = Disabled
345 defparam csr.MODE_10 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
346     = Output with loopback , 4 = Disabled
347 defparam csr.MODE_11 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
348     = Output with loopback , 4 = Disabled
349 defparam csr.MODE_12 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
350     = Output with loopback , 4 = Disabled
351 defparam csr.MODE_13 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
352     = Output with loopback , 4 = Disabled
353 defparam csr.MODE_14 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
354     = Output with loopback , 4 = Disabled
355 defparam csr.MODE_15 = 4; // 0 = Output , 1 = Input , 2 = Ouput and Input , 3
356     = Output with loopback , 4 = Disabled
357 defparam csr.IRQ_EN = IRQ_EN; // 0 = Enable interrupt , 1 = Disable interrupt
358
359 //=====END SLAVE MODULE=====
360 assign user_datain_1[31:1] = 0;
361 assign user_datain_0[31:1] = 0;
362 //=====BEGIN TEST IP MODULE=====
363 test_ip test_ip1(
364 .clk(clk),
365 .reset(reset),
366
367 // ***Base Module Logic I/O***
368
369 // inputs
370 .address(user_dataout_0),
371
372 .writeData0(user_dataout_2),
373 .writeData1(user_dataout_3),
374 .writeData2(user_dataout_4),
375 .writeData3(user_dataout_5),
376
377 .mode(user_dataout_1), //0 for read , 1 for write
378 .modeAccessed(user_chipselect[1]&&user_read),
379
380 // outputs
381 .readData0(user_datain_2),
382 .readData1(user_datain_3),
383 .readData2(user_datain_4),
384 .readData3(user_datain_5),
385
386 .writeReady(user_datain_1[0]),
387 .readReady(user_datain_0[0]),
388
389 // ***Memory Logic***
390 // write logic
391 .master_write_control_fixed_location(master_write_control_fixed_location),
392 .master_write_control_write_base(master_write_control_write_base),
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
999

```

```

377 . master_write_control_write_length(master_write_control_write_length),
378 . master_write_user_write_buffer(master_write_user_write_buffer),
379 . master_write_user_buffer_input_data(master_write_user_buffer_input_data),
380 . master_write_user_buffer_full(master_write_user_buffer_full),
381 . master_write_user_control_go(master_write_user_control_go),
382 . master_write_user_control_done(master_write_user_control_done),
383
384 // read logic
385 . master_read_control_fixed_location(master_read_control_fixed_location),
386 . master_read_control_read_base(master_read_control_read_base),
387 . master_read_control_read_length(master_read_control_read_length),
388 . master_read_user_read_buffer(master_read_user_read_buffer),
389 . master_read_user_buffer_output_data(master_read_user_buffer_output_data),
390 . master_read_user_data_available(master_read_user_data_available),
391 . master_read_user_control_go(master_read_user_control_go),
392 . master_read_user_control_done(master_read_user_control_done)
393 );
394
395 //=====END TEST IP MODULE=====
396
397 endmodule

```

Listing 16.7: Demo IP Top Verilog File (test_ip_top.v)

16.3.9 Demo IP Top Verilog File (test_ip.v)

```

1 module test_ip(
2   clk,
3   reset,
4
5   // ***Base Module Logic I/O***
6
7   // inputs
8   address,
9
10  writeData0,
11  writeData1,
12  writeData2,
13  writeData3,
14
15  mode, //0 for read, 1 for write
16  modeAccessed,
17
18  // outputs
19  readData0,
20  readData1,
21  readData2,
22  readData3,
23
24  writeReady,
25  readReady,
26
27  // ***Memory Logic***
28  // write logic
29  master_write_control_fixed_location,
30  master_write_control_write_base,
31  master_write_control_write_length,
32  master_write_user_write_buffer,
33  master_write_user_buffer_input_data,
34  master_write_user_buffer_full,
35  master_write_user_control_go,
36  master_write_user_control_done,
37
38  // read logic
39  master_read_control_fixed_location,
40  master_read_control_read_base,

```

```

41 master_read_control_read_length ,
42 master_read_user_read_buffer ,
43 master_read_user_buffer_output_data ,
44 master_read_user_data_available ,
45 master_read_user_control_go ,
46 master_read_user_control_done
47 );
48
49
50 input clk;
51 input reset;
52
53 input address;
54
55 input writeData0;
56 input writeData1;
57 input writeData2;
58 input writeData3;
59
60 input mode;
61 input modeAccessed;
62
63 // outputs
64 output reg [31:0] readData0;
65 output reg [31:0] readData1;
66 output reg [31:0] readData2;
67 output reg [31:0] readData3;
68
69 output reg writeReady;
70 output reg readReady;
71
72 // avalon connections
73 output reg master_write_control_fixed_location;
74 output reg [31:0] master_write_control_write_base;
75 output reg [31:0] master_write_control_write_length;
76 output reg master_write_user_write_buffer;
77 output reg [31:0] master_write_user_buffer_input_data;
78 input master_write_user_buffer_full;
79 output reg master_write_user_control_go;
80 input master_write_user_control_done;
81
82 output reg master_read_control_fixed_location;
83 output reg [31:0] master_read_control_read_base;
84 output reg [31:0] master_read_control_read_length;
85 output reg master_read_user_read_buffer;
86 input [31:0] master_read_user_buffer_output_data;
87 input master_read_user_data_available;
88 output reg master_read_user_control_go;
89 input master_read_user_control_done;
90
91 reg [1:0] writeState;
92 // 00 ready
93 // 01 writingToBuffer
94 // 11 writeCommand
95 // 10 waitingForFinish
96 reg [1:0] writeByteNum;
97
98 reg [1:0] readState;
99 // 00 ready/sendCommand
100 // 01 readingFromBuffer
101 // 11 waitingForFinish
102 // 10 UNUSED
103 reg [1:0] readByteNum;
104
105 reg [31:0] writeAddr;
106
107 reg [31:0] writeData0r;
108 reg [31:0] writeData1r;

```

```

109 reg [31:0] writeData2r;
110 reg [31:0] writeData3r;
111
112 initial
113 begin
114 readData0 <= 0;
115 readData1 <= 0;
116 readData2 <= 0;
117 readData3 <= 0;
118
119 writeReady <= 1;
120 readReady <= 1;
121
122 writeState <= 0;
123 writeByteNum <= 0;
124 readState <= 0;
125 readByteNum <= 0 ;
126
127 writeAddr <= 0;
128
129 writeData0r <= 0;
130 writeData1r <= 0;
131 writeData2r <= 0;
132 writeData3r <= 0;
133
134 master_write_user_buffer_input_data <= 0;
135
136 master_write_control_fixed_location <= 0;
137 master_write_control_write_base <= 0;
138 master_write_control_write_length <= 0;
139 master_write_user_write_buffer <= 0;
140 master_write_user_control_go <= 0;
141
142 master_read_control_fixed_location <= 0;
143 master_read_control_read_base <= 0;
144 master_read_control_read_length <= 0;
145 master_read_user_read_buffer <= 0;
146 master_read_user_control_go <= 0;
147 end
148
149 // write function
150 always @ (posedge clk or posedge reset)
151 begin
152 if (reset)
153 begin
154 writeReady <= 1;
155 writeState <= 0;
156 writeByteNum <= 0;
157
158 writeAddr <= 0;
159
160 writeData0r <= 0;
161 writeData1r <= 0;
162 writeData2r <= 0;
163 writeData3r <= 0;
164
165 master_write_user_buffer_input_data <= 0;
166
167 master_write_control_fixed_location <= 0;
168 master_write_control_write_base <= 0;
169 master_write_control_write_length <= 0;
170 master_write_user_write_buffer <= 0;
171 master_write_user_control_go <= 0;
172 end
173 else
174 begin
175 case (writeState)
176 2'b00: // ready

```

```

177 begin
178   if(modeAccessed && mode == 1)
179   begin
180     //begin the process of writing
181     writeReady <= 0; //no longer ready to accept new write command
182
183     writeAddr <= address;
184     writeData0r <= writeData0;
185     writeData1r <= writeData1;
186     writeData2r <= writeData2;
187     writeData3r <= writeData3;
188
189     writeState <= 2'b01;
190     writeByteNum <= 2'b00;
191   end
192 end
193
194 2'b01:
195 begin
196   //now writing bytes into fifo buffer
197   case(writeByteNum)
198     2'b00:
199     begin
200       master_write_user_buffer_input_data <= writeData0r;
201       writeByteNum <= 2'b01;
202     end
203     2'b01:
204     begin
205       master_write_user_buffer_input_data <= writeData1r;
206       writeByteNum <= 2'b11;
207     end
208     2'b11:
209     begin
210       master_write_user_buffer_input_data <= writeData2r;
211       writeByteNum <= 2'b10;
212     end
213     2'b10:
214     begin
215       master_write_user_buffer_input_data <= writeData3r;
216       writeByteNum <= 2'b00; //return to 0
217       writeState <= 2'b11;//move to next phase
218     end
219   endcase
220 end
221
222 2'b11:
223 begin
224   // send command
225   master_write_control_write_base <= writeAddr;
226   master_write_control_write_length <= 32'd16;
227   master_write_user_control_go <= 1;
228   writeState <= 2'b10;
229 end
230
231 2'b10:
232 begin
233   // send command
234   master_write_user_control_go <= 0;
235   if(master_write_user_control_done == 1)
236   begin
237     writeState <= 2'b00;//done
238     writeReady <= 1;
239   end
240 end
241 endcase
242 end
243

```

```

245 // read function
246 always @ (posedge clk)
247 begin
248   if (reset)
249     begin
250       readData0 <= 0;
251       readData1 <= 0;
252       readData2 <= 0;
253       readData3 <= 0;
254
255       readReady <= 1;
256
257       readState <= 0;
258       readByteNum <= 0;
259
260       master_read_control_fixed_location <= 0;
261       master_read_control_read_base <= 0;
262       master_read_control_read_length <= 0;
263       master_read_user_read_buffer <= 0;
264       master_read_user_control_go <= 0;
265   end
266 else
267 begin
268   case (readState)
269     2'b00:// ready
270     begin
271       master_read_user_read_buffer <= 0;
272
273       if (modeAccessed && mode == 0)
274         begin
275           // send read command
276           readReady <= 0; //no longer ready to accept new write command
277
278           master_read_control_read_base <= address;
279           master_read_control_read_length <= 32'd16;
280           master_read_user_control_go <= 1;
281
282           readState <= 2'b01;
283           readByteNum <= 2'b00;
284         end
285     end
286
287   2'b01:
288   begin
289     master_read_user_control_go <= 0;
290
291     //now writing bytes into fifo buffer
292     if (master_read_user_data_available == 1)
293       begin
294         case (readByteNum)
295           2'b00:
296             begin
297               readData0 <= master_read_user_buffer_output_data;
298               master_read_user_read_buffer <= 1;
299
300               readByteNum <= 2'b01;
301             end
302           2'b01:
303             begin
304               readData1 <= master_read_user_buffer_output_data;
305               master_read_user_read_buffer <= 1;
306
307               readByteNum <= 2'b11;
308             end
309           2'b11:
310             begin
311               readData2 <= master_read_user_buffer_output_data;
312               master_read_user_read_buffer <= 1;

```

```

313         readByteNum <= 2'b10;
314     end
315     2'b10:
316     begin
317         readData3 <= master_read_user_buffer_output_data;
318         master_read_user_read_buffer <= 1;
319
320         readByteNum <= 2'b00;
321         readState <= 2'b10;
322         end
323     endcase
324   end
325   else
326   begin
327     master_read_user_read_buffer <= 0;
328   end
329 end
330
331 2'b10:
332 begin
333 // wait for done
334   master_read_user_control_go <= 0;
335   if(master_read_user_control_done == 1)
336   begin
337     readState <= 2'b00; //done
338     readReady <= 1;
339   end
340   end
341 endcase
342 end
343 end
344
345 endmodule
346

```

Listing 16.8: Demo IP Verilog File (test_ip.v)

16.3.10 Demo IP _hw.tcl File (IPTest_hw.tcl)

```

1 # TCL File Generated by Component Editor 8.0
2 # Sat May 31 20:41:17 PDT 2008
3 # DO NOT MODIFY
4
5 #Based on the TCL Files for the Avalon Master Template and the Avalon Slave Template by
6 # Altera Corp.
7
8
9 # +-----
10 # | module burst_read_master
11 # |
12 set_module_property DESCRIPTION "ARM Mapped IP Test"
13 set_module_property NAME IPTTest
14 set_module_property VERSION 1.0
15 set_module_property GROUP "Test IP"
16 set_module_property AUTHOR cyarp
17 set_module_property ICON_PATH SCU_Logo_Seperate_Small.gif
18 set_module_property DISPLAY_NAME "Test IP"
19 set_module_property TOP_LEVEL_HDL_FILE test_ip_top.v
20 set_module_property TOP_LEVEL_HDL_MODULE test_ip_top
21 #set_module_property INSTANTIATE_IN_SYSTEM_MODULE true
22 set_module_property EDITABLE false
23 set_module_property SIMULATION_MODEL_IN_VERILOG false
24 set_module_property SIMULATION_MODEL_IN_VHDL false
25 set_module_property SIMULATION_MODEL_HAS_TULIPS false
26 set_module_property SIMULATION_MODEL_IS_OBFUSCATED false

```

```

27
28 set_module_assignment embeddedsw.dts.compatible "simple-bus"
29 #
30 # |
31 # +-----
32
33
34 set_module_property ELABORATION_CALLBACK elaborate_me
35 set_module_property VALIDATION_CALLBACK validate_me
36
37
38 # +-----
39 # | files
40 # |
41 add_file test_ip_top.v {SYNTHESIS SIMULATION}
42 add_file test_ip.v {SYNTHESIS SIMULATION}
43 add_file custom_master.v {SYNTHESIS SIMULATION}
44 add_file burst_write_master.v {SYNTHESIS SIMULATION}
45 add_file burst_read_master.v {SYNTHESIS SIMULATION}
46 add_file write_master.v {SYNTHESIS SIMULATION}
47 add_file latency_aware_read_master.v {SYNTHESIS SIMULATION}
48 add_file slave_template.v {SYNTHESIS SIMULATION}
49 add_file interrupt_logic.v {SYNTHESIS SIMULATION}
50 add_file slave_template_macros.h {SYNTHESIS SIMULATION}
51 #
52 # +-----
53
54 # +-----
55 # | parameters
56 # |
57
58 # ======BEGIN GENERAL PARMs=====
59 # Avalon Master Settings
60 add_parameter DATA_WIDTH Integer 32 "Width of the data path"
61 set_parameter_property DATA_WIDTH VISIBLE false
62 set_parameter_property DATA_WIDTH DISPLAY_NAME "Data Width"
63 set_parameter_property DATA_WIDTH GROUP "Avalon-MM Master Properties"
64 set_parameter_property DATA_WIDTH AFFECTS_PORT_WIDTHS true
65 set_parameter_property DATA_WIDTH ALLOWED_RANGES {8 16 32 64 128 256 512 1024}
66
67 add_parameter ADDRESS_WIDTH Integer "32" "Address Width"
68 set_parameter_property ADDRESS_WIDTH VISIBLE false
69 set_parameter_property ADDRESS_WIDTH DISPLAY_NAME "Address Width"
70 set_parameter_property ADDRESS_WIDTH GROUP "Avalon-MM Master Properties"
71 set_parameter_property ADDRESS_WIDTH AFFECTS_PORT_WIDTHS true
72 set_parameter_property ADDRESS_WIDTH ALLOWED_RANGES {32 31 30 29 28 27 26 25 24 23 22 21
    20 19 18 17 16 14 13 12 11 10 9 8 7 6 5 4}
73 # ======END GENERAL PARMs=====
74
75 # ======BEGIN WRITE MASTER PARMs=====
76 # Burst Settings
77 add_parameter MASTER_WRITE_BURST_CAPABLE Integer 0 "Enable bursting"
78 set_parameter_property MASTER_WRITE_BURST_CAPABLE VISIBLE true
79 set_parameter_property MASTER_WRITE_BURST_CAPABLE DISPLAY_NAME "Burst Capable"
80 set_parameter_property MASTER_WRITE_BURST_CAPABLE GROUP "Write Master - Burst Properties"
81 set_parameter_property MASTER_WRITE_BURST_CAPABLE AFFECTS_PORT_WIDTHS true
82 set_parameter_property MASTER_WRITE_BURST_CAPABLE ALLOWED_RANGES {"0:Disabled" "1:Enabled"}
    }
83
84 add_parameter MASTER_WRITE_MAXIMUMBURSTCOUNT Integer "2" "Maximum Burst Count"
85 set_parameter_property MASTER_WRITE_MAXIMUMBURSTCOUNT VISIBLE true
86 set_parameter_property MASTER_WRITE_MAXIMUMBURSTCOUNT DISPLAY_NAME "Maximum Burst Count"
87 set_parameter_property MASTER_WRITE_MAXIMUMBURSTCOUNT GROUP "Write Master - Burst
    Properties"
88 set_parameter_property MASTER_WRITE_MAXIMUMBURSTCOUNT AFFECTS_PORT_WIDTHS false
89 set_parameter_property MASTER_WRITE_MAXIMUMBURSTCOUNT ALLOWED_RANGES {1 2 4 8 16 32 64
    128}
90

```

```

91 add_parameter MASTER_WRITE_BURST_COUNT_WIDTH Integer "2" "Enable bursting"
92 set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH VISIBLE false
93 set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH DISPLAY_NAME "Burst Count Width"
94 set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH GROUP "Write Master - Burst
Properties"
95 set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH AFFECTS_PORT_WIDTHS true
96 set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH ALLOWED_RANGES {1:8}
97
98
99 # Other Settings
100 add_parameter MASTER_WRITE_FIFO_DEPTH Integer "32" "FIFO depth"
101 set_parameter_property MASTER_WRITE_FIFO_DEPTH VISIBLE true
102 set_parameter_property MASTER_WRITE_FIFO_DEPTH DISPLAY_NAME "FIFO Depth"
103 set_parameter_property MASTER_WRITE_FIFO_DEPTH GROUP "Write Master - Other Properties"
104 set_parameter_property MASTER_WRITE_FIFO_DEPTH AFFECTS_PORT_WIDTHS false
105 set_parameter_property MASTER_WRITE_FIFO_DEPTH ALLOWED_RANGES {4 8 16 32 64 128 256}
106
107 add_parameter MASTER_WRITE_FIFO_DEPTH_LOG2 Integer "5" "log2(FIFO Depth)"
108 set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 VISIBLE false
109 set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 DISPLAY_NAME "log2(FIFO Depth)"
110 set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 GROUP "Write Master - Other Properties
"
111 set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 AFFECTS_PORT_WIDTHS false
112 set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 ALLOWED_RANGES {2:8}
113
114 add_parameter MASTER_WRITE_MEMORY_BASED_FIFO Integer 1 "Select false if you want register
based (0) FIFO instead of memory (1)"
115 set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO VISIBLE true
116 set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO DISPLAY_NAME "Memory based FIFO"
117 set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO GROUP "Write Master - Other
Properties"
118 set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO AFFECTS_PORT_WIDTHS false
119 set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO ALLOWED_RANGES {"1:Memory" "0:Logic"
}
120 # =====END WRITE MASTER PARMSS=====
121
122 # =====BEGIN READ MASTER PARMSS=====
123 # Burst Settings
124 add_parameter MASTER_READ_BURST_CAPABLE Integer 0 "Enable bursting"
125 set_parameter_property MASTER_READ_BURST_CAPABLE VISIBLE true
126 set_parameter_property MASTER_READ_BURST_CAPABLE DISPLAY_NAME "Burst Capable"
127 set_parameter_property MASTER_READ_BURST_CAPABLE GROUP "Read Master - Burst Properties"
128 set_parameter_property MASTER_READ_BURST_CAPABLE AFFECTS_PORT_WIDTHS true
129 set_parameter_property MASTER_READ_BURST_CAPABLE ALLOWED_RANGES {"0:Disabled" "1:Enabled"}
130
131 add_parameter MASTER_READ_MAXIMUM_BURST_COUNT Integer "2" "Maximum Burst Count"
132 set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT VISIBLE true
133 set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT DISPLAY_NAME "Maximum Burst Count"
134 set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT GROUP "Read Master - Burst
Properties"
135 set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT AFFECTS_PORT_WIDTHS false
136 set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT ALLOWED_RANGES {1 2 4 8 16 32 64
128}
137
138 add_parameter MASTER_READ_BURST_COUNT_WIDTH Integer "2" "Enable bursting"
139 set_parameter_property MASTER_READ_BURST_COUNT_WIDTH VISIBLE false
140 set_parameter_property MASTER_READ_BURST_COUNT_WIDTH DISPLAY_NAME "Burst Count Width"
141 set_parameter_property MASTER_READ_BURST_COUNT_WIDTH GROUP "Read Master - Burst Properties
"
142 set_parameter_property MASTER_READ_BURST_COUNT_WIDTH AFFECTS_PORT_WIDTHS true
143 set_parameter_property MASTER_READ_BURST_COUNT_WIDTH ALLOWED_RANGES {1:8}
144
145
146 # Other Settings
147 add_parameter MASTER_READ_FIFO_DEPTH Integer "32" "FIFO depth"
148 set_parameter_property MASTER_READ_FIFO_DEPTH VISIBLE true
149 set_parameter_property MASTER_READ_FIFO_DEPTH DISPLAY_NAME "FIFO Depth"
150 set_parameter_property MASTER_READ_FIFO_DEPTH GROUP "Read Master - Other Properties"

```

```

151 set_parameter_property MASTER_READ_FIFO_DEPTH AFFECTS_PORT_WIDTHS false
152 set_parameter_property MASTER_READ_FIFO_DEPTH ALLOWED_RANGES {4 8 16 32 64 128 256}
153
154 add_parameter MASTER_READ_FIFO_DEPTH_LOG2 Integer "5" "log2(FIFO Depth)"
155 set_parameter_property MASTER_READ_FIFO_DEPTH_LOG2 VISIBLE false
156 set_parameter_property MASTER_READ_FIFO_DEPTH_LOG2 DISPLAY_NAME "log2(FIFO Depth)"
157 set_parameter_property MASTER_READ_FIFO_DEPTH_LOG2 GROUP "Read Master - Other Properties"
158 set_parameter_property MASTER_READ_FIFO_DEPTH_LOG2 AFFECTS_PORT_WIDTHS false
159 set_parameter_property MASTER_READ_FIFO_DEPTH_LOG2 ALLOWED_RANGES {2:8}
160
161 add_parameter MASTER_READ_MEMORY_BASED_FIFO Integer 1 "Select false if you want register
162   based (0) FIFO instead of memory (1)"
163 set_parameter_property MASTER_READ_MEMORY_BASED_FIFO VISIBLE true
164 set_parameter_property MASTER_READ_MEMORY_BASED_FIFO DISPLAY_NAME "Memory based FIFO"
165 set_parameter_property MASTER_READ_MEMORY_BASED_FIFO GROUP "Read Master - Other Properties
166   "
167 set_parameter_property MASTER_READ_MEMORY_BASED_FIFO AFFECTS_PORT_WIDTHS false
168 set_parameter_property MASTER_READ_MEMORY_BASED_FIFO ALLOWED_RANGES {"1:Memory" "0:Logic"}
169 # =====END READ MASTER PARMSS=====
170
171 # =====BEGIN SLAVE PARMS=====
172 add_parameter IRQ_EN int 0 "Enable or disable the interrupt capabilities of input ports"
173 set_parameter_property IRQ_EN DISPLAY_NAME "Interrupt capabilities"
174 set_parameter_property IRQ_EN GROUP "CSR - Interrupt"
175 set_parameter_property IRQ_EN AFFECTS_PORT_WIDTHS true
176 set_parameter_property IRQ_EN ALLOWED_RANGES {"0:Disabled" "1:Enabled"}
177 # =====END SLAVE PARMS=====
178
179 # |
180 # +-----
181 # | connection point clk_in
182 # |
183 add_interface clk_in clock end
184 set_interface_property clk_in ptfSchematicName ""
185
186 add_interface_port clk_in clk clk Input 1
187 add_interface_port clk_in reset reset Input 1
188 # |
189 # +-----
190
191 # |
192 # | connection point master_write
193 # |
194 add_interface master_write avalon start
195 set_interface_property master_write linewrapBursts false
196 set_interface_property master_write adaptsTo ""
197 set_interface_property master_write doStreamReads false
198 set_interface_property master_write doStreamWrites false
199 set_interface_property master_write burstOnBurstBoundariesOnly false
200
201 set_interface_property master_write ASSOCIATED_CLOCK clk_in
202
203 add_interface_port master_write master_write_address address Output -1
204 #add_interface_port master_write master_write_read read Output 1
205 add_interface_port master_write master_write_write write Output 1
206 add_interface_port master_write master_write_bytewable bytewable Output -1
207 #add_interface_port master_write master_write_readdata readdata Input -1
208 #add_interface_port master_write master_write_readdatavalid readdatavalid Input 1
209 add_interface_port master_write master_write_writedata writedata Output -1
210 add_interface_port master_write master_write_burstcount burstcount Output -1
211 add_interface_port master_write master_write_waitrequest waitrequest Input 1
212 # |
213 # +-----
214
215 # |
216 # | connection point master_read

```

```

217 # |
218 add_interface master_read avalon start
219 set_interface_property master_read linewrapBursts false
220 set_interface_property master_read adaptsTo ""
221 set_interface_property master_read doStreamReads false
222 set_interface_property master_read doStreamWrites false
223 set_interface_property master_read burstOnBurstBoundariesOnly false
224
225 set_interface_property master_read ASSOCIATED_CLOCK clk_in
226
227 add_interface_port master_read master_read_address address Output -1
228 add_interface_port master_read master_read_read read Output 1
229 #add_interface_port master_read master_read_write write Output 1
230 add_interface_port master_read master_read_bytelenable byteenable Output -1
231 add_interface_port master_read master_read_readdata readdata Input -1
232 add_interface_port master_read master_read_readdatavalid readdatavalid Input 1
233 #add_interface_port master_read master_read_writedata writedata Output -1
234 add_interface_port master_read master_read_burstcount burstcount Output -1
235 add_interface_port master_read master_read_waitrequest waitrequest Input 1
236 #
237 # +-----
238
239 # +-----
240 # | connection point csr
241 #
242 add_interface csr avalon end
243 set_interface_property csr holdTime 0
244 set_interface_property csr linewrapBursts false
245 set_interface_property csr minimumUninterruptedRunLength 1
246 set_interface_property csr bridgesToMaster ""
247 set_interface_property csr isMemoryDevice false
248 set_interface_property csr burstOnBurstBoundariesOnly false
249 set_interface_property csr addressSpan 512
250 set_interface_property csr timingUnits Cycles
251 set_interface_property csr setupTime 0
252 set_interface_property csr writeWaitTime 0
253 set_interface_property csr isNonVolatileStorage false
254 set_interface_property csr addressAlignment DYNAMIC
255 set_interface_property csr maximumPendingReadTransactions 0
256 set_interface_property csr readWaitTime 0
257 set_interface_property csr readLatency 3
258 set_interface_property csr printableDevice false
259
260 set_interface_property csr ASSOCIATED_CLOCK clk_in
261
262 add_interface_port csr slave_address address Input 9
263 add_interface_port csr slave_read read Input 1
264 add_interface_port csr slave_write write Input 1
265 add_interface_port csr slave_readdata readdata Output -1
266 add_interface_port csr slave_writedata writedata Input -1
267
268 # +-----
269 # | connection point csr_irq
270 #
271 add_interface csr_irq interrupt end
272 set_interface_property csr_irq associatedAddressablePoint csr
273
274 set_interface_property csr_irq ASSOCIATED_CLOCK clk_in
275
276 add_interface_port csr_irq slave_irq irq Output 1
277 #
278 #
279 # +-----
280
281 proc elaborate_me {} {
282
283 # set all the new port widths - this is all based
284 set the_data_width [get_parameter_value DATA_WIDTH]

```

```

285 set the_address_width [get_parameter_value ADDRESS.WIDTH]
286 set the_byteenable_width [expr {$the_data_width / 8} ]
287
288 # ======BEGIN MASTER WRITE ELABORATE=====
289 set master_write_the_burst_count_width [get_parameter_value
290     MASTER_WRITE_BURST_COUNT_WIDTH]
291
292 set_port_property master_write_address WIDTH $the_address_width
293 set_port_property master_write_byteenable WIDTH $the_byteenable_width
294 #set_port_property master_write_readdata WIDTH $the_data_width
295 set_port_property master_write_writedata WIDTH $the_data_width
296 set_port_property master_write_burstcount WIDTH $master_write_the_burst_count_width
297
298 # determine the master direction and burst capabilities
299 # write master
300 set master_write_the_burst_capable [get_parameter_value MASTER_WRITE_BURST_CAPABLE]
301
302
303 # switch between read and write master signals (excluding burstcount)
304 #set_port_property master_write_read TERMINATION true
305 set_port_property master_write_write TERMINATION false
306 #set_port_property master_write_readdata TERMINATION true
307 #set_port_property master_write_readdatavalid TERMINATION true
308 set_port_property master_write_writedata TERMINATION false
309
310 # enable/disable the burstcount signal
311 if { $master_write_the_burst_capable == 0 } {
312     set_port_property master_write_burstcount TERMINATION true
313 } else {
314     set_port_property master_write_burstcount TERMINATION false
315 }
316 # ======END MASTER WRITE ELABORATE=====
317
318 # ======BEGIN MASTER READ ELABORATE=====
319 set master_read_the_burst_count_width [get_parameter_value MASTER_READ_BURST_COUNT_WIDTH]
320
321 set_port_property master_read_address WIDTH $the_address_width
322 set_port_property master_read_byteenable WIDTH $the_byteenable_width
323 set_port_property master_read_readdata WIDTH $the_data_width
324 #set_port_property master_read_writedata WIDTH $the_data_width
325 set_port_property master_read_burstcount WIDTH $master_read_the_burst_count_width
326
327
328 # determine the master direction and burst capabilities
329 # write master
330 set master_read_the_burst_capable [get_parameter_value MASTER_READ_BURST_CAPABLE]
331
332 # switch between read and write master signals (excluding burstcount)
333 set_port_property master_read_read TERMINATION false
334 #set_port_property master_read_write TERMINATION true
335 set_port_property master_read_readdata TERMINATION false
336 set_port_property master_read_readdatavalid TERMINATION false
337 #set_port_property master_read_writedata TERMINATION true
338
339 # enable/disable the burstcount signal
340 if { $master_read_the_burst_capable == 0 } {
341     set_port_property master_read_burstcount TERMINATION true
342 } else {
343     set_port_property master_read_burstcount TERMINATION false
344 }
345 # ======END MASTER READ ELABORATE=====
346
347 # ======BEGIN SLAVE ELABORATE=====
348 set_port_property slave_readdata WIDTH $the_data_width
349 set_port_property slave_writedata WIDTH $the_data_width
350

```

```

351 set the_irq_en [get_parameter_value IRQ_EN]
352
353     ## adding the slave_byteenable and user_byteenable signals only if the data width is
354     ## greater than 8 bits
355 if { $the_data_width != 8 } {
356     add_interface_port csr slave_byteenable byteenable Input [expr {$the_data_width / 8} ]
357 #if { $the_enable_sync_signals == 1 } {
#    add_interface_port user_interface user_byteenable export Output [expr {
#        $the_data_width / 8} ]
358 #}
359 }
360
361     ## terminate interrupt signal if it is not turned on
362 expr {($the_irq_en == 1) ? [set_port_property slave_irq TERMINATION false] : [
363     set_port_property slave_irq TERMINATION true]};
# =====END SLAVE ELABORATE=====
364 }
365
366
367 proc validate_me {} {
368
369     ## =====BEGIN MASTER WRITE VALIDATE=====
370     # read in all the parameter that matter for validation
371     set master_write_the_burst_capable [get_parameter_value MASTER_WRITE_BURST_CAPABLE]
372     set master_write_the_maximum_burst_count [get_parameter_value
373         MASTER_WRITE_MAXIMUMBURSTCOUNT]
374     set master_write_the_fifo_depth [get_parameter_value MASTER_WRITE_FIFO_DEPTH]
375
376     # when burst is enabled check to make sure FIFO depth is at least twice as large (also
377     # enable/disable burst count)
378     if { $master_write_the_burst_capable == 1 } {
379         set_parameter_property MASTER_WRITE_MAXIMUMBURSTCOUNT ENABLED true
380         if { $master_write_the_fifo_depth < [expr {$master_write_the_maximum_burst_count * 2}] } {
381             send_message Error "The FIFO Depth must be at least twice as large as Maximum Burst
382             Count."
383         }
384     } else {
385         set_parameter_property MASTER_WRITE_MAXIMUMBURSTCOUNT ENABLED false
386     }
387
388     set master_write_the_burst_count [get_parameter_value MASTER_WRITE_MAXIMUMBURSTCOUNT]
389     set master_write_the_burst_count_width [expr {($log($master_write_the_burst_count)) / log
390         (2)) + 1}]
391
392     set master_write_the_fifo_depth [get_parameter_value MASTER_WRITE_FIFO_DEPTH]
393     set master_write_the_fifo_depth_log2 [expr {log($master_write_the_fifo_depth) / log(2)}]
394     set_parameter_value MASTER_WRITE_BURST_COUNT_WIDTH $master_write_the_burst_count_width
395     set_parameter_value MASTER_WRITE_FIFO_DEPTH_LOG2 $master_write_the_fifo_depth_log2
# =====END MASTER WRITE VALIDATE=====

396     ## =====BEGIN MASTER READ VALIDATE=====
397     # read in all the parameter that matter for validation
398     set master_read_the_burst_capable [get_parameter_value MASTER_READ_BURST_CAPABLE]
399     set master_read_the_maximum_burst_count [get_parameter_value
400         MASTER_READ_MAXIMUMBURSTCOUNT]
401     set master_read_the_fifo_depth [get_parameter_value MASTER_READ_FIFO_DEPTH]
402
403     # when burst is enabled check to make sure FIFO depth is at least twice as large (also
404     # enable/disable burst count)
405     if { $master_read_the_burst_capable == 1 } {
406         set_parameter_property MASTER_READ_MAXIMUMBURSTCOUNT ENABLED true
407         if { $master_read_the_fifo_depth < [expr {$master_read_the_maximum_burst_count * 2}] } {
408             send_message Error "The FIFO Depth must be at least twice as large as Maximum Burst
409             Count."
410         }
411     }

```

```

408 } else {
409     set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT ENABLED false
410 }
411
412
413 set master_read_the_burst_count [get_parameter_value MASTER_READ_MAXIMUM_BURST_COUNT]
414 set master_read_the_burst_count_width [expr {($master_read_the_burst_count) / log(2)
415 ) + 1}]
416
417 set master_read_the_fifo_depth [get_parameter_value MASTER_READ_FIFO_DEPTH]
418 set master_read_the_fifo_depth_log2 [expr {log($master_read_the_fifo_depth) / log(2)}]
419
420 set_parameter_value MASTER_READ_BURST_COUNT_WIDTH $master_read_the_burst_count_width
421 set_parameter_value MASTER_READ_FIFO_DEPTH_LOG2 $master_read_the_fifo_depth_log2
422 # =====END MASTER READ VALIDATE=====
423
424 }

```

Listing 16.9: Demo IP .hw.tcl File (IPTest_hw.tcl)

16.3.11 Avalon-MM Custom Master - Altera Template (custom_master.v)

```

1 /*
2 This file is a simple top level that will generate one of four types of
3 Avalon-MM master. As a result all the ports must be declared and it will be
4 up to the component .tcl file to stub unused signals.
5 */
6
7 // altera_message_off 10034
8
9 module custom_master (
10     clk,
11     reset,
12
13     // control inputs and outputs
14     control_fixed_location,
15     control_read_base,
16     control_read_length,
17     control_write_base,
18     control_write_length,
19     control_go,
20     control_done,
21     control_early_done,
22
23     // user logic inputs and outputs
24     user_read_buffer,
25     user_write_buffer,
26     user_buffer_input_data,
27     user_buffer_output_data,
28     user_data_available,
29     user_buffer_full,
30
31     // master inputs and outputs
32     master_address,
33     master_read,
34     master_write,
35     master_bytelenable,
36     master_readdata,
37     master_readdatavalid,
38     master_writedata,
39     master_burstcount,
40     master_waitrequest
41 );
42
43 parameter MASTER_DIRECTION = 0;           // 0 for read master, 1 for write master
44 parameter DATA_WIDTH = 32;

```

```

45 parameter MEMORY_BASED_FIFO = 1;           // 0 for LE/ALUT FIFOs, 1 for memory FIFOs (
46   highly recommend 1)
47 parameter FIFO_DEPTH = 32;
48 parameter FIFO_DEPTH_LOG2 = 5;
49 parameter ADDRESS_WIDTH = 32;
50 parameter BURST_CAPABLE = 0;                // 1 to enable burst, 0 to disable it
51 parameter MAXIMUM_BURST_COUNT = 2;
52 parameter BURST_COUNT_WIDTH = 2;

53
54 input clk;
55 input reset;

56
57 // control inputs and outputs
58 input control_fixed_location;
59 input [ADDRESS_WIDTH-1:0] control_read_base;    // for read master
60 input [ADDRESS_WIDTH-1:0] control_read_length;   // for read master
61 input [ADDRESS_WIDTH-1:0] control_write_base;    // for write master
62 input [ADDRESS_WIDTH-1:0] control_write_length;  // for write master
63 input control_go;
64 output wire control_done;
65 output wire control_early_done;               // for read master

66
67 // user logic inputs and outputs
68 input user_read_buffer;                     // for read master
69 input user_write_buffer;                    // for write master
70 input [DATA_WIDTH-1:0] user_buffer_input_data; // for write master
71 output wire [DATA_WIDTH-1:0] user_buffer_output_data; // for read master
72 output wire user_data_available;           // for read master
73 output wire user_buffer_full;              // for write master

74
75 // master inputs and outputs
76 output wire [ADDRESS_WIDTH-1:0] master_address;
77 output wire master_read;                   // for read master
78 output wire master_write;                 // for write master
79 output wire [(DATA_WIDTH/8)-1:0] master_byteenable;
80 input [DATA_WIDTH-1:0] master_readdata;     // for read master
81 input master_readdatavalid;               // for read master
82 output wire [DATA_WIDTH-1:0] master_writedata; // for write master
83 output wire [BURST_COUNT_WIDTH-1:0] master_burstcount; // for bursting read and write
84   masters
85   input master_waitrequest;

86
87 generate // big generate if statement to select the appropriate master depending on the
88   direction and burst parameters
89 if(MASTER_DIRECTION == 0)
90 begin
91   if(BURST_CAPABLE == 1)
92     begin
93       burst_read_master a_burst_read_master(
94         .clk (clk),
95         .reset (reset),
96         .control_fixed_location (control_fixed_location),
97         .control_read_base (control_read_base),
98         .control_read_length (control_read_length),
99         .control_go (control_go),
100        .control_done (control_done),
101        .control_early_done (control_early_done),
102        .user_read_buffer (user_read_buffer),
103        .user_buffer_data (user_buffer_output_data),
104        .user_data_available (user_data_available),
105        .master_address (master_address),
106        .master_read (master_read),
107        .master_byteenable (master_byteenable),
108        .master_readdata (master_readdata),
109        .master_readdatavalid (master_readdatavalid),
110        .master_burstcount (master_burstcount),

```

```

110      .master_waitrequest (master_waitrequest)
111  );
112  defparam a_burst_read_master.DATAWIDTH = DATA.WIDTH;
113  defparam a_burst_read_master.MAXBURSTCOUNT = MAXIMUM_BURST.COUNT;
114  defparam a_burst_read_master.BURSTCOUNTWIDTH = BURST_COUNT.WIDTH;
115  defparam a_burst_read_master.BYTEENABLEWIDTH = DATA.WIDTH/8;
116  defparam a_burst_read_master.ADDRESSWIDTH = ADDRESS.WIDTH;
117  defparam a_burst_read_master.FIFODEPTH = FIFO_DEPTH;
118  defparam a_burst_read_master.FIFODEPTH_LOG2 = FIFO_DEPTH.LOG2;
119  defparam a_burst_read_master.FIFOUSEMEMORY = MEMORY_BASED_FIFO;
120 end
121 else
122 begin
123   latency_aware_read_master a_latency_aware_read_master(
124     .clk (clk),
125     .reset (reset),
126     .control_fixed_location (control_fixed_location),
127     .control_read_base (control_read_base),
128     .control_read_length (control_read_length),
129     .control_go (control_go),
130     .control_done (control_done),
131     .control_early_done (control_early_done),
132     .user_read_buffer (user_read_buffer),
133     .user_buffer_data (user_buffer_output_data),
134     .user_data_available (user_data_available),
135     .master_address (master_address),
136     .master_read (master_read),
137     .master_byteenable (master_byteenable),
138     .master_readdata (master_readdata),
139     .master_readdatavalid (master_readdatavalid),
140     .master_waitrequest (master_waitrequest)
141   );
142   defparam a_latency_aware_read_master.DATAWIDTH = DATA.WIDTH;
143   defparam a_latency_aware_read_master.BYTEENABLEWIDTH = DATA.WIDTH/8;
144   defparam a_latency_aware_read_master.ADDRESSWIDTH = ADDRESS.WIDTH;
145   defparam a_latency_aware_read_master.FIFODEPTH = FIFO_DEPTH;
146   defparam a_latency_aware_read_master.FIFODEPTH_LOG2 = FIFO_DEPTH.LOG2;
147   defparam a_latency_aware_read_master.FIFOUSEMEMORY = MEMORY_BASED_FIFO;
148 end
149 end
150 else
151 begin
152   if(BURST.CAPABLE == 1)
153 begin
154   burst_write_master a_burst_write_master(
155     .clk (clk),
156     .reset (reset),
157     .control_fixed_location (control_fixed_location),
158     .control_write_base (control_write_base),
159     .control_write_length (control_write_length),
160     .control_go (control_go),
161     .control_done (control_done),
162     .user_write_buffer (user_write_buffer),
163     .user_buffer_data (user_buffer_input_data),
164     .user_buffer_full (user_buffer_full),
165     .master_address (master_address),
166     .master_write (master_write),
167     .master_byteenable (master_byteenable),
168     .master_writedata (master_writedata),
169     .master_burstcount (master_burstcount),
170     .master_waitrequest (master_waitrequest)
171   );
172   defparam a_burst_write_master.DATAWIDTH = DATA.WIDTH;
173   defparam a_burst_write_master.MAXBURSTCOUNT = MAXIMUM_BURST.COUNT;
174   defparam a_burst_write_master.BURSTCOUNTWIDTH = BURST_COUNT.WIDTH;
175   defparam a_burst_write_master.BYTEENABLEWIDTH = DATA.WIDTH/8;
176   defparam a_burst_write_master.ADDRESSWIDTH = ADDRESS.WIDTH;
177   defparam a_burst_write_master.FIFODEPTH = FIFO_DEPTH;

```

```

178 defparam a_burst_write_master.FIFODEPTH.LOG2 = FIFO_DEPTH.LOG2;
179 defparam a_burst_write_master.FIFOUSEMEMORY = MEMORY_BASED_FIFO;
180 end
181 else
182 begin
183     write_master a_write_master(
184         .clk (clk),
185         .reset (reset),
186         .control_fixed_location (control_fixed_location),
187         .control_write_base (control_write_base),
188         .control_write_length (control_write_length),
189         .control_go (control_go),
190         .control_done (control_done),
191         .user_write_buffer (user_write_buffer),
192         .user_buffer_data (user_buffer_input_data),
193         .user_buffer_full (user_buffer_full),
194         .master_address (master_address),
195         .master_write (master_write),
196         .master_byteenable (master_byteenable),
197         .master_writedata (master_writedata),
198         .master_waitrequest (master_waitrequest)
199 );
200     defparam a_write_master.DATAWIDTH = DATA.WIDTH;
201     defparam a_write_master.BYTEENABLEWIDTH = DATA.WIDTH/8;
202     defparam a_write_master.ADDRESSWIDTH = ADDRESS.WIDTH;
203     defparam a_write_master.FIFODEPTH = FIFO_DEPTH;
204     defparam a_write_master.FIFODEPTH.LOG2 = FIFO_DEPTH.LOG2;
205     defparam a_write_master.FIFOUSEMEMORY = MEMORY_BASED_FIFO;
206 end
207 end
208 endgenerate
209
210
211 endmodule

```

Listing 16.10: Avalon-MM Custom Master - Altera Template (custom_master.v)

16.3.12 Avalon-MM Burst Read Master - Altera Template (burst_read_master.v)

```

1 /*
2  Legal Notice: (C)2007 Altera Corporation. All rights reserved. Your
3  use of Altera Corporation's design tools, logic functions and other
4  software and tools, and its AMPP partner logic functions, and any
5  output files any of the foregoing (including device programming or
6  simulation files), and any associated documentation or information are
7  expressly subject to the terms and conditions of the Altera Program
8  License Subscription Agreement or other applicable license agreement,
9  including , without limitation, that your use is for the sole purpose
10 of programming logic devices manufactured by Altera and sold by Altera
11 or its authorized distributors. Please refer to the applicable
12 agreement for further details.
13 */
14
15 /*
16 Author: JCJB
17 Date: 11/04/2007
18
19 This bursting read master is passed a word aligned address, length in bytes,
20 and a 'go' bit. The master will continue to post full length bursts until
21 the length register reaches a value less than a full burst. A single final
22 burst is then posted and when all the reads return the done bit will be asserted.
23
24 To use this master you must simply drive the control signals into this block,
25 and also read the data from the exposed read FIFO. To read from the exposed FIFO
use the 'user_read_buffer' signal to pop data from the FIFO 'user_buffer_data'.
27

```

```

28     The signal 'user_data_available' is asserted whenever data is available from the
29     exposed FIFO.
30
31 */
32
33 // altera_message_off 10230
34
35 module burst_read_master (
36     clk ,
37     reset ,
38
39     // control inputs and outputs
40     control_fixed_location ,
41     control_read_base ,
42     control_read_length ,
43     control_go ,
44     control_done ,
45     control_early_done ,
46
47     // user logic inputs and outputs
48     user_read_buffer ,
49     user_buffer_data ,
50     user_data_available ,
51
52     // master inputs and outputs
53     master_address ,
54     master_read ,
55     master_bytelenable ,
56     master_readdata ,
57     master_readdatavalid ,
58     master_burstcount ,
59     master_waitrequest
60 );
61
62 parameter DATAWIDTH = 32;
63 parameter MAXBURSTCOUNT = 4;
64 parameter BURSTCOUNTWIDTH = 3;
65 parameter BYTENABLEWIDTH = 4;
66 parameter ADDRESSWIDTH = 32;
67 parameter FIFODEPTH = 32;
68 parameter FIFODEPTHLOG2 = 5;
69 parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead
70
71 input clk;
72 input reset;
73
74
75 // control inputs and outputs
76 input control_fixed_location;
77 input [ADDRESSWIDTH-1:0] control_read_base;
78 input [ADDRESSWIDTH-1:0] control_read_length;
79 input control_go;
80 output wire control_done;
81 output wire control_early_done; // don't use this unless you know what you are doing,
82 // it's going to fire when the last read is posted, not when the last data returns!
83
84 // user logic inputs and outputs
85 input user_read_buffer;
86 output wire [DATAWIDTH-1:0] user_buffer_data;
87 output wire user_data_available;
88
89 // master inputs and outputs
90 input master_waitrequest;
91 input master_readdatavalid;
92     input [DATAWIDTH-1:0] master_readdata;
93     output wire [ADDRESSWIDTH-1:0] master_address;
94     output wire master_read;

```

```

95    output wire [BYTEENABLEWIDTH-1:0] master_byteenable;
96    output wire [BURSTCOUNTWIDTH-1:0] master_burstcount;
97
98    // internal control signals
99    reg control_fixed_location_d1;
100   wire fifo_empty;
101   reg [ADDRESSWIDTH-1:0] address;
102   reg [ADDRESSWIDTH-1:0] length;
103   reg [FIFODEPTH_LOG2-1:0] reads_pending;
104   wire increment_address;
105   wire [BURSTCOUNTWIDTH-1:0] burst_count;
106   wire [BURSTCOUNTWIDTH-1:0] first_short_burst_count;
107   wire first_short_burst_enable;
108   wire [BURSTCOUNTWIDTH-1:0] final_short_burst_count;
109   wire final_short_burst_enable;
110   wire [BURSTCOUNTWIDTH-1:0] burst_boundary_word_address;
111   reg burst_begin;
112   wire too_many_reads_pending;
113   wire [FIFODEPTH_LOG2-1:0] fifo_used;
114
115
116
117 // registering the control_fixed_location bit
118 always @ (posedge clk or posedge reset)
119 begin
120   if (reset == 1)
121     begin
122       control_fixed_location_d1 <= 0;
123     end
124   else
125     begin
126       if (control_go == 1)
127         begin
128           control_fixed_location_d1 <= control_fixed_location;
129         end
130     end
131 end
132
133
134
135 // master address logic
136 always @ (posedge clk or posedge reset)
137 begin
138   if (reset == 1)
139     begin
140       address <= 0;
141     end
142   else
143     begin
144       if (control_go == 1)
145         begin
146           address <= control_read_base;
147         end
148       else if ((increment_address == 1) & (control_fixed_location_d1 == 0))
149         begin
150           address <= address + (burst_count * BYTEENABLEWIDTH); // always performing word
151           size accesses , increment by the burst count presented
152         end
153     end
154
155
156
157 // master length logic
158 always @ (posedge clk or posedge reset)
159 begin
160   if (reset == 1)
161     begin

```

```

162      length <= 0;
163    end
164  else
165  begin
166    if(control_go == 1)
167    begin
168      length <= control_read_length;
169    end
170    else if(increment_address == 1)
171    begin
172      length <= length - (burst_count * BYTENABLEWIDTH); // always performing word
173      size accesses, decrement by the burst count presented
174    end
175  end
176
177
178
179 // controlled signals going to the master/control ports
180 assign master_address = address;
181 assign master_bytetable = -1; // all ones, always performing word size accesses
182 assign master_burstcount = burst_count;
183 assign control_done = (length == 0) & (reads_pending == 0); // need to make sure that
184     the reads have returned before firing the done bit
185 assign control_early_done = (length == 0); // advanced feature, you should use ,
186     control_done' if you need all the reads to return first
187 assign master_read = (too_many_reads_pending == 0) & (length != 0);
188 assign burst_boundary_word_address = ((address / BYTENABLEWIDTH) & (MAXBURSTCOUNT - 1))
189 ;
190 assign first_short_burst_enable = (burst_boundary_word_address != 0);
191 assign final_short_burst_enable = (length < (MAXBURSTCOUNT * BYTENABLEWIDTH));
192
193 assign first_short_burst_count = ((burst_boundary_word_address & 1'b1) == 1'b1)? 1 : // if
194     the burst boundary isn't a multiple of 2 then must post a burst of 1 to get to a
195     multiple of 2 for the next burst
196     (((MAXBURSTCOUNT - burst_boundary_word_address) < (length /
197         BYTENABLEWIDTH))?
198         (MAXBURSTCOUNT - burst_boundary_word_address) : (length /
199             BYTENABLEWIDTH));
200 assign final_short_burst_count = (length / BYTENABLEWIDTH);
201
202 assign burst_count = (first_short_burst_enable == 1)? first_short_burst_count : // this
203     will get the transfer back on a burst boundary,
204     (final_short_burst_enable == 1)? final_short_burst_count : MAXBURSTCOUNT;
205 assign increment_address = (too_many_reads_pending == 0) & (master_waitrequest == 0) &
206     (length != 0);
207 assign too_many_reads_pending = (reads_pending + fifo_used) >= (FIFODEPTH -
208     MAXBURSTCOUNT - 4); // make sure there are fewer reads posted than room in the FIFO
209
210
211 // tracking FIFO
212 always @ (posedge clk or posedge reset)
213 begin
214   if (reset == 1)
215   begin
216     reads_pending <= 0;
217   end
218   else
219   begin
220     if(increment_address == 1)
221     begin
222       if(master_readdatavalid == 0)
223       begin
224         reads_pending <= reads_pending + burst_count;
225       end
226       else
227       begin

```

```

218     reads_pending <= reads_pending + burst_count - 1; // a burst read was posted ,
219     but a word returned
220 end
221 else
222 begin
223     if(master_readdatavalid == 0)
224     begin
225         reads_pending <= reads_pending; // burst read was not posted and no read
226         returned
227     end
228     else
229     begin
230         reads_pending <= reads_pending - 1; // burst read was not posted but a word
231         returned
232     end
233 end
234
235
236 // read data feeding user logic
237 assign user_data_available = !fifo_empty;
238 scfifo the_master_to_user_fifo (
239     .aclr (reset),
240     .clock (clk),
241     .data (master_readdata),
242     .empty (fifo_empty),
243     .q (user_buffer.data),
244     .rdreq (user_read_buffer),
245     .usedw (fifo_used),
246     .wrreq (master_readdatavalid)
247 );
248 defparam the_master_to_user_fifo.lpm_width = DATAWIDTH;
249 defparam the_master_to_user_fifo.lpm_numwords = FIFODEPTH;
250 defparam the_master_to_user_fifo.lpm_showahead = "ON";
251 defparam the_master_to_user_fifo.use_eab = (FIFOUSEMEMORY == 1)? "ON" : "OFF";
252 defparam the_master_to_user_fifo.add_ram_output_register = "OFF";
253 defparam the_master_to_user_fifo.underflow_checking = "OFF";
254 defparam the_master_to_user_fifo.overflow_checking = "OFF";
255
256 endmodule

```

Listing 16.11: Avalon-MM Burst Read Master - Altera Template (burst_read_master.v)

16.3.13 Avalon-MM Latency Aware Read Master - Altera Template (latency_aware_read_master.v)

```

1 /*
2  Legal Notice: (C)2007 Altera Corporation. All rights reserved. Your
3  use of Altera Corporation's design tools, logic functions and other
4  software and tools, and its AMPP partner logic functions, and any
5  output files any of the foregoing (including device programming or
6  simulation files), and any associated documentation or information are
7  expressly subject to the terms and conditions of the Altera Program
8  License Subscription Agreement or other applicable license agreement,
9  including, without limitation, that your use is for the sole purpose
10 of programming logic devices manufactured by Altera and sold by Altera
11 or its authorized distributors. Please refer to the applicable
12 agreement for further details.
13 */
14 /*
15  Author: JCJB
16  Date: 11/04/2007
17
18
19

```

```

20 This latency aware read master is passed a word aligned address, length in bytes,
21 and a 'go' bit. The master will continue to post reads until the length register
22 reaches a value of zero. When all the reads return the done bit will be asserted.
23
24 To use this master you must simply drive the control signals into this block,
25 and also read the data from the exposed read FIFO. To read from the exposed FIFO
26 use the 'user_read_buffer' signal to pop data from the FIFO 'user_buffer_data'.
27 The signal 'user_data_available' is asserted whenever data is available from the
28 exposed FIFO.
29
30 */
31 // altera_message_off 10230
32
33
34 module latency_aware_read_master (
35   clk ,
36   reset ,
37
38   // control inputs and outputs
39   control_fixed_location ,
40   control_read_base ,
41   control_read_length ,
42   control_go ,
43   control_done ,
44   control_early_done ,
45
46   // user logic inputs and outputs
47   user_read_buffer ,
48   user_buffer_data ,
49   user_data_available ,
50
51   // master inputs and outputs
52   master_address ,
53   master_read ,
54   master_bytelenable ,
55   master_readdata ,
56   master_readdatavalid ,
57   master_waitrequest
58 );
59
60 parameter DATAWIDTH = 32;
61 parameter BYTENABLEWIDTH = 4;
62 parameter ADDRESSWIDTH = 32;
63 parameter FIFODEPTH = 32;
64 parameter FIFODEPTH_LOG2 = 5;
65 parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead
66
67 input clk;
68 input reset;
69
70
71 // control inputs and outputs
72 input control_fixed_location;
73 input [ADDRESSWIDTH-1:0] control_read_base;
74 input [ADDRESSWIDTH-1:0] control_read_length;
75 input control_go;
76 output wire control_done;
77 output wire control_early_done; // don't use this unless you know what you are doing!
78
79 // user logic inputs and outputs
80 input user_read_buffer;
81 output wire [DATAWIDTH-1:0] user_buffer_data;
82 output wire user_data_available;
83
84 // master inputs and outputs
85 input master_waitrequest;
86 input master_readdatavalid;
87

```

```

88     input [DATAWIDTH-1:0] master_readdata;
89     output wire [ADDRESSWIDTH-1:0] master_address;
90     output wire master_read;
91     output wire [BYTEENABLEWIDTH-1:0] master_byteenable;
92
93 // internal control signals
94 reg control_fixed_location_d1;
95 wire fifo_empty;
96 reg [ADDRESSWIDTH-1:0] address;
97 reg [ADDRESSWIDTH-1:0] length;
98 reg [FIFODEPTH_LOG2-1:0] reads_pending;
99 wire increment_address;
100 wire too_many_pending_reads;
101 reg too_many_pending_reads_d1;
102 wire [FIFODEPTH_LOG2-1:0] fifo_used;
103
104
105
106
107 // registering the control_fixed_location bit
108 always @ (posedge clk or posedge reset)
109 begin
110     if (reset == 1)
111         begin
112             control_fixed_location_d1 <= 0;
113         end
114     else
115         begin
116             if (control_go == 1)
117                 begin
118                     control_fixed_location_d1 <= control_fixed_location;
119                 end
120             end
121         end
122
123
124
125 // master address logic
126 assign master_address = address;
127 assign master_byteenable = -1; // all ones, always performing word size accesses
128 always @ (posedge clk or posedge reset)
129 begin
130     if (reset == 1)
131         begin
132             address <= 0;
133         end
134     else
135         begin
136             if (control_go == 1)
137                 begin
138                     address <= control_read_base;
139                 end
140             else if ((increment_address == 1) & (control_fixed_location_d1 == 0))
141                 begin
142                     address <= address + BYTEENABLEWIDTH; // always performing word size accesses
143                 end
144             end
145         end
146
147
148
149 // master length logic
150 always @ (posedge clk or posedge reset)
151 begin
152     if (reset == 1)
153         begin
154             length <= 0;
155         end

```

```

156     else
157     begin
158         if(control_go == 1)
159         begin
160             length <= control_read_length;
161         end
162         else if(increment_address == 1)
163         begin
164             length <= length - BYTEENABLEWIDTH; // always performing word size accesses
165         end
166     end
167
168
169
170 // control logic
171 assign too_many_pending_reads = (fifo_used + reads_pending) >= (FIFODEPTH - 4);
172 assign master_read = (length != 0) & (too_many_pending_reads_d1 == 0);
173 assign increment_address = (length != 0) & (too_many_pending_reads_d1 == 0) &
174     (master_waitrequest == 0);
175 assign control_done = (reads_pending == 0) & (length == 0); // master done posting
176     reads and all reads have returned
177 assign control_early_done = (length == 0); // if you need all the pending reads to
178     return then use 'control_done' instead of this signal
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220

```

```

221      end
222    end
223  end
224
225
226
227 // read data feeding user logic
228 assign user_data_available = !fifo_empty;
229 scfifo the_master_to_user_fifo (
230   .aclr (reset),
231   .clock (clk),
232   .data (master_readdata),
233   .empty (fifo_empty),
234   .q (user_buffer_data),
235   .rdreq (user_read_buffer),
236   .usedw (fifo_used),
237   .wrreq (master_readdatavalid)
238 );
239 defparam the_master_to_user_fifo.lpm_width = DATAWIDTH;
240 defparam the_master_to_user_fifo.lpm_numwords = FIFODEPTH;
241 defparam the_master_to_user_fifo.lpm_showahead = "ON";
242 defparam the_master_to_user_fifo.use_eab = (FIFOUSEMEMORY == 1) ? "ON" : "OFF";
243 defparam the_master_to_user_fifo.add_ram_output_register = "OFF";
244 defparam the_master_to_user_fifo.underflow_checking = "OFF";
245 defparam the_master_to_user_fifo.overflow_checking = "OFF";
246
247 endmodule

```

Listing 16.12: Avalon-MM Latency Aware Read Master - Altera Template (latency_aware_read_master.v)

16.3.14 Avalon-MM Write Master - Altera Template (write_master.v)

```

1 /*
2  Legal Notice: (C)2007 Altera Corporation. All rights reserved. Your
3  use of Altera Corporation's design tools, logic functions and other
4  software and tools, and its AMPP partner logic functions, and any
5  output files any of the foregoing (including device programming or
6  simulation files), and any associated documentation or information are
7  expressly subject to the terms and conditions of the Altera Program
8  License Subscription Agreement or other applicable license agreement,
9  including, without limitation, that your use is for the sole purpose
10 of programming logic devices manufactured by Altera and sold by Altera
11 or its authorized distributors. Please refer to the applicable
12 agreement for further details.
13 */
14
15 /*
16
17 Author: JCJB
18 Date: 11/04/2007
19
20 This simple write master is passed a word aligned address, length in bytes,
21 and a 'go' bit. The master will continue to post writes until the length register
22 reaches zero. When the length register reaches zero the 'done' bit is asserted.
23
24 To use this master you must simply drive the control signals into this block,
25 and also write the data to the exposed write FIFO. To read from the exposed FIFO
26 use the 'user_write_buffer' signal to push data into the FIFO 'user_buffer_data'.
27 The signal 'user_buffer_full' is asserted whenever the exposed buffer is full.
28 You should not attempt to write data to the exposed FIFO if it is full.
29
30 */
31
32
33 // altera_message_off 10230
34

```

```

35 module write_master (
36   clk ,
37   reset ,
38
39   // control inputs and outputs
40   control_fixed_location ,
41   control_write_base ,
42   control_write_length ,
43   control_go ,
44   control_done ,
45
46   // user logic inputs and outputs
47   user_write_buffer ,
48   user_buffer_data ,
49   user_buffer_full ,
50
51   // master inputs and outputs
52   master_address ,
53   master_write ,
54   master_bytelenable ,
55   master_writedata ,
56   master_waitrequest
57 );
58
59
60 parameter DATAWIDTH = 32;
61 parameter BYTENABLEWIDTH = 4;
62 parameter ADDRESSWIDTH = 32;
63 parameter FIFODEPTH = 32;
64 parameter FIFODEPTH_LOG2 = 5;
65 parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead
66
67
68
69   input  clk;
70   input  reset;
71
72   // control inputs and outputs
73   input  control_fixed_location; // this only makes sense to enable when MAXBURSTCOUNT = 1
74   input [ADDRESSWIDTH-1:0] control_write_base;
75   input [ADDRESSWIDTH-1:0] control_write_length;
76   input  control_go;
77   output wire control_done;
78
79   // user logic inputs and outputs
80   input  user_write_buffer;
81   input [DATAWIDTH-1:0] user_buffer_data;
82   output wire user_buffer_full;
83
84   // master inputs and outputs
85   input  master_waitrequest;
86   output wire [ADDRESSWIDTH-1:0] master_address;
87   output wire master_write;
88   output wire [BYTENABLEWIDTH-1:0] master_bytelenable;
89   output wire [DATAWIDTH-1:0] master_writedata;
90
91
92   // internal control signals
93   reg  control_fixed_location_d1 ;
94   reg [ADDRESSWIDTH-1:0] address; // this increments for each word
95   reg [ADDRESSWIDTH-1:0] length;
96   wire increment_address; // this increments the 'address' register when write is
97     asserted and waitrequest is de-asserted
98   wire read_fifo;
99   wire user_buffer_empty;
100
101

```

```

102 // registering the control_fixed_location bit
103 always @ (posedge clk or posedge reset)
104 begin
105   if (reset == 1)
106     begin
107       control_fixed_location_d1 <= 0;
108     end
109   else
110     begin
111       if (control_go == 1)
112         begin
113           control_fixed_location_d1 <= control_fixed_location;
114         end
115     end
116   end
117
118
119
120 // master word increment counter
121 always @ (posedge clk or posedge reset)
122 begin
123   if (reset == 1)
124     begin
125       address <= 0;
126     end
127   else
128     begin
129       if (control_go == 1)
130         begin
131           address <= control_write_base;
132         end
133       else if ((increment_address == 1) & (control_fixed_location_d1 == 0))
134         begin
135           address <= address + BYTEENABLEWIDTH; // always performing word size accesses
136         end
137     end
138   end
139
140
141 // master length logic
142 always @ (posedge clk or posedge reset)
143 begin
144   if (reset == 1)
145     begin
146       length <= 0;
147     end
148   else
149     begin
150       if (control_go == 1)
151         begin
152           length <= control_write_length;
153         end
154       else if (increment_address == 1)
155         begin
156           length <= length - BYTEENABLEWIDTH; // always performing word size accesses
157         end
158     end
159   end
160
161
162
163 // controlled signals going to the master/control ports
164 assign master_address = address;
165 assign master_byteenable = -1; // all ones, always performing word size accesses
166 assign control_done = (length == 0);
167 assign master_write = (user_buffer_empty == 0) & (control_done == 0);
168

```

```

169 assign increment_address = (user_buffer_empty == 0) & (master_waitrequest == 0) & (
170   control_done == 0);
171 assign read_fifo = increment_address;
172
173 // write data feed by user logic
174 scfifo the_user_to_master_fifo (
175   .aclr (reset),
176   .clock (clk),
177   .data (user_buffer_data),
178   .full (user_buffer_full),
179   .empty (user_buffer_empty),
180   .q (master_writedata),
181   .rdreq (read_fifo),
182   .wrreq (user_write_buffer)
183 );
184 defparam the_user_to_master_fifo.lpm_width = DATAWIDTH;
185 defparam the_user_to_master_fifo.lpm_numwords = FIFODEPTH;
186 defparam the_user_to_master_fifo.lpm_showahead = "ON";
187 defparam the_user_to_master_fifo.use_eab = (FIFOUSEMEMORY == 1)? "ON" : "OFF";
188 defparam the_user_to_master_fifo.add_ram_output_register = "OFF";
189 defparam the_user_to_master_fifo.underflow_checking = "OFF";
190 defparam the_user_to_master_fifo.overflow_checking = "OFF";
191
192 endmodule

```

Listing 16.13: Avalon-MM Write Master (write_master.v)

16.3.15 Avalon-MM Burst Write Master (burst_write_master.v)

```

1 /*
2  Legal Notice: (C)2007 Altera Corporation. All rights reserved. Your
3  use of Altera Corporation's design tools, logic functions and other
4  software and tools, and its AMPP partner logic functions, and any
5  output files any of the foregoing (including device programming or
6  simulation files), and any associated documentation or information are
7  expressly subject to the terms and conditions of the Altera Program
8  License Subscription Agreement or other applicable license agreement,
9  including, without limitation, that your use is for the sole purpose
10 of programming logic devices manufactured by Altera and sold by Altera
11 or its authorized distributors. Please refer to the applicable
12 agreement for further details.
13 */
14 /*
15
16 Author: JCJB
17 Date: 11/04/2007
18
19 This bursting write master is passed a word aligned address, length in bytes,
20 and a 'go' bit. The master will continue to post full length bursts until
21 the length register reaches a value less than a full burst. A single final
22 burst is then posted when enough data has been buffered and then the done bit
23 will be asserted.
24
25 To use this master you must simply drive the control signals into this block,
26 and also write the data to the exposed write FIFO. To read from the exposed FIFO
27 use the 'user_write_buffer' signal to push data into the FIFO 'user_buffer_data'.
28 The signal 'user_buffer_full' is asserted whenever the exposed buffer is full.
29 You should not attempt to write data to the exposed FIFO if it is full.
30
31 */
32 /*
33 // altera_message_off 10230
34
35
36 module burst_write_master (
37

```

```

38    clk ,
39    reset ,
40
41    // control inputs and outputs
42    control_fixed_location ,
43    control_write_base ,
44    control_write_length ,
45    control_go ,
46    control_done ,
47
48    // user logic inputs and outputs
49    user_write_buffer ,
50    user_buffer_data ,
51    user_buffer_full ,
52
53    // master inputs and outputs
54    master_address ,
55    master_write ,
56    master_bytelenable ,
57    master_writedata ,
58    master_burstcount ,
59    master_waitrequest
60 );
61
62
63 parameter DATAWIDTH = 32;
64 parameter MAXBURSTCOUNT = 4;
65 parameter BURSTCOUNTWIDTH = 3;
66 parameter BYTENABLEWIDTH = 4;
67 parameter ADDRESSWIDTH = 32;
68 parameter FIFODEPTH = 32; // must be at least twice MAXBURSTCOUNT in order to be
69   efficient
70 parameter FIFODEPTH_LOG2 = 5;
71 parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead
72
73
74 input clk;
75 input reset;
76
77 // control inputs and outputs
78 input control_fixed_location; // this only makes sense to enable when MAXBURSTCOUNT = 1
79 input [ADDRESSWIDTH-1:0] control_write_base;
80 input [ADDRESSWIDTH-1:0] control_write_length;
81 input control_go;
82 output wire control_done;
83
84 // user logic inputs and outputs
85 input user_write_buffer;
86 input [DATAWIDTH-1:0] user_buffer_data;
87 output wire user_buffer_full;
88
89 // master inputs and outputs
90 input master_waitrequest;
91 output reg [ADDRESSWIDTH-1:0] master_address;
92 output wire master_write;
93 output wire [BYTENABLEWIDTH-1:0] master_bytelenable;
94 output wire [DATAWIDTH-1:0] master_writedata;
95 output reg [BURSTCOUNTWIDTH-1:0] master_burstcount;
96
97 // internal control signals
98 reg control_fixed_location_d1;
99 reg [ADDRESSWIDTH-1:0] length;
100 wire final_short_burst_enable; // when the length is less than MAXBURSTCOUNT * # of
101   bytes per word (BYTENABLEWIDTH) (i.e. end of the transfer)
102 wire final_short_burst_ready; // when there is enough data in the FIFO for the final
103   burst

```

```

102    wire [BURSTCOUNTWIDTH-1:0] burst_boundary_word_address; // represents the word offset
103    within the burst boundary
104    wire [BURSTCOUNTWIDTH-1:0] first_short_burst_count;
105    wire [BURSTCOUNTWIDTH-1:0] final_short_burst_count;
106    wire first_short_burst_enable; // when the transfer doesn't start on a burst boundary
107    wire first_short_burst_ready; // when there is enough data in the FIFO to get the
108    master back into burst alignment
109    wire full_burst_ready; // when there is enough data in the FIFO for a full burst
110    wire increment_address; // this increments the 'address' register when write is
111    asserted and waitrequest is de-asserted
112    wire burst_begin; // used to register the registers 'burst_address' and 'burst_count_d1'
113    ' as well as drive the master_address and burst_count muxes
114    wire read_fifo;
115
116
117    // registering the control_fixed_location bit
118    always @ (posedge clk or posedge reset)
119    begin
120        if (reset == 1)
121            begin
122                control_fixed_location_d1 <= 0;
123            end
124        else
125            begin
126                if (control_go == 1)
127                    begin
128                        control_fixed_location_d1 <= control_fixed_location;
129                    end
130            end
131    end
132
133
134    // set when control_go fires, and reset once the first burst starts
135    always @ (posedge clk or posedge reset)
136    begin
137        if (reset == 1)
138            begin
139                first_transfer <= 0;
140            end
141        else
142            begin
143                if (control_go == 1)
144                    begin
145                        first_transfer <= 1;
146                    end
147                else if (burst_begin == 1)
148                    begin
149                        first_transfer <= 0;
150                    end
151            end
152    end
153
154
155    // master address (held constant during burst)
156    always @ (posedge clk or posedge reset)
157    begin
158        if (reset == 1)
159            begin
160                master_address <= 0;
161            end
162        else

```

```

164 begin
165   if (control_go == 1)
166     begin
167       master_address <= control_write_base;
168     end
169   else if ((first_transfer == 0) & (burst_begin == 1) & (control_fixed_location_d1 ==
170             0))
171     begin
172       master_address <= master_address + (master_burstcount * BYTEENABLEWIDTH); // we
173         don't want address + BYTEENABLEWIDTH for the first access
174     end
175   end
176
177 // master length logic
178 always @ (posedge clk or posedge reset)
179 begin
180   if (reset == 1)
181     begin
182       length <= 0;
183     end
184   else
185     begin
186       if (control_go == 1)
187         begin
188           length <= control_write_length;
189         end
190       else if (increment_address == 1)
191         begin
192           length <= length - BYTEENABLEWIDTH; // always performing word size accesses
193         end
194       end
195     end
196
197
198 // register the master burstcount (held constant during burst)
199 always @ (posedge clk or posedge reset)
200 begin
201   if (reset == 1)
202     begin
203       master_burstcount <= 0;
204     end
205   else
206     begin
207       if (burst_begin == 1)
208         begin
209           master_burstcount <= burst_count;
210         end
211     end
212   end
213
214
215
216 // burst counter. This is set to the burst count being posted then counts down when
217 // each word
218 // of data goes out. If it reaches 0 (i.e. not reloaded after 1) then the master stalls
219 // due to
220 // a lack of data to post a new burst.
221 always @ (posedge clk or posedge reset)
222 begin
223   if (reset == 1)
224     begin
225       burst_counter <= 0;
226     end
227   else
228     begin

```

```

228     if (control_go == 1)
229     begin
230         burst_counter <= 0;
231     end
232     else if (burst_begin == 1)
233     begin
234         burst_counter <= burst_count;
235     end
236     else if (increment_address == 1) // decrements each write, burst_counter will only
237         hit 0 if burst_begin doesn't fire on the next cycle
238     begin
239         burst_counter <= burst_counter - 1;
240     end
241 end
242
243
244
245
246 // burst boundaries are on the master "width * maximum burst count". The burst boundary
247 // word address will be used to determine how far off the boundary the transfer starts
248 // from.
249 assign burst_boundary_word_address = ((master_address / BYTEENABLEWIDTH) & (
250     MAXBURSTCOUNT - 1));
251
252 // first short burst enable will only be active on the first transfer (if applicable).
253 // It will either post the amount of words remaining to reach the end of the burst
254 // boundary or post the remainder of the transfer whichever is shorter. If the transfer
255 // is very short and not aligned on a burst boundary then the same logic as the final
256 // short transfer is used
257 assign first_short_burst_enable = (burst_boundary_word_address != 0) & (first_transfer
258     == 1);
259 assign first_short_burst_count = ((burst_boundary_word_address & 1'b1) == 1'b1)? 1 : // 
260     if the burst boundary isn't a multiple of 2 then must post a burst of 1 to get to a
261     multiple of 2 for the next burst
262     (((MAXBURSTCOUNT - burst_boundary_word_address) < (length /
263         BYTEENABLEWIDTH))?
264         (MAXBURSTCOUNT - burst_boundary_word_address) : final_short_burst_count)
265     ;
266 assign first_short_burst_ready = (fifo_used > first_short_burst_count) | ((fifo_used ==
267     first_short_burst_count) & (burst_counter == 0));
268
269 // when there isn't enough data for a full burst at the end of the transfer a short
270 // burst is sent out instead
271 assign final_short_burst_enable = (length < (MAXBURSTCOUNT * BYTEENABLEWIDTH));
272 assign final_short_burst_count = (length/BYTEENABLEWIDTH);
273 assign final_short_burst_ready = (fifo_used > final_short_burst_count) | ((fifo_used ==
274     final_short_burst_count) & (burst_counter == 0)); // this will add a one cycle
275     stall between bursts, since fifo_used has a cycle of latency, this only affects the
276     last burst
277
278 // since the fifo has a latency of 1 we need to make sure we don't under flow
279 assign full_burst_ready = (fifo_used > MAXBURSTCOUNT) | ((fifo_used == MAXBURSTCOUNT) &
280     (burst_counter == 0)); // when fifo used watermark equals the burst count the
281     statemachine must stall for one cycle, this will make sure that when a burst begins
282     there really is enough data present in the FIFO
283
284
285
286 assign master_byteenable = -1; // all ones, always performing word size accesses
287 assign control_done = (length == 0);
288 assign master_write = (control_done == 0) & (burst_counter != 0); // burst_counter = 0
289     means the transfer is done, or not enough data in the fifo for a new burst
290
291 // fifo controls and the burst_begin responsible for timing most of the circuit,
292 // burst_begin starts the writing statemachine
293 assign burst_begin = (((first_short_burst_enable == 1) & (first_short_burst_ready == 1))
294     | ((final_short_burst_enable == 1) & (final_short_burst_ready == 1))
295     | (full_burst_ready == 1))

```

```

274     & (control_done == 0) // since the FIFO can have data before the master
275     starts we need to disable this bit from firing when length = 0
276     & ((burst_counter == 0) | ((burst_counter == 1) & (master_waitrequest == 0) &
277     (length > (MAXBURSTCOUNT * BYTEENABLEWIDTH)))); // need to make a short
278     final burst doesn't start right after a full burst completes.
279
280     assign burst_count = (first_short_burst_enable == 1)? first_short_burst_count : // alignment correction gets priority, if the transfer is short and unaligned this will
281     cover both
282     (final_short_burst_enable == 1)? final_short_burst_count : MAXBURSTCOUNT;
283
284
285     assign increment_address = (master_write == 1) & (master_waitrequest == 0); // writing
286     is occurring without wait states
287     assign read_fifo = increment_address;
288
289
290     // write data feed by user logic
291     scfifo the_user_to_master_fifo (
292     .aclr (reset),
293     .usedw (fifo_used),
294     .clock (clk),
295     .data (user_buffer_data),
296     .almost_full (user_buffer_full),
297     .q (master_writedata),
298     .rdreq (read_fifo),
299     .wrreq (user_write_buffer)
300   );
301   defparam the_user_to_master_fifo.lpm_width = DATAWIDTH;
302   defparam the_user_to_master_fifo.lpm_numwords = FIFODEPTH;
303   defparam the_user_to_master_fifo.lpm_showahead = "ON";
304   defparam the_user_to_master_fifo.almost_full_value = (FIFODEPTH - 2);
305   defparam the_user_to_master_fifo.use_eab = (FIFOUSEMEMORY == 1)? "ON" : "OFF";
306   defparam the_user_to_master_fifo.add_ram_output_register = "OFF"; // makes timing the
307   burst begin single simplifier
308   defparam the_user_to_master_fifo.underflow_checking = "OFF";
309   defparam the_user_to_master_fifo.overflow_checking = "OFF";
310
311 endmodule

```

Listing 16.14: Avalon-MM Burst Write Master - Altera Template (burst_write_master.v)

16.3.16 Avalon-MM Interrupt Logic - Altera Template (interrupt_logic.v)

```

1 module interrupt_logic (
2   clk ,
3   reset ,
4   data_in ,
5   write ,
6   write_data ,
7   address_decode ,
8   irq_mask_reg_en ,
9   edge_capture_reg_en ,
10  read_data ,
11  irq_out
12 );
13
14 parameter DATA_WIDTH = 32;
15
16 input clk ;
17 input reset ;
18 input [DATA_WIDTH-1:0] data_in ;
19 input write ;
20 input [DATA_WIDTH-1:0] write_data ;
21 input address_decode ;
22 input irq_mask_reg_en ;

```

```

23| input edge_capture_reg_en;
24| output reg [DATA_WIDTH-1:0] read_data;
25| output wire irq_out;
26|
27| // internal logic
28| reg [DATA_WIDTH-1:0] data_in_d1;
29| reg [DATA_WIDTH-1:0] data_in_d2;
30| reg [DATA_WIDTH-1:0] data_in_d3;
31| wire [DATA_WIDTH-1:0] edge_detect;
32| reg [DATA_WIDTH-1:0] edge_capture;
33| reg [DATA_WIDTH-1:0] irq_mask;
34| wire edge_capture_wr_strobe;
35| wire irq_mask_wr_strobe;
36| wire [DATA_WIDTH-1:0] readdata_mux;
37|
38|
39| // interrupt mask register
40| always @ (posedge clk or posedge reset)
41| begin
42|   if (reset == 1)
43|     begin
44|       irq_mask <= 0;
45|     end
46|   else if (irq_mask_wr_strobe)
47|     begin
48|       irq_mask <= write_data;
49|     end
50| end
51|
52| // double registers for asynchronous input and assure metastability
53| always @ (posedge clk or posedge reset)
54| begin
55|   if (reset == 1)
56|     begin
57|       data_in_d1 <= 0;
58|       data_in_d2 <= 0;
59|     end
60|   else
61|     begin
62|       data_in_d1 <= data_in;
63|       data_in_d2 <= data_in_d1;
64|     end
65| end
66|
67| // edge detection logic
68| always @ (posedge clk or posedge reset)
69| begin
70|   if (reset == 1)
71|     begin
72|       data_in_d3 <= 0;
73|     end
74|   else
75|     begin
76|       data_in_d3 <= data_in_d2;
77|     end
78| end
79|
80| assign edge_detect = data_in_d2 & ~data_in_d3;
81|
82| // edge capture registers with separate clear bit
83| generate
84| genvar BIT;
85| for(BIT = 0; BIT < DATA_WIDTH; BIT = BIT + 1)
86| begin: edge_capture_generation
87|   always @ (posedge clk or posedge reset)
88|   begin
89|     if (reset == 1)
90|       begin

```

```

91      edge_capture[BIT] <= 0;
92    end
93  else
94  begin
95    if (edge_capture_wr_strobe && write_data[BIT])
96      edge_capture[BIT] <= 0;
97    else if (edge_detect[BIT])
98      edge_capture[BIT] <= 1;
99    end
100  end
101end
102endgenerate
103
104// register the readdata_mux
105always @ (posedge clk or posedge reset)
106begin
107  if (reset == 1)
108    begin
109      read_data <= 0;
110    end
111  else
112    begin
113      read_data <= readdata_mux;
114    end
115end
116
117
118assign readdata_mux = ({DATA_WIDTH{{irq_mask_reg_en == 1'b1}}} & irq_mask) | ({DATA_WIDTH{{edge_capture_reg_en == 1'b1}}} & edge_capture);
119assign irq_mask_wr_strobe = (write == 1'b1) && (address_decode == 1'b1) && (
120  irq_mask_reg_en == 1'b1);
121assign edge_capture_wr_strobe = (write == 1'b1) && (address_decode == 1'b1) && (
122  edge_capture_reg_en == 1'b1);
123assign irq_out = |(edge_capture & irq_mask);
124
125endmodule

```

Listing 16.15: Avalon-MM Interrupt Logic - Altera Template (interrupt_logic.v)

16.3.17 Avalon-MM Slave Template - Altera Template (slave_template.v)

```

1 /*
2  Legal Notice: (C)2009 Altera Corporation. All rights reserved. Your
3  use of Altera Corporation's design tools, logic functions and other
4  software and tools, and its AMPP partner logic functions, and any
5  output files any of the foregoing (including device programming or
6  simulation files), and any associated documentation or information are
7  expressly subject to the terms and conditions of the Altera Program
8  License Subscription Agreement or other applicable license agreement,
9  including, without limitation, that your use is for the sole purpose
10 of programming logic devices manufactured by Altera and sold by Altera
11 or its authorized distributors. Please refer to the applicable
12 agreement for further details.
13 */
14
15 /*
16 This slave component has a parameterizable data width and 16 input/output
17 words. There are five modes for each addressable word in this component
18 as follows:
19
20 Mode = 0 --> Output only
21 Mode = 1 --> Input only
22 Mode = 2 --> Output and input (independent I/O, default)
23 Mode = 3 --> Output with loopback (software readable output registers)
24 Mode = 4 --> Disabled
25

```

```

26
27 This component is always available so the waitrequest signal is not used
28 and it has fixed read and write latencies. The write latency is 0 and the
29 read latency is 3 cycles. If you attempt to access a location that doesn't
30 support the necessary functionality then you will either write to a
31 non-existent register (write to space) or will readback 0 as the inputs
32 will be grounded if they are disabled. Inputs or outputs that are removed
33 will be due to the component tcl file stubbing the signals. Disabled outputs
34 will not be exposed at the top of the system and the Quartus II software
35 will optimize the register away. Disabled inputs (except in the loopback
36 mode) will not be exposed at the top and internally be wired to ground.
37 The Quartus II software as a result will optimize the input registers to
38 be hardcoded wires set to ground automatically.
39
40 In order for your external logic to know which register is being accessed
41 you will need to enable 'ENABLE_SYNC_SIGNALS' by setting it to 1. When
42 enabled, the user_chipselect/byteenable/read/write signals will be exposed to your
43 external logic which you can use to determine which register is being accessed
44 and whether it's a read or write access.
45
46 If you use the synchronization signals use the following to qualify them:
47
48 Read: user_chipselect[x] AND user_read
49 Write: user_chipselect[x] AND user_write AND user_byteenable
50
51 Note: Reads return the full word regardless of the byteenables presented.
52 */
53
54
55
56
57 module slave_template (
58     // signals to connect to an Avalon clock source interface
59     clk,
60     reset,
61
62     // signals to connect to an Avalon-MM slave interface
63     slave_address,
64     slave_read,
65     slave_write,
66     slave_readdata,
67     slave_writedata,
68     slave_byteenable,
69
70     // interrupt signals
71     slave_irq,
72
73     // signals to connect to custom user logic (up to 16 input and output pairs)
74     user_dataout_0,
75     user_dataout_1,
76     user_dataout_2,
77     user_dataout_3,
78     user_dataout_4,
79     user_dataout_5,
80     user_dataout_6,
81     user_dataout_7,
82     user_dataout_8,
83     user_dataout_9,
84     user_dataout_10,
85     user_dataout_11,
86     user_dataout_12,
87     user_dataout_13,
88     user_dataout_14,
89     user_dataout_15,
90     user_datain_0,
91     user_datain_1,
92     user_datain_2,
93     user_datain_3,

```

```

94    user_datain_4 ,
95    user_datain_5 ,
96    user_datain_6 ,
97    user_datain_7 ,
98    user_datain_8 ,
99    user_datain_9 ,
100   user_datain_10 ,
101   user_datain_11 ,
102   user_datain_12 ,
103   user_datain_13 ,
104   user_datain_14 ,
105   user_datain_15 ,
106
107  // optional signals so that your external logic knows what location is being accessed
108  user_chipselect ,
109  user_byteenable ,
110  user_write ,
111  user_read
112 );
113
114 // most of the set values will only be used by the component .tcl file. The DATA_WIDTH
115 // and MODEX = 3 influence the hardware created.
116 // ENABLE SYNC SIGNALS isn't used by this hardware at all but it provided anyway so that
117 // it can be exposed in the component .tcl file
118 // to control the stubbing of certain signals.
119 parameter DATA_WIDTH = 32;           // word size of each input and output register
120 parameter ENABLE_SYNC_SIGNALS = 0;  // only used by the component .tcl file, 1 to expose
121 // user_chipselect/write/read, 0 to stub them
122 parameter MODE_0 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
123 // Output with loopback, 4 = Disabled
124 parameter MODE_1 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
125 // Output with loopback, 4 = Disabled
126 parameter MODE_2 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
127 // Output with loopback, 4 = Disabled
128 parameter MODE_3 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
129 // Output with loopback, 4 = Disabled
130 parameter MODE_4 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
131 // Output with loopback, 4 = Disabled
132 parameter MODE_5 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
133 // Output with loopback, 4 = Disabled
134 parameter MODE_6 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
135 // Output with loopback, 4 = Disabled
136 parameter MODE_7 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
137 // Output with loopback, 4 = Disabled
138 parameter MODE_8 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
139 // Output with loopback, 4 = Disabled
140 parameter MODE_9 = 2;              // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
141 // Output with loopback, 4 = Disabled
142 parameter MODE_10 = 2;             // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
143 // Output with loopback, 4 = Disabled
144 parameter MODE_11 = 2;             // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
145 // Output with loopback, 4 = Disabled
146 parameter MODE_12 = 2;             // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
147 // Output with loopback, 4 = Disabled
148 parameter MODE_13 = 2;             // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
149 // Output with loopback, 4 = Disabled
150 parameter MODE_14 = 2;             // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
151 // Output with loopback, 4 = Disabled
152 parameter MODE_15 = 2;             // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
153 // Output with loopback, 4 = Disabled
154 parameter IRQ_EN = 0;              // 0 = Enable interrupt, 1 = Disable interrupt
155
156
157
158 // clock interface
159 input clk;
160 input reset;

```

```

143 // slave interface
144 input [8:0] slave_address;
145 input slave_read;
146 input slave_write;
147 output wire [DATA_WIDTH-1:0] slave_readdata;
148 input [DATA_WIDTH-1:0] slave_writedata;
149 input [(DATA_WIDTH/8)-1:0] slave_byteenable;
150 output wire slave_irq;
151
152
153 // user interface
154 output wire [DATA_WIDTH-1:0] user_dataout_0;
155 output wire [DATA_WIDTH-1:0] user_dataout_1;
156 output wire [DATA_WIDTH-1:0] user_dataout_2;
157 output wire [DATA_WIDTH-1:0] user_dataout_3;
158 output wire [DATA_WIDTH-1:0] user_dataout_4;
159 output wire [DATA_WIDTH-1:0] user_dataout_5;
160 output wire [DATA_WIDTH-1:0] user_dataout_6;
161 output wire [DATA_WIDTH-1:0] user_dataout_7;
162 output wire [DATA_WIDTH-1:0] user_dataout_8;
163 output wire [DATA_WIDTH-1:0] user_dataout_9;
164 output wire [DATA_WIDTH-1:0] user_dataout_10;
165 output wire [DATA_WIDTH-1:0] user_dataout_11;
166 output wire [DATA_WIDTH-1:0] user_dataout_12;
167 output wire [DATA_WIDTH-1:0] user_dataout_13;
168 output wire [DATA_WIDTH-1:0] user_dataout_14;
169 output wire [DATA_WIDTH-1:0] user_dataout_15;
170
171 input [DATA_WIDTH-1:0] user_datain_0;
172 input [DATA_WIDTH-1:0] user_datain_1;
173 input [DATA_WIDTH-1:0] user_datain_2;
174 input [DATA_WIDTH-1:0] user_datain_3;
175 input [DATA_WIDTH-1:0] user_datain_4;
176 input [DATA_WIDTH-1:0] user_datain_5;
177 input [DATA_WIDTH-1:0] user_datain_6;
178 input [DATA_WIDTH-1:0] user_datain_7;
179 input [DATA_WIDTH-1:0] user_datain_8;
180 input [DATA_WIDTH-1:0] user_datain_9;
181 input [DATA_WIDTH-1:0] user_datain_10;
182 input [DATA_WIDTH-1:0] user_datain_11;
183 input [DATA_WIDTH-1:0] user_datain_12;
184 input [DATA_WIDTH-1:0] user_datain_13;
185 input [DATA_WIDTH-1:0] user_datain_14;
186 input [DATA_WIDTH-1:0] user_datain_15;
187 output wire [15:0] user_chipselect;
188 output wire [(DATA_WIDTH/8)-1:0] user_byteenable;
189 output wire user_write;
190 output wire user_read;
191
192
193 // internal logic signals
194 wire [(DATA_WIDTH/8)-1:0] internal_byteenable; // when DATA_WIDTH is 8 bits need to
195 // hardcode this signal to 1 since the fabric doesn't support 1 bit byte enables
196 reg [(DATA_WIDTH/8)-1:0] internal_byteenable_d1;
197 wire [15:0] address_decode;
198 reg [15:0] address_decode_d1; // used to select the first stage of mux
199 // pipelining (a, b, c, or d)
200 wire [3:0] address_bank_decode; // used to select the second stage of mux
201 // pipelining (mux a, b, c, or d)
202 reg [3:0] address_bank_decode_d1; // used to select the second stage of mux
203 // pipelining (mux a, b, c, or d)
204 reg slave_read_d1; // used to qualify the first stage of mux
205 // pipelining (a, b, c, or d)
206 reg slave_read_d2; // used to qualify the second stage of mux
207 // pipelining (slave_readdata)
208 reg slave_write_d1; // used by the option user_write signal
209 reg [DATA_WIDTH-1:0] user_datain_0_d1;
210 reg [DATA_WIDTH-1:0] user_datain_1_d1;

```

```

205 reg [DATA_WIDTH-1:0] user_datain_2_d1;
206 reg [DATA_WIDTH-1:0] user_datain_3_d1;
207 reg [DATA_WIDTH-1:0] user_datain_4_d1;
208 reg [DATA_WIDTH-1:0] user_datain_5_d1;
209 reg [DATA_WIDTH-1:0] user_datain_6_d1;
210 reg [DATA_WIDTH-1:0] user_datain_7_d1;
211 reg [DATA_WIDTH-1:0] user_datain_8_d1;
212 reg [DATA_WIDTH-1:0] user_datain_9_d1;
213 reg [DATA_WIDTH-1:0] user_datain_10_d1;
214 reg [DATA_WIDTH-1:0] user_datain_11_d1;
215 reg [DATA_WIDTH-1:0] user_datain_12_d1;
216 reg [DATA_WIDTH-1:0] user_datain_13_d1;
217 reg [DATA_WIDTH-1:0] user_datain_14_d1;
218 reg [DATA_WIDTH-1:0] user_datain_15_d1;
219 reg [DATA_WIDTH-1:0] mux_first_stage_a; // muxed inputs 0-3
220 reg [DATA_WIDTH-1:0] mux_first_stage_b; // muxed inputs 4-7
221 reg [DATA_WIDTH-1:0] mux_first_stage_c; // muxed inputs 8-11
222 reg [DATA_WIDTH-1:0] mux_first_stage_d; // muxed inputs 12-15
223 reg [DATA_WIDTH-1:0] int_mux_first_stage_a; // muxed inputs 0-3
224 reg [DATA_WIDTH-1:0] int_mux_first_stage_b; // muxed inputs 4-7
225 reg [DATA_WIDTH-1:0] int_mux_first_stage_c; // muxed inputs 8-11
226 reg [DATA_WIDTH-1:0] int_mux_first_stage_d; // muxed inputs 12-15
227 reg [DATA_WIDTH-1:0] read_from_user;
228 reg [DATA_WIDTH-1:0] read_from_int_regs;
229 wire [DATA_WIDTH-1:0] readdata_0;
230 wire [DATA_WIDTH-1:0] readdata_1;
231 wire [DATA_WIDTH-1:0] readdata_2;
232 wire [DATA_WIDTH-1:0] readdata_3;
233 wire [DATA_WIDTH-1:0] readdata_4;
234 wire [DATA_WIDTH-1:0] readdata_5;
235 wire [DATA_WIDTH-1:0] readdata_6;
236 wire [DATA_WIDTH-1:0] readdata_7;
237 wire [DATA_WIDTH-1:0] readdata_8;
238 wire [DATA_WIDTH-1:0] readdata_9;
239 wire [DATA_WIDTH-1:0] readdata_10;
240 wire [DATA_WIDTH-1:0] readdata_11;
241 wire [DATA_WIDTH-1:0] readdata_12;
242 wire [DATA_WIDTH-1:0] readdata_13;
243 wire [DATA_WIDTH-1:0] readdata_14;
244 wire [DATA_WIDTH-1:0] readdata_15;
245 wire [15:0] irq_out;
246 wire [DATA_WIDTH-1:0] priority_int_src;
247 wire user_datain_reg_en;
248 wire user_dataout_reg_en;
249 wire irq_mask_reg_en;
250 wire edge_capture_reg_en;
251
252
253 // when the data width is 8 need to hardcode the 1 bit byteenable to high
254 generate
255   if (DATA_WIDTH == 8)
256     begin
257       assign internal_byteenable = 1'b1;
258     end
259   else
260     begin
261       assign internal_byteenable = slave_byteenable;
262     end
263 endgenerate
264
265
266 // sixteen address decodes (using one-hot encoding) A bank is considered to be a
267 // grouping of four addresses.
268 assign address_decode[0] = (slave_address[7:4] == 4'b0000) & (slave_write | slave_read);
269 assign address_decode[1] = (slave_address[7:4] == 4'b0001) & (slave_write | slave_read);
270 assign address_decode[2] = (slave_address[7:4] == 4'b0010) & (slave_write | slave_read);
271 assign address_decode[3] = (slave_address[7:4] == 4'b0011) & (slave_write | slave.read);

```

```

272 assign address_decode[5] = (slave_address[7:4] == 4'b0101) & (slave_write | slave_read);
273 assign address_decode[6] = (slave_address[7:4] == 4'b0110) & (slave_write | slave_read);
274 assign address_decode[7] = (slave_address[7:4] == 4'b0111) & (slave_write | slave.read);
275 assign address_decode[8] = (slave_address[7:4] == 4'b1000) & (slave_write | slave.read);
276 assign address_decode[9] = (slave_address[7:4] == 4'b1001) & (slave_write | slave.read);
277 assign address_decode[10] = (slave_address[7:4] == 4'b1010) & (slave_write | slave.read);
278 ;
279 assign address_decode[11] = (slave_address[7:4] == 4'b1011) & (slave_write | slave.read);
280 ;
281 assign address_decode[12] = (slave_address[7:4] == 4'b1100) & (slave_write | slave.read);
282 ;
283 assign address_decode[13] = (slave_address[7:4] == 4'b1101) & (slave_write | slave.read);
284 ;
285 assign address_decode[14] = (slave_address[7:4] == 4'b1110) & (slave_write | slave.read);
286 ;
287 assign address_decode[15] = (slave_address[7:4] == 4'b1111) & (slave_write | slave.read);
288 ;
289 assign address_bank_decode[0] = (address_decode_d1[3:0] != 0)? 1'b1 : 1'b0;
290 assign address_bank_decode[1] = (address_decode_d1[7:4] != 0)? 1'b1 : 1'b0;
291 assign address_bank_decode[2] = (address_decode_d1[11:8] != 0)? 1'b1 : 1'b0;
292 assign address_bank_decode[3] = (address_decode_d1[15:12] != 0)? 1'b1 : 1'b0;
293
294 // registering various address decoding registers and the input data
295 always @ (posedge clk or posedge reset)
296 begin
297     if (reset == 1)
298         begin
299             slave_read_d1 <= 0;
300             slave_read_d2 <= 0;
301             slave_write_d1 <= 0;
302             address_decode_d1 <= 0;
303             address_bank_decode_d1 <= 0;
304             internal_byteenable_d1 <= 0;
305             user_datain_0_d1 <= 0;
306             user_datain_1_d1 <= 0;
307             user_datain_2_d1 <= 0;
308             user_datain_3_d1 <= 0;
309             user_datain_4_d1 <= 0;
310             user_datain_5_d1 <= 0;
311             user_datain_6_d1 <= 0;
312             user_datain_7_d1 <= 0;
313             user_datain_8_d1 <= 0;
314             user_datain_9_d1 <= 0;
315             user_datain_10_d1 <= 0;
316             user_datain_11_d1 <= 0;
317             user_datain_12_d1 <= 0;
318             user_datain_13_d1 <= 0;
319             user_datain_14_d1 <= 0;
320             user_datain_15_d1 <= 0;
321         end
322     else
323         begin
324             slave_read_d1 <= slave_read;
325             slave_read_d2 <= slave_read_d1;
326             slave_write_d1 <= slave_write;
327             internal_byteenable_d1 <= internal_byteenable;
328             if ((slave_read == 1) | (slave_write == 1))
329                 begin
330                     address_decode_d1 <= address_decode;
331                 end
332             if (slave_read_d1 == 1)
333                 begin
334                     address_bank_decode_d1 <= address_bank_decode;

```

```

334     end
335     if((address_decode[0] == 1) & (slave_read == 1))
336     begin
337         user_datain_0_d1 <= user_datain_0;
338     end
339     if((address_decode[1] == 1) & (slave_read == 1))
340     begin
341         user_datain_1_d1 <= user_datain_1;
342     end
343     if((address_decode[2] == 1) & (slave_read == 1))
344     begin
345         user_datain_2_d1 <= user_datain_2;
346     end
347     if((address_decode[3] == 1) & (slave_read == 1))
348     begin
349         user_datain_3_d1 <= user_datain_3;
350     end
351     if((address_decode[4] == 1) & (slave_read == 1))
352     begin
353         user_datain_4_d1 <= user_datain_4;
354     end
355     if((address_decode[5] == 1) & (slave_read == 1))
356     begin
357         user_datain_5_d1 <= user_datain_5;
358     end
359     if((address_decode[6] == 1) & (slave_read == 1))
360     begin
361         user_datain_6_d1 <= user_datain_6;
362     end
363     if((address_decode[7] == 1) & (slave_read == 1))
364     begin
365         user_datain_7_d1 <= user_datain_7;
366     end
367     if((address_decode[8] == 1) & (slave_read == 1))
368     begin
369         user_datain_8_d1 <= user_datain_8;
370     end
371     if((address_decode[9] == 1) & (slave_read == 1))
372     begin
373         user_datain_9_d1 <= user_datain_9;
374     end
375     if((address_decode[10] == 1) & (slave_read == 1))
376     begin
377         user_datain_10_d1 <= user_datain_10;
378     end
379     if((address_decode[11] == 1) & (slave_read == 1))
380     begin
381         user_datain_11_d1 <= user_datain_11;
382     end
383     if((address_decode[12] == 1) & (slave_read == 1))
384     begin
385         user_datain_12_d1 <= user_datain_12;
386     end
387     if((address_decode[13] == 1) & (slave_read == 1))
388     begin
389         user_datain_13_d1 <= user_datain_13;
390     end
391     if((address_decode[14] == 1) & (slave_read == 1))
392     begin
393         user_datain_14_d1 <= user_datain_14;
394     end
395     if((address_decode[15] == 1) & (slave_read == 1))
396     begin
397         user_datain_15_d1 <= user_datain_15;
398     end
399 end
400

```

```

402 // Instantiate interrupt logic according to port type chosen and irq_en
403 generate
404   if ((IRQ_EN == 1) && ((MODE_0 == 1) || (MODE_0 == 2)))
405     begin
406       interrupt_logic interrupt_0 (clk, reset, user_datain_0, slave_write, slave_writedata
407                                 , address_decode[0], irq_mask_reg_en, edge_capture_reg_en, readdata_0, irq_out
408                                 [0]);
409       defparam interrupt_0.DATA_WIDTH = DATA_WIDTH;
410     end
411   else
412     begin
413       assign readdata_0 = 0;
414       assign irq_out[0] = 1'b0;
415     end
416
417   if ((IRQ_EN == 1) && ((MODE_1 == 1) || (MODE_1 == 2)))
418     begin
419       interrupt_logic interrupt_1 (clk, reset, user_datain_1, slave_write, slave_writedata
420                                 , address_decode[1], irq_mask_reg_en, edge_capture_reg_en, readdata_1, irq_out
421                                 [1]);
422       defparam interrupt_1.DATA_WIDTH = DATA_WIDTH;
423     end
424   else
425     begin
426       assign readdata_1 = 0;
427       assign irq_out[1] = 1'b0;
428     end
429
430   if ((IRQ_EN == 1) && ((MODE_2 == 1) || (MODE_2 == 2)))
431     begin
432       interrupt_logic interrupt_2 (clk, reset, user_datain_2, slave_write, slave_writedata
433                                 , address_decode[2], irq_mask_reg_en, edge_capture_reg_en, readdata_2, irq_out
434                                 [2]);
435       defparam interrupt_2.DATA_WIDTH = DATA_WIDTH;
436     end
437   else
438     begin
439       assign readdata_2 = 0;
440       assign irq_out[2] = 1'b0;
441     end
442
443   if ((IRQ_EN == 1) && ((MODE_3 == 1) || (MODE_3 == 2)))
444     begin
445       interrupt_logic interrupt_3 (clk, reset, user_datain_3, slave_write, slave_writedata
446                                 , address_decode[3], irq_mask_reg_en, edge_capture_reg_en, readdata_3, irq_out
447                                 [3]);
448       defparam interrupt_3.DATA_WIDTH = DATA_WIDTH;
449     end
450   else
451     begin
452       assign readdata_3 = 0;
453       assign irq_out[3] = 1'b0;
454     end
455
456   if ((IRQ_EN == 1) && ((MODE_4 == 1) || (MODE_4 == 2)))
457     begin
458       interrupt_logic interrupt_4 (clk, reset, user_datain_4, slave_write, slave_writedata
459                                 , address_decode[4], irq_mask_reg_en, edge_capture_reg_en, readdata_4, irq_out
460                                 [4]);
461       defparam interrupt_4.DATA_WIDTH = DATA_WIDTH;
462     end
463   else
464     begin
465       assign readdata_4 = 0;
466       assign irq_out[4] = 1'b0;
467     end
468
469   if ((IRQ_EN == 1) && ((MODE_5 == 1) || (MODE_5 == 2)))

```

```

460 begin
461     interrupt_logic interrupt_5 (clk, reset, user_datain_5, slave_write, slave_writedata
462         , address_decode[5], irq_mask_reg_en, edge_capture_reg_en, readdata_5, irq_out
463         [5]);
464     defparam interrupt_5.DATA_WIDTH = DATA_WIDTH;
465 end
466 else
467 begin
468     assign readdata_5 = 0;
469     assign irq_out[5] = 1'b0;
470 end
471
472 if ((IRQ_EN == 1) && ((MODE_6 == 1) || (MODE_6 == 2)))
473 begin
474     interrupt_logic interrupt_6 (clk, reset, user_datain_6, slave_write, slave_writedata
475         , address_decode[6], irq_mask_reg_en, edge_capture_reg_en, readdata_6, irq_out
476         [6]);
477     defparam interrupt_6.DATA_WIDTH = DATA_WIDTH;
478 end
479 else
480 begin
481     assign readdata_6 = 0;
482     assign irq_out[6] = 1'b0;
483 end
484
485 if ((IRQ_EN == 1) && ((MODE_7 == 1) || (MODE_7 == 2)))
486 begin
487     interrupt_logic interrupt_7 (clk, reset, user_datain_7, slave_write, slave_writedata
488         , address_decode[7], irq_mask_reg_en, edge_capture_reg_en, readdata_7, irq_out
489         [7]);
490     defparam interrupt_7.DATA_WIDTH = DATA_WIDTH;
491 end
492 else
493 begin
494     assign readdata_7 = 0;
495     assign irq_out[7] = 1'b0;
496 end
497
498 if ((IRQ_EN == 1) && ((MODE_8 == 1) || (MODE_8 == 2)))
499 begin
500     interrupt_logic interrupt_8 (clk, reset, user_datain_8, slave_write, slave_writedata
501         , address_decode[8], irq_mask_reg_en, edge_capture_reg_en, readdata_8, irq_out
502         [8]);
503     defparam interrupt_8.DATA_WIDTH = DATA_WIDTH;
504 end
505 else
506 begin
507     assign readdata_8 = 0;
508     assign irq_out[8] = 1'b0;
509 end
510
511 if ((IRQ_EN == 1) && ((MODE_9 == 1) || (MODE_9 == 2)))
512 begin
513     interrupt_logic interrupt_9 (clk, reset, user_datain_9, slave_write, slave_writedata
514         , address_decode[9], irq_mask_reg_en, edge_capture_reg_en, readdata_9, irq_out
515         [9]);
516     defparam interrupt_9.DATA_WIDTH = DATA_WIDTH;
517 end
518 else
519 begin
520     assign readdata_9 = 0;
521     assign irq_out[9] = 1'b0;
522 end
523
524 if ((IRQ_EN == 1) && ((MODE_10 == 1) || (MODE_10 == 2)))
525 begin
526     interrupt_logic interrupt_10 (clk, reset, user_datain_10, slave_write,
527         slave_writedata, address_decode[10], irq_mask_reg_en, edge_capture_reg_en,

```

```

      readdata_10 , irq_out[10]);
517  defparam interrupt_10 .DATA_WIDTH = DATA_WIDTH;
518 end
519 else
520 begin
521 assign readdata_10 = 0;
522 assign irq_out[10] = 1'b0;
523 end
524
525 if ((IRQ_EN == 1) && ((MODE_11 == 1) || (MODE_11 == 2)))
526 begin
527 interrupt_logic interrupt_11 (clk, reset, user_datain_11, slave_write,
528 slave_writedata, address_decode[11], irq_mask_reg_en, edge_capture_reg_en,
529 readdata_11, irq_out[11]);
530 defparam interrupt_11 .DATA_WIDTH = DATA_WIDTH;
531 end
532 else
533 begin
534 assign readdata_11 = 0;
535 assign irq_out[11] = 1'b0;
536 end
537
538 if ((IRQ_EN == 1) && ((MODE_12 == 1) || (MODE_12 == 2)))
539 begin
540 interrupt_logic interrupt_12 (clk, reset, user_datain_12, slave_write,
541 slave_writedata, address_decode[12], irq_mask_reg_en, edge_capture_reg_en,
542 readdata_12, irq_out[12]);
543 defparam interrupt_12 .DATA_WIDTH = DATA_WIDTH;
544 end
545 else
546 begin
547 assign readdata_12 = 0;
548 assign irq_out[12] = 1'b0;
549 end
550
551 if ((IRQ_EN == 1) && ((MODE_13 == 1) || (MODE_13 == 2)))
552 begin
553 interrupt_logic interrupt_13 (clk, reset, user_datain_13, slave_write,
554 slave_writedata, address_decode[13], irq_mask_reg_en, edge_capture_reg_en,
555 readdata_13, irq_out[13]);
556 defparam interrupt_13 .DATA_WIDTH = DATA_WIDTH;
557 end
558 else
559 begin
560 assign readdata_13 = 0;
561 assign irq_out[13] = 1'b0;
562 end
563
564 if ((IRQ_EN == 1) && ((MODE_14 == 1) || (MODE_14 == 2)))
565 begin
566 interrupt_logic interrupt_14 (clk, reset, user_datain_14, slave_write,
567 slave_writedata, address_decode[14], irq_mask_reg_en, edge_capture_reg_en,
568 readdata_14, irq_out[14]);
569 defparam interrupt_14 .DATA_WIDTH = DATA_WIDTH;
570 end
571 else
572 begin
573 assign readdata_14 = 0;
574 assign irq_out[14] = 1'b0;
575 end
576
577 if ((IRQ_EN == 1) && ((MODE_15 == 1) || (MODE_15 == 2)))
578 begin
579 interrupt_logic interrupt_15 (clk, reset, user_datain_15, slave_write,
580 slave_writedata, address_decode[15], irq_mask_reg_en, edge_capture_reg_en,
581 readdata_15, irq_out[15]);
582 defparam interrupt_15 .DATA_WIDTH = DATA_WIDTH;
583 end

```

```

574     else
575     begin
576       assign readdata_15 = 0;
577       assign irq_out[15] = 1'b0;
578     end
579
580   endgenerate
581
582 // sixteen output registers which use byteneables to register each byte. Disabling the
583 // write enable for output registers that were not needed.
584 register_with_bytelanes register_0 (clk,reset,slave_writedata,slave_write&address_decode
585 [0]&user_dataout_reg_en,internal_byteneable,user_dataout_0);
586 defparam register_0 .DATA_WIDTH = DATA_WIDTH;
587 register_with_bytelanes register_1 (clk,reset,slave_writedata,slave_write&address_decode
588 [1]&user_dataout_reg_en,internal_byteneable,user_dataout_1);
589 defparam register_1 .DATA_WIDTH = DATA_WIDTH;
590 register_with_bytelanes register_2 (clk,reset,slave_writedata,slave_write&address_decode
591 [2]&user_dataout_reg_en,internal_byteneable,user_dataout_2);
592 defparam register_2 .DATA_WIDTH = DATA_WIDTH;
593 register_with_bytelanes register_3 (clk,reset,slave_writedata,slave_write&address_decode
594 [3]&user_dataout_reg_en,internal_byteneable,user_dataout_3);
595 defparam register_3 .DATA_WIDTH = DATA_WIDTH;
596 register_with_bytelanes register_4 (clk,reset,slave_writedata,slave_write&address_decode
597 [4]&user_dataout_reg_en,internal_byteneable,user_dataout_4);
598 defparam register_4 .DATA_WIDTH = DATA_WIDTH;
599 register_with_bytelanes register_5 (clk,reset,slave_writedata,slave_write&address_decode
600 [5]&user_dataout_reg_en,internal_byteneable,user_dataout_5);
601 defparam register_5 .DATA_WIDTH = DATA_WIDTH;
602 register_with_bytelanes register_6 (clk,reset,slave_writedata,slave_write&address_decode
603 [6]&user_dataout_reg_en,internal_byteneable,user_dataout_6);
604 defparam register_6 .DATA_WIDTH = DATA_WIDTH;
605 register_with_bytelanes register_7 (clk,reset,slave_writedata,slave_write&address_decode
606 [7]&user_dataout_reg_en,internal_byteneable,user_dataout_7);
607 defparam register_7 .DATA_WIDTH = DATA_WIDTH;
608 register_with_bytelanes register_8 (clk,reset,slave_writedata,slave_write&address_decode
609 [8]&user_dataout_reg_en,internal_byteneable,user_dataout_8);
610 defparam register_8 .DATA_WIDTH = DATA_WIDTH;
611 register_with_bytelanes register_9 (clk,reset,slave_writedata,slave_write&address_decode
612 [9]&user_dataout_reg_en,internal_byteneable,user_dataout_9);
613 defparam register_9 .DATA_WIDTH = DATA_WIDTH;
614 register_with_bytelanes register_10 (clk,reset,slave_writedata,slave_write&
615 address_decode[10]&user_dataout_reg_en,internal_byteneable,user_dataout_10);
616 defparam register_10 .DATA_WIDTH = DATA_WIDTH;
617 register_with_bytelanes register_11 (clk,reset,slave_writedata,slave_write&
618 address_decode[11]&user_dataout_reg_en,internal_byteneable,user_dataout_11);
619 defparam register_11 .DATA_WIDTH = DATA_WIDTH;
620 register_with_bytelanes register_12 (clk,reset,slave_writedata,slave_write&
621 address_decode[12]&user_dataout_reg_en,internal_byteneable,user_dataout_12);
622 defparam register_12 .DATA_WIDTH = DATA_WIDTH;
623 register_with_bytelanes register_13 (clk,reset,slave_writedata,slave_write&
624 address_decode[13]&user_dataout_reg_en,internal_byteneable,user_dataout_13);
625 defparam register_13 .DATA_WIDTH = DATA_WIDTH;
626 register_with_bytelanes register_14 (clk,reset,slave_writedata,slave_write&
627 address_decode[14]&user_dataout_reg_en,internal_byteneable,user_dataout_14);
628 defparam register_14 .DATA_WIDTH = DATA_WIDTH;
629 register_with_bytelanes register_15 (clk,reset,slave_writedata,slave_write&
630 address_decode[15]&user_dataout_reg_en,internal_byteneable,user_dataout_15);
631 defparam register_15 .DATA_WIDTH = DATA_WIDTH;
632
633
634 // registered slave_readdata mux for all the return values , if an input register is
635 // disabled then a zero will be passed in
636 always @ (posedge clk or posedge reset)
637 begin
638   if (reset == 1)
639   begin
640     read_from_user <= 0;
641   end

```

```

624     else
625     begin
626       if (slave_read_d1 == 1) // first multiplexer stage
627       begin
628         // when the mode parameters are set to 3 then the outputs are looped back as
629         // inputs
630         case (address_decode_d1[3:0])
631           4'b0001: mux_first_stage_a <= (MODE_0 == 3)? user_dataout_0 : user_datain_0_d1;
632           4'b0010: mux_first_stage_a <= (MODE_1 == 3)? user_dataout_1 : user_datain_1_d1;
633           4'b0100: mux_first_stage_a <= (MODE_2 == 3)? user_dataout_2 : user_datain_2_d1;
634           4'b1000: mux_first_stage_a <= (MODE_3 == 3)? user_dataout_3 : user_datain_3_d1;
635         endcase
636         case (address_decode_d1[7:4])
637           4'b0001: mux_first_stage_b <= (MODE_4 == 3)? user_dataout_4 : user_datain_4_d1;
638           4'b0010: mux_first_stage_b <= (MODE_5 == 3)? user_dataout_5 : user_datain_5_d1;
639           4'b0100: mux_first_stage_b <= (MODE_6 == 3)? user_dataout_6 : user_datain_6_d1;
640           4'b1000: mux_first_stage_b <= (MODE_7 == 3)? user_dataout_7 : user_datain_7_d1;
641         endcase
642         case (address_decode_d1[11:8])
643           4'b0001: mux_first_stage_c <= (MODE_8 == 3)? user_dataout_8 : user_datain_8_d1;
644           4'b0010: mux_first_stage_c <= (MODE_9 == 3)? user_dataout_9 : user_datain_9_d1;
645           4'b0100: mux_first_stage_c <= (MODE_10 == 3)? user_dataout_10 :
646             user_datain_10_d1;
647           4'b1000: mux_first_stage_c <= (MODE_11 == 3)? user_dataout_11 :
648             user_datain_11_d1;
649         endcase
650         case (address_decode_d1[15:12])
651           4'b0001: mux_first_stage_d <= (MODE_12 == 3)? user_dataout_12 :
652             user_datain_12_d1;
653           4'b0010: mux_first_stage_d <= (MODE_13 == 3)? user_dataout_13 :
654             user_datain_13_d1;
655           4'b0100: mux_first_stage_d <= (MODE_14 == 3)? user_dataout_14 :
656             user_datain_14_d1;
657           4'b1000: mux_first_stage_d <= (MODE_15 == 3)? user_dataout_15 :
658             user_datain_15_d1;
659         endcase
660       end
661     end
662   end
663
664   if (slave_read_d2 == 1) // second multiplexer stage
665   begin
666     case (address_bank_decode_d1[3:0])
667       4'b0001: read_from_user <= mux_first_stage_a;
668       4'b0010: read_from_user <= mux_first_stage_b;
669       4'b0100: read_from_user <= mux_first_stage_c;
670       4'b1000: read_from_user <= mux_first_stage_d;
671     endcase
672   end
673 end
674
675 // multiplex the readdata from all I/O interrupt registers
676 always @ (posedge clk or posedge reset)
677 begin
678   if (reset == 1)
679   begin
680     read_from_int_regs <= 0;
681   end
682   else
683   begin
684     if (slave_read_d1 == 1) // first multiplexer stage
685     begin
686       // when the mode parameters are set to 3 then the outputs are looped back as
687       // inputs
688       case (address_decode_d1[3:0])
689         4'b0001: int_mux_first_stage_a <= readdata_0;
690         4'b0010: int_mux_first_stage_a <= readdata_1;
691         4'b0100: int_mux_first_stage_a <= readdata_2;
692         4'b1000: int_mux_first_stage_a <= readdata_3;
693     end

```

```

684     endcase
685     case (address_decode_d1[7:4])
686       4'b0001: int_mux_first_stage_b <= readdata_4;
687       4'b0010: int_mux_first_stage_b <= readdata_5;
688       4'b0100: int_mux_first_stage_b <= readdata_6;
689       4'b1000: int_mux_first_stage_b <= readdata_7;
690   endcase
691   case (address_decode_d1[11:8])
692     4'b0001: int_mux_first_stage_c <= readdata_8;
693     4'b0010: int_mux_first_stage_c <= readdata_9;
694     4'b0100: int_mux_first_stage_c <= readdata_10;
695     4'b1000: int_mux_first_stage_c <= readdata_11;
696   endcase
697   case (address_decode_d1[15:12])
698     4'b0001: int_mux_first_stage_d <= readdata_12;
699     4'b0010: int_mux_first_stage_d <= readdata_13;
700     4'b0100: int_mux_first_stage_d <= readdata_14;
701     4'b1000: int_mux_first_stage_d <= readdata_15;
702   endcase
703 end
704
705 if (slave_read_d2 == 1) // second multiplexer stage
706 begin
707   case (address_bank_decode_d1[3:0])
708     4'b0001: read_from_int_regs <= int_mux_first_stage_a;
709     4'b0010: read_from_int_regs <= int_mux_first_stage_b;
710     4'b0100: read_from_int_regs <= int_mux_first_stage_c;
711     4'b1000: read_from_int_regs <= int_mux_first_stage_d;
712   endcase
713 end
714 end
715 end
716
717 assign priority_int_src = (irq_out[0] == 1) ? 1 :
718   (irq_out[1] == 1) ? 2 :
719   (irq_out[2] == 1) ? 3 :
720   (irq_out[3] == 1) ? 4 :
721   (irq_out[4] == 1) ? 5 :
722   (irq_out[5] == 1) ? 6 :
723   (irq_out[6] == 1) ? 7 :
724   (irq_out[7] == 1) ? 8 :
725   (irq_out[8] == 1) ? 9 :
726   (irq_out[9] == 1) ? 10 :
727   (irq_out[10] == 1) ? 11 :
728   (irq_out[11] == 1) ? 12 :
729   (irq_out[12] == 1) ? 13 :
730   (irq_out[13] == 1) ? 14 :
731   (irq_out[14] == 1) ? 15 :
732   (irq_out[15] == 1) ? 16 : 0;
733
734 assign user_write = slave_write_d1; // outputs are registered so need a delayed copy of
735   the write signal
736 assign user_read = slave_read;
737 assign user_chipselect = (slave_write_d1 == 1)? address_decode_d1 : address_decode; // for write cycles need the delayed copy of the address decode since outputs are registered
738 assign user_bytelenable = (slave_write_d1 == 1)? internal_bytelenable_d1 :
739   internal_bytelenable; // for write cycles need the delayed copy of the bytelenables, don't use the bytelenables for reads since the full word is always sent on read transfers
740 assign slave_readdata = (slave_address == 9'h100)? priority_int_src : (
741   user_datain_reg_en == 1)? read_from_user : read_from_int_regs;
742 assign slave_irq = | (irq_out);
743
744 endmodule

```

```

745 // helper module to simplify having a register of variable width and containing
746 // independent byte lanes
747 module register_with_bytelanes (
748   clk ,
749   reset ,
750   data_in ,
751   write ,
752   byte_enables ,
753   data_out
754 );
755
756 parameter DATA_WIDTH = 32;
757
758 input clk;
759 input reset;
760
761 input [DATA_WIDTH-1:0] data_in;
762 input write;
763 input [(DATA_WIDTH/8)-1:0] byte_enables;
764 output reg [DATA_WIDTH-1:0] data_out;
765
766 // generating write logic for each group of 8 bits for 'data_out'
767 generate
768   genvar LANE;
769   for (LANE = 0; LANE < (DATA_WIDTH/8); LANE = LANE+1)
770     begin: register_bytelane_generation
771       always @ (posedge clk or posedge reset)
772       begin
773         if (reset == 1)
774           begin
775             data_out[(LANE*8)+7:(LANE*8)] <= 0;
776           end
777         else
778           begin
779             if ((byte_enables[LANE] == 1) & (write == 1))
780               begin
781                 data_out[(LANE*8)+7:(LANE*8)] <= data_in[(LANE*8)+7:(LANE*8)]; // write to
782                   // each byte lane with write = 1 and the lane bytelenable = 1
783               end
784             end
785           end
786     endgenerate
787
788 endmodule

```

Listing 16.16: Avalon-MM Slave Template - Altera Template (slave_template.v)

16.3.18 Avalon-MM Slave Template Macros - Altera Template (slave_template_macros.h)

```

1  /*******************************************************************************
2  * These are all word addresses. Please shift the
3  * offset accordingly when using them in IORD or IOWR.
4  * E.g.: Write to a 32-bit user_datain_0 port
5  *        IOWR_32DIRECT(BASE, DATA_IN_0 * 4, 0xFFFFFFFF)
6  * Another example: Read a 16-bit edge capture reg
7  *        IORD_16DIRECT(BASE, EDGE_CAPTURE_0 * 2)
8  *
9  ******************************************************************************/
10 #define DATA_IN_0 0x00
11 #define DATA_IN_1 0x10
12 #define DATA_IN_2 0x20
13 #define DATA_IN_3 0x30
14 #define DATA_IN_4 0x40

```

```

16 #define DATA_IN_5 0x50
17 #define DATA_IN_6 0x60
18 #define DATA_IN_7 0x70
19 #define DATA_IN_8 0x80
20 #define DATA_IN_9 0x90
21 #define DATA_IN_10 0xA0
22 #define DATA_IN_11 0xB0
23 #define DATA_IN_12 0xC0
24 #define DATA_IN_13 0xD0
25 #define DATA_IN_14 0xE0
26 #define DATA_IN_15 0xF0
27
28 #define DATA_OUT_0 0x01
29 #define DATA_OUT_1 0x11
30 #define DATA_OUT_2 0x21
31 #define DATA_OUT_3 0x31
32 #define DATA_OUT_4 0x41
33 #define DATA_OUT_5 0x51
34 #define DATA_OUT_6 0x61
35 #define DATA_OUT_7 0x71
36 #define DATA_OUT_8 0x81
37 #define DATA_OUT_9 0x91
38 #define DATA_OUT_10 0xA1
39 #define DATA_OUT_11 0xB1
40 #define DATA_OUT_12 0xC1
41 #define DATA_OUT_13 0xD1
42 #define DATA_OUT_14 0xE1
43 #define DATA_OUT_15 0xF1
44
45 #define IRQ_MASK_0 0x02
46 #define IRQ_MASK_1 0x12
47 #define IRQ_MASK_2 0x22
48 #define IRQ_MASK_3 0x32
49 #define IRQ_MASK_4 0x42
50 #define IRQ_MASK_5 0x52
51 #define IRQ_MASK_6 0x62
52 #define IRQ_MASK_7 0x72
53 #define IRQ_MASK_8 0x82
54 #define IRQ_MASK_9 0x92
55 #define IRQ_MASK_10 0xA2
56 #define IRQ_MASK_11 0xB2
57 #define IRQ_MASK_12 0xC2
58 #define IRQ_MASK_13 0xD2
59 #define IRQ_MASK_14 0xE2
60 #define IRQ_MASK_15 0xF2
61
62 #define EDGE_CAPTURE_0 0x03
63 #define EDGE_CAPTURE_1 0x13
64 #define EDGE_CAPTURE_2 0x23
65 #define EDGE_CAPTURE_3 0x33
66 #define EDGE_CAPTURE_4 0x43
67 #define EDGE_CAPTURE_5 0x53
68 #define EDGE_CAPTURE_6 0x63
69 #define EDGE_CAPTURE_7 0x73
70 #define EDGE_CAPTURE_8 0x83
71 #define EDGE_CAPTURE_9 0x93
72 #define EDGE_CAPTURE_10 0xA3
73 #define EDGE_CAPTURE_11 0xB3
74 #define EDGE_CAPTURE_12 0xC3
75 #define EDGE_CAPTURE_13 0xD3
76 #define EDGE_CAPTURE_14 0xE3
77 #define EDGE_CAPTURE_15 0xF3
78
79 #define PRIORITIZED_INTERRUPT_SRC 0x100

```

Listing 16.17: Avalon-MM Slave Template Macros - Altera Template (slave_template_macros.h)

16.4 FPGA Demo - Software Code

16.4.1 Partial Directory Structure

```
yocto-ghrd/
├── bitbake
│   └── <FILES>
├── build
│   └── conf
│       └── <FILES>
│           └── bblayers.conf
├── documentation
│   └── <FILES>
├── meta
│   └── <FILES>
├── meta-altera
│   ├── <FILES>
│   ├── recipes-core
│   │   └── <FILES>
│   │       ├── images
│   │       └── <FILES>
│   │           └── altera-gsrd.image.bb
│   ├── recipes-gsrd
│   │   └── <FILES>
│   ├── gsrd-altera
│   │   └── <FILES>
│   ├── linux
│   │   └── <FILES>
│   │       └── files
│   │           └── socfpga.dts
│   ├── meta-hob
│   │   └── <FILES>
│   ├── meta-linaro
│   │   └── <FILES>
│   ├── meta-oe
│   │   └── <FILES>
│   ├── meta-skeleton
│   │   └── <FILES>
│   ├── meta-snr-des
│   │   └── conf
│   │       └── layer.conf
│   ├── recipes-orion
│   │   └── smart-cam-ipstest
│   │       └── files
│   │           └── test.c
│   │           └── smart-cam-ipstest_1.0.bb
│   ├── meta-yocto
│   │   └── <FILES>
│   ├── meta-yocto-bsp
│   │   └── <FILES>
│   ├── scripts
│   │   └── <FILES>
│   ├── altera-init
│   └── LICENSE
```

```

└── make-sd.sh
└── oe-init-build-env
└── README
└── README.hardware

```

16.4.2 Senior Design Test BitBake Recipe (smart-cam-iptest_1.0.bb)

```

1 DESCRIPTION = "Software to Test ARM / FPGA Interface and Shared DDR3"
2 SECTION = "smart-cam-iptest"
3 LICENSE = "BSD"
4 LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/BSD;md5=3775480a712fc46a69647678acb234cb"
5 PR = "r0"
6 FILES_${PN} = "/home/root/snr-des/testIP/*"
7 FILES_${PN}-dbg = "/usr /home/root/snr-des/testIP/.debug"
8
9 SRC_URI = "file://test.c"
10
11 S = "${WORKDIR}"
12
13 #based on http://www.multitech.net/developer/products/multiconnect-ocg/development/writing-
14 #and http://www.yoctoproject.org/docs/current/dev-manual/dev-manual.html#new-recipe-
15 #writing-a-new-recipe
16
17 do_compile() {
18     ${CC} ${CFLAGS} ${LDFLAGS} test.c -o test
19 }
20
21 do_install() {
22     install -d ${D}/home/root/snr-des/testIP
23
24     install -m 0755 test ${D}/home/root/snr-des/testIP/test
25 }
```

Listing 16.18: Senior Design Test BitBake Recipe (smart-cam-iptest_1.0.bb)

16.4.3 Senior Design Test Application Source (test.c)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <errno.h>
5 #include <stdbool.h>
6 #include <sys/types.h>
7 #include <sys/stat.h>
8 #include <fcntl.h>
9 #include <sys/mman.h>
10
11 #define DDR3_ARM_BASE 0xc0000000
12 #define DDR3_ARM_END 0xcfffffff
13 #define DDR3_IP_BASE 0x00000000
14 #define DDR3_IP_END 0x0fffffff
15 // the linux image uses pages of width 4 KB or 0x00001000
16 // as it turns out, this is the mem space of the entire bridge
17 // used to connect to the CSR. Since we need to map an entire
18 // page, we need to take 0xff23000 to 0xff230fff.
19 // However, the real CSR starts at 0xff23800
20 #define IP_CSR_OFFSET 0x00000800
21 #define IP_CSR_BASE 0xff230000
22 #define IP_CSR_END 0xff230fff
23
24 // based on readme-first.txt from the altera template
25 // each register is the 2nd hexidesimal digit (16 place)
26 // the least significan hex digit represents the access
```

```

27 //mode such as 0 = read , 1 = write , 2 = interrupt mask
28 //3 = edge capture
29 #define ADDR_OFFSET      0x01 // address      (write)
30 #define MODE_OFFSET       0x11 //mode          (write) (0 = read , 1 = write)
31 #define DATA0W_OFFSET     0x21 // data0        (write)
32 #define DATA1W_OFFSET     0x31 // data1        (write)
33 #define DATA2W_OFFSET     0x41 // data2        (write)
34 #define DATA3W_OFFSET     0x51 // data3        (write)
35
36 #define READ_RDY_OFFSET   0x00 //readReady    (read) (0 when ready)
37 #define WRITE_RDY_OFFSET  0x10 // writeReady   (read) (0 when ready)
38 #define DATA0R_OFFSET     0x20 // data0        (read)
39 #define DATA1R_OFFSET     0x30 // data1        (read)
40 #define DATA2R_OFFSET     0x40 // data2        (read)
41 #define DATA3R_OFFSET     0x50 // data3        (read)
42
43 //Constants used by IP CSR
44 #define READ_MODE 0
45 #define WRITE_MODE 1
46 #define IP_READY 0
47
48 //info on how to use open and mmap on /dev/mem was based on http://stackoverflow.com/
49 //questions/9662193/how-to-access-kernel-space-from-user-space-in-linux
50 //o_sync is essential to ensure writes to and reads from /dev/mem are not buffered by the
51 //OS. Bitwise oring is used to specify multiple parms
52
53 int main(int argc, char** argv)
54 {
55     int memSize = DDR3_ARM_END - DDR3_ARM_BASE + 1;
56     int csrSize = IP_CSR-END - IP_CSR-BASE + 1;
57
58     int status;
59
60     bool testPass;
61
62     int test0, test1, test2, test3;
63
64     int d0 = 0;
65     int d1 = 0;
66     int d2 = 0;
67     int d3 = 0;
68
69     int memOffset = 0;
70
71     int exitStatus = EXIT_SUCCESS;
72
73     int* ddr3Map;
74     int* csrMap;
75
76     int memFile = open("/dev/mem", O_RDWR | O_SYNC);
77
78     if(memFile == -1){
79         int errNum = errno;
80         char* errorMsg = strerror(errNum);
81         printf("Error opening /dev/mem: %s\n", errorMsg);
82         return EXIT_FAILURE;
83     }
84
85     ddr3Map = (int *) mmap(NULL, memSize, PROT_READ | PROT_WRITE, MAP_SHARED, memFile,
86                           DDR3_ARM_BASE);
87
88     if(ddr3Map == MAP_FAILED){
89         int errNum = errno;
90         char* errorMsg = strerror(errNum);
91         printf("Error mapping DDR3: %s\n", errorMsg);
92         return EXIT_FAILURE;
93     }

```

```

92
93 csrMap = (int *) mmap(NULL, csrSize , PROT_READ | PROT_WRITE, MAP_SHARED, memFile ,
94 IP_CSR_BASE);
95
96 if(csrMap == MAP_FAILED){
97     int errNum = errno;
98     char* errorMsg = strerror(errNum);
99     printf("Error mapping IP CSR: %s\n", errorMsg);
100    return EXIT_FAILURE;
101 }
102
103 //***** Begin Test Code *****
104 //Read Test
105 test0 = 0x0000000F;
106 test1 = 0x000000F0;
107 test2 = 0x00000F00;
108 test3 = 0x0000F000;
109
110 *(ddr3Map + memOffset) = test0;
111 *(ddr3Map + memOffset+1) = test1;
112 *(ddr3Map + memOffset+2) = test2;
113 *(ddr3Map + memOffset+3) = test3;
114
115 printf("Wrote into DDR3 from ARM\n");
116 printf("0x%x\n", test0);
117 printf("0x%x\n", test1);
118 printf("0x%x\n", test2);
119 printf("0x%x\n", test3);
120
121 //read using IP
122 *(csrMap + IP_CSR_OFFSET + ADDR_OFFSET) = DDR3_IP.BASE+memOffset;
123 *(csrMap + IP_CSR_OFFSET + MODE_OFFSET) = READ_MODE;
124
125 while(((*csrMap + IP_CSR_OFFSET + READ_RDY_OFFSET) & 1) != IP_READY){
126     ; //wait for IP to finish read operation
127     //this is a spin lock, I know, but this is a simple test of IP
128     //and not a high speed application
129 }
130
131 d0 = *(csrMap + IP_CSR_OFFSET + DATA0R_OFFSET);
132 d1 = *(csrMap + IP_CSR_OFFSET + DATA1R_OFFSET);
133 d2 = *(csrMap + IP_CSR_OFFSET + DATA2R_OFFSET);
134 d3 = *(csrMap + IP_CSR_OFFSET + DATA3R_OFFSET);
135
136 printf("Read from DDR3 in IP\n");
137 printf("0x%x\n", d0);
138 printf("0x%x\n", d1);
139 printf("0x%x\n", d2);
140 printf("0x%x\n", d3);
141
142 testPass = test0==d0 && test1==d1 && test2==d2 && test3==d3;
143
144 if(testPass){
145     printf("***** Read Test: Success *****");
146 }
147 else{
148     printf("***** Read Test: Fail *****");
149 }
150
151 // Write test
152 memOffset = 0x00000010;
153
154 test0 = 0x11111111;
155 test1 = 0x22222222;
156 test2 = 0x33333333;
157 test3 = 0x44444444;

```

```

159 *(csrMap + IP_CSR_OFFSET + ADDR_OFFSET) = DDR3_IP_BASE+memOffset;
160 *(csrMap + IP_CSR_OFFSET + DATA0W_OFFSET) = test0;
161 *(csrMap + IP_CSR_OFFSET + DATA1W_OFFSET) = test1;
162 *(csrMap + IP_CSR_OFFSET + DATA2W_OFFSET) = test2;
163 *(csrMap + IP_CSR_OFFSET + DATA3W_OFFSET) = test3;
164 *(csrMap + IP_CSR_OFFSET + MODE_OFFSET) = WRITE_MODE;
165
166
167 while (((*(csrMap + IP_CSR_OFFSET + WRITE_RDY_OFFSET) & 1) != IP_READY) {
168     ; // wait for IP to finish write operation
169     // this is a spin lock, I know, but this is a simple test of IP
170     // and not a high speed application
171 }
172
173 printf("Wrote into DDR3 from IP\n");
174 printf("0x%08x\n", test0);
175 printf("0x%08x\n", test1);
176 printf("0x%08x\n", test2);
177 printf("0x%08x\n", test3);
178
179 printf("\n");
180
181 d0 = *(ddr3Map + memOffset);
182 d1 = *(ddr3Map + memOffset + 1);
183 d2 = *(ddr3Map + memOffset + 2);
184 d3 = *(ddr3Map + memOffset + 3);
185
186 printf("Read from DDR3 in ARM\n");
187 printf("0x%08x\n", d0);
188 printf("0x%08x\n", d1);
189 printf("0x%08x\n", d2);
190 printf("0x%08x\n", d3);
191
192 testPass = test0==d0 && test1==d1 && test2==d2 && test3==d3;
193
194 if (testPass){
195     printf("***** Write Test: Success *****");
196 }
197 else{
198     printf("***** Write Test: Fail *****");
199 }
200
201 // **** End Test Code ****
202
203 status = munmap(ddr3Map, memSize);
204
205 if (status == -1){
206     int errNum = errno;
207     char* errorMsg = strerror(errNum);
208     printf("Error un-mapping DDR3: %s\n", errorMsg);
209     exitStatus = EXIT_FAILURE;
210 }
211
212 status = munmap(csrMap, csrSize);
213
214 if (status == -1){
215     int errNum = errno;
216     char* errorMsg = strerror(errNum);
217     printf("Error un-mapping CSR: %s\n", errorMsg);
218     exitStatus = EXIT_FAILURE;
219 }
220
221 status = close(memFile);
222
223 if (status == -1){
224     int errNum = errno;
225     char* errorMsg = strerror(errNum);

```

```

227     printf("Error closing /dev/mem: %s\n", errorMsg);
228     exitStatus = EXIT_FAILURE;
229 }
230
231 return exitStatus;
232 }
```

Listing 16.19: Senior Design Test Application Source (test.c)

16.4.4 Senior Design Layer Configuration File (layer.conf)

```

1 # Based on OE Layer.conf file
2
3 # We have a conf and classes directory , append to BBPATH
4 BBPATH .= ":${LAYERDIR}"
5
6 # We have a recipes directory , add to BBFILES
7 BBFILES += "${LAYERDIR}/recipes -*/**/*.bb ${LAYERDIR}/recipes -*/**/*.bbappend"
8
9 BBFILE_COLLECTIONS += "snr-des-layer"
10 BBFILE_PATTERN_snr-des-layer := "^${LAYERDIR}/*"
11
12 # Define the priority for recipes (.bb files) from this layer ,
13 # choosing carefully how this layer interacts with all of the
14 # other layers .
15
16 BBFILE_PRIORITY_openembedded-layer = "1"
```

Listing 16.20: Senior Design Layer Configuration File (layer.conf)

16.4.5 Modified bblayers File (bblayers.conf)

```

1 # LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
2 # changes incompatibly
3 LCONF_VERSION = "6"
4
5 BBPATH = "${TOPDIR}"
6 BBFILES ?= ""
7 BBLAYERS = " \
8   /home/christopher/yocto-ghrd/meta-snr-des \
9   /home/christopher/yocto-ghrd/meta-oe \
10  /home/christopher/yocto-ghrd/meta \
11  /home/christopher/yocto-ghrd/meta-yocto \
12  /home/christopher/yocto-ghrd/meta-yocto-bsp \
13  /home/christopher/yocto-ghrd/meta-yocto-bsp \
14  /home/christopher/yocto-ghrd/meta-linaro \
15  /home/christopher/yocto-ghrd/meta-altera \
16  "
```

Listing 16.21: Modified bblayers File (bblayers.conf)

16.4.6 Modified Image File (altera-gsrd-image.bb)

```

1 DESCRIPTION = "The set of packages for development and testing provided by Altera"
2
3 LICENSE = "MIT"
4
```

```

6 | ALTERA_IMAGE_INSTALL ?= "initramfs-altera kernel-modules netbase busybox base-passwd base-
7 |   files tinylogin sysvinit initscripts e2fsprogs mtd-utils gdb gdbserver bash strace
8 |   openssh openssl elfutils sysfsutils usbutils dtc gawk ethtool grep lighttpd iutils
9 |   make pciutils portmap sed setserial wget autoconf diffutils perl libav live555 live555-
10 |   -openrtsp live555-mediaserver minicom valgrind i2c-tools lttng-modules lttng-tools
11 |   iptables oprofile net-tools gator openssh-sftp-server gsr-altera pio-interrupt-altera
12 |   initscripts-altera util-linux smart-cam-ipstest"
13 | IMAGE_INSTALL ?= "${ALTERA_IMAGE_INSTALL}"
14 |
15 | inherit core-image
16 |
17 | # altera-image.inc must be included after inherit core-image to override functionality
18 |
19 | include altera-image.inc

```

Listing 16.22: Modified Image File (altera-gsr-altera.bb)

16.4.7 Modified Device Tree File (socfpga.dts)

```

1 | /*
2 | * This devicetree is generated by sopc2dts on Sat May 03 22:38:06 PDT 2014
3 | * Sopc2dts is written by Walter Goossens <waltergoossens@home.nl>
4 | * in cooperation with the nios2 community <Nios2-dev@sopc.et.ntust.edu.tw>
5 | */
6 | /dts-v1/;
7 |
8 | {
9 |     model = "ALTR,socfpga-cyclone5";
10 |     compatible = "ALTR,socfpga-cyclone5";
11 |     #address-cells = < 1 >;
12 |     #size-cells = < 1 >;
13 |
14 |     aliases {
15 |         ethernet0 = "/sopc/ethernet@0xff702000";
16 |         serial0 = "/sopc/serial@xffc02000";
17 |         serial1 = "/sopc/serial@xffc03000";
18 |         timer0 = "/sopc/timer@xffc08000";
19 |         timer1 = "/sopc/timer@xffc09000";
20 |         timer2 = "/sopc/timer@0ffd00000";
21 |         timer3 = "/sopc/timer@0ffd01000";
22 |     }; //end aliases
23 |
24 |     cpus {
25 |         #address-cells = < 1 >;
26 |         #size-cells = < 0 >;
27 |
28 |         hps_0_arm_a9_0: cpu@0x0 {
29 |             device_type = "cpu";
30 |             compatible = "arm,cortex-a9-1.0", "arm,cortex-a9";
31 |             reg = < 0x00000000 >;
32 |         }; //end cpu@0x0 (hps_0_arm_a9_0)
33 |
34 |         hps_0_arm_a9_1: cpu@0x1 {
35 |             device_type = "cpu";
36 |             compatible = "arm,cortex-a9-1.0", "arm,cortex-a9";
37 |             reg = < 0x00000001 >;
38 |         }; //end cpu@0x1 (hps_0_arm_a9_1)
39 |     }; //end cpus
40 |
41 |     memory@0 {
42 |         device_type = "memory";
43 |         reg = < 0xC0000000 0x10000000
44 |                 0xFFFFF0000 0x00010000 >;
45 |     }; //end memory@0
46 |
47 |     sopc@0 {

```

```

48 device_type = "soc";
49 ranges;
50 #address-cells = < 1 >;
51 #size-cells = < 1 >;
52 compatible = "ALTR, avalon", "simple-bus";
53 bus-frequency = < 50000000 >;
54
55 hps_0_h2f_lw: bridge@0xff200000 {
56     compatible = "altr ,h2f_lw_bridge -1.0", "simple -bus";
57     reg = < 0xFF200000 0x00200000 >;
58     #address-cells = < 1 >;
59     #size-cells = < 1 >;
60     ranges = < 0x00010000 0xFF210000 0x00000008
61         0x00010040 0xFF210040 0x00000020
62         0x00010080 0xFF210080 0x00000010
63         0x000100C0 0xFF2100C0 0x00000010
64         0x00020000 0xFF220000 0x00000008
65         0x00030000 0xFF230000 0x00001000 >;
66
67 led_pio: gpio@0x10040 {
68     compatible = "ALTR, pio -13.0.1.99.2", "ALTR, pio -1.0", "altr ,pio -1.0";
69     reg = < 0x00010040 0x00000020 >;
70     width = < 4 >; /* width type NUMBER */
71     resetvalue = < 0 >; /* resetValue type NUMBER */
72     #gpio-cells = < 2 >;
73     gpio-controller;
74 }; //end gpio@0x10040 (led_pio)
75
76 dipsw_pio: gpio@0x10080 {
77     compatible = "ALTR, pio -13.0.1.99.2", "ALTR, pio -1.0", "altr ,pio -1.0";
78     reg = < 0x00010080 0x00000010 >;
79     interrupt-parent = < &hps_0_arm_gic_0 >;
80     interrupts = < 0 40 1 >;
81     width = < 4 >; /* width type NUMBER */
82     resetvalue = < 0 >; /* resetValue type NUMBER */
83     edge-type = < 2 >; /* embeddedsw.dts.params.edge-type type NUMBER */
84     level-trigger = < 0 >; /* embeddedsw.dts.params.level-trigger type NUMBER */
85     #gpio-cells = < 2 >;
86     gpio-controller;
87 }; //end gpio@0x10080 (dipsw_pio)
88
89 button_pio: gpio@0x100c0 {
90     compatible = "ALTR, pio -13.0.1.99.2", "ALTR, pio -1.0", "altr ,pio -1.0";
91     reg = < 0x000100C0 0x00000010 >;
92     interrupt-parent = < &hps_0_arm_gic_0 >;
93     interrupts = < 0 41 1 >;
94     width = < 2 >; /* width type NUMBER */
95     resetvalue = < 0 >; /* resetValue type NUMBER */
96     edge-type = < 1 >; /* embeddedsw.dts.params.edge-type type NUMBER */
97     level-trigger = < 0 >; /* embeddedsw.dts.params.level-trigger type NUMBER */
98     #gpio-cells = < 2 >;
99     gpio-controller;
100 }; //end gpio@0x100c0 (button_pio)
101
102 jtag_uart: serial@0x20000 {
103     compatible = "ALTR, juart -13.0.1.99.2", "ALTR, juart -1.0";
104     reg = < 0x00020000 0x00000008 >;
105     interrupt-parent = < &hps_0_arm_gic_0 >;
106     interrupts = < 0 42 4 >;
107 }; //end serial@0x20000 (jtag_uart)
108
109 mm_clock_crossing_bridge_1: bridge@0x30000 {
110     compatible = "ALTR, avalon -13.0", "simple -bus";
111     reg = < 0x00030000 0x00001000 >;
112     #address-cells = < 1 >;
113     #size-cells = < 1 >;
114     ranges = < 0x00000800 0x00030800 0x00000800 >;
115 };

```

```

116     IPTest_0: unknown@0x800 {
117         compatible = "unknown,unknown-1.0", "simple-bus";
118         reg = < 0x00000800 0x00000800 >;
119     }; //end unknown@0x800 (IPTest_0)
120 }; //end bridge@0x30000 (mm_clock_crossing_bridge_1)
121 }; //end bridge@0xff200000 (hps_0_h2f_lw)
122
123 hps_0_arm_gic_0: intc@0xffffed000 {
124     compatible = "arm,cortex-a9-gic-1.0", "arm,cortex-a9-gic";
125     reg = < 0xFFFFED000 0x00001000
126         0xFFFECL00 0x00001000 >;
127     interrupt-controller;
128     #interrupt-cells = < 3 >;
129 }; //end intc@0xffffed000 (hps_0_arm_gic_0)
130
131 hps_0_L2: L2-cache@0xffffef000 {
132     compatible = "arm,pl310-cache-1.0", "arm,pl310-cache";
133     reg = < 0xFFFFEF000 0x00001000 >;
134     interrupt-parent = < &hps_0_arm_gic_0 >;
135     interrupts = < 0 38 4 >;
136     cache-level = < 2 >; /* embeddedsw.dts.params.cache-level type NUMBER */
137     cache-unified; /* appended from boardinfo */
138     arm,tag-latency = <1 1 1>;
139     arm,data-latency = <2 1 1>;
140 }; //end L2-cache@0xffffef000 (hps_0_L2)
141
142 hps_0_dma: dma@0xffe01000 {
143     compatible = "arm,pl330-1.0", "arm,pl330", "arm,primecell";
144     reg = < 0xFFE01000 0x00001000 >;
145     interrupt-parent = < &hps_0_arm_gic_0 >;
146     interrupts = < 0 104 4 >;
147 }; //end dma@0xffe01000 (hps_0_dma)
148
149 hps_0_sysmgr: sysmgr@0xffd08000 {
150     compatible = "altr,sys-mgr-1.0", "altr,sys-mgr";
151     reg = < 0xFFD08000 0x00004000 >;
152 }; //end sysmgr@0xffd08000 (hps_0_sysmgr)
153
154 hps_0_clkmgr: clkmgr@0xffd04000 {
155     compatible = "altr,clk-mgr-1.0", "altr,clk-mgr";
156     reg = < 0xFFD04000 0x00001000 >;
157 }; //end clkmgr@0xffd04000 (hps_0_clkmgr)
158
159 hps_0_rstmgr: rstmgr@0xffd05000 {
160     compatible = "altr,rst-mgr-1.0", "altr,rst-mgr";
161     reg = < 0xFFD05000 0x00001000 >;
162 }; //end rstmgr@0xffd05000 (hps_0_rstmgr)
163
164 hps_0_fpgamgr: fpgamgr@0xff706000 {
165     compatible = "altr,fpga-mgr-1.0", "altr,fpga-mgr";
166     reg = < 0xFF706000 0x00001000
167         0xFFB90000 0x00001000 >;
168     interrupt-parent = < &hps_0_arm_gic_0 >;
169     interrupts = < 0 175 4 >;
170     transport = "mmio"; /* embeddedsw.dts.params.transport type STRING */
171 }; //end fpgamgr@0xff706000 (hps_0_fpgamgr)
172
173 hps_0_uart0: serial@0xffc02000 {
174     compatible = "snps,dw-apb-uart-1.0", "snps,dw-apb-uart";
175     reg = < 0xFFC02000 0x00001000 >;
176     interrupt-parent = < &hps_0_arm_gic_0 >;
177     interrupts = < 0 162 4 >;
178     reg-io-width = < 4 >; /* embeddedsw.dts.params.reg-io-width type NUMBER */
179     reg-shift = < 2 >; /* embeddedsw.dts.params.reg-shift type NUMBER */
180     clock-frequency = < 100000000 >; /* appended from boardinfo */
181 }; //end serial@0xffc02000 (hps_0_uart0)
182
183 hps_0_uart1: serial@0xffc03000 {

```

```

184 compatible = "snps,dw-apb-uart-1.0", "snps,dw-apb-uart";
185 reg = < 0xFFC03000 0x00001000 >;
186 interrupt-parent = < &hps_0_arm_gic_0 >;
187 interrupts = < 0 163 4 >;
188 reg-io-width = < 4 >; /* embeddedsw.dts.params.reg-io-width type NUMBER */
189 reg-shift = < 2 >; /* embeddedsw.dts.params.reg-shift type NUMBER */
190 clock-frequency = < 100000000 >; /* appended from boardinfo */
191 }; //end serial@0xffc03000 (hps_0_uart1)
192
193 hps_0_timer0: timer@0xffc08000 {
194     compatible = "snps,dw-apb-timer-sp-1.0", "snps,dw-apb-timer-sp";
195     reg = < 0xFFC08000 0x00001000 >;
196     interrupt-parent = < &hps_0_arm_gic_0 >;
197     interrupts = < 0 167 4 >;
198     clock-frequency = < 100000000 >; /* appended from boardinfo */
199 }; //end timer@0xffc08000 (hps_0_timer0)
200
201 hps_0_timer1: timer@0xffc09000 {
202     compatible = "snps,dw-apb-timer-sp-1.0", "snps,dw-apb-timer-sp";
203     reg = < 0xFFC09000 0x00001000 >;
204     interrupt-parent = < &hps_0_arm_gic_0 >;
205     interrupts = < 0 168 4 >;
206     clock-frequency = < 100000000 >; /* appended from boardinfo */
207 }; //end timer@0xffc09000 (hps_0_timer1)
208
209 hps_0_timer2: timer@0xffd00000 {
210     compatible = "snps,dw-apb-timer-osc-1.0", "snps,dw-apb-timer-osc";
211     reg = < 0xFFD00000 0x00001000 >;
212     interrupt-parent = < &hps_0_arm_gic_0 >;
213     interrupts = < 0 169 4 >;
214     clock-frequency = < 25000000 >; /* appended from boardinfo */
215 }; //end timer@0xffd00000 (hps_0_timer2)
216
217 hps_0_timer3: timer@0xffd01000 {
218     compatible = "snps,dw-apb-timer-osc-1.0", "snps,dw-apb-timer-osc";
219     reg = < 0xFFD01000 0x00001000 >;
220     interrupt-parent = < &hps_0_arm_gic_0 >;
221     interrupts = < 0 170 4 >;
222     clock-frequency = < 25000000 >; /* appended from boardinfo */
223 }; //end timer@0xffd01000 (hps_0_timer3)
224
225 hps_0_gpio0: gpio@0xff708000 {
226     compatible = "snps,dw-gpio-1.0", "snps,dw-gpio";
227     reg = < 0xFF708000 0x00001000 >;
228     interrupt-parent = < &hps_0_arm_gic_0 >;
229     interrupts = < 0 164 4 >;
230     #gpio-cells = < 2 >;
231     gpio-controller;
232 }; //end gpio@0xff708000 (hps_0_gpio0)
233
234 hps_0_gpio1: gpio@0xff709000 {
235     compatible = "snps,dw-gpio-1.0", "snps,dw-gpio";
236     reg = < 0xFF709000 0x00001000 >;
237     interrupt-parent = < &hps_0_arm_gic_0 >;
238     interrupts = < 0 165 4 >;
239     #gpio-cells = < 2 >;
240     gpio-controller;
241 }; //end gpio@0xff709000 (hps_0_gpio1)
242
243 hps_0_gpio2: gpio@0xff70a000 {
244     compatible = "snps,dw-gpio-1.0", "snps,dw-gpio";
245     reg = < 0xFF70A000 0x00001000 >;
246     interrupt-parent = < &hps_0_arm_gic_0 >;
247     interrupts = < 0 166 4 >;
248     #gpio-cells = < 2 >;
249     gpio-controller;
250 }; //end gpio@0xff70a000 (hps_0_gpio2)
251

```

```

252 hps_0_i2c0: i2c@0xffc04000 {
253     compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
254     reg = < 0xFFC04000 0x00001000 >;
255     interrupt-parent = < &hps_0_arm_gic_0 >;
256     interrupts = < 0 158 4 >;
257     emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type
258         NUMBER */
259     #address-cells = < 1 >;
260     #size-cells = < 0 >;
261     speed-mode = < 0 >; /* appended from boardinfo */
262
263 lcd: newhaven,nhd-0216k3z-nsw-bbw@0x28 {
264     compatible = "newhaven,nhd-0216k3z-nsw-bbw";
265     reg = < 0x00000028 >;
266     height = < 2 >; /* appended from boardinfo */
267     width = < 16 >; /* appended from boardinfo */
268 }; //end newhaven,nhd-0216k3z-nsw-bbw@0x28 (lcd)
269
270 eeprom: atmel,24c32@0x51 {
271     compatible = "atmel,24c32";
272     reg = < 0x00000051 >;
273     pagesize = < 32 >; /* appended from boardinfo */
274 }; //end atmel,24c32@0x51 (eeprom)
275 }; //end i2c@0xffc04000 (hps_0_i2c0)
276
277 hps_0_i2c1: i2c@0xffc05000 {
278     compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
279     reg = < 0xFFC05000 0x00001000 >;
280     interrupt-parent = < &hps_0_arm_gic_0 >;
281     interrupts = < 0 159 4 >;
282     emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type
283         NUMBER */
284 }; //end i2c@0xffc05000 (hps_0_i2c1)
285
286 hps_0_i2c2: i2c@0xffc06000 {
287     compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
288     reg = < 0xFFC06000 0x00001000 >;
289     interrupt-parent = < &hps_0_arm_gic_0 >;
290     interrupts = < 0 160 4 >;
291     emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type
292         NUMBER */
293 }; //end i2c@0xffc06000 (hps_0_i2c2)
294
295 hps_0_i2c3: i2c@0xffc07000 {
296     compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
297     reg = < 0xFFC07000 0x00001000 >;
298     interrupt-parent = < &hps_0_arm_gic_0 >;
299     interrupts = < 0 161 4 >;
300     emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type
301         NUMBER */
302 }; //end i2c@0xffc07000 (hps_0_i2c3)
303
304 hps_0_nand0: flash@0xff900000 {
305     compatible = "denali,nand-1.0", "denali,denali-nand-dt";
306     reg = < 0xFF900000 0x00100000
307         0xFFB80000 0x00010000 >;
308     interrupt-parent = < &hps_0_arm_gic_0 >;
309     interrupts = < 0 144 4 >;
310     #address-cells = < 1 >; /* embeddedsw.dts.params.#adress-cells type NUMBER */
311     #size-cells = < 1 >; /* embeddedsw.dts.params.#size-cells type NUMBER */
312     reg-names = "nand_data", "denali_reg"; /* embeddedsw.dts.params.reg-names type
313         STRING */
314     bank-width = < 2 >;
315     device-width = < 1 >;
316 }; //end flash@0xff900000 (hps_0_nand0)
317
318 hps_0_qspi: flash@0xff705000 {
319     compatible = "cadence,qspi-1.0", "cadence,qspi";

```

```

315     reg = < 0xFF705000 0x00001000
316         0xFFA00000 0x00001000 >;
317     interrupt-parent = < &hps_0_arm_gic_0 >;
318     interrupts = < 0 151 4 >;
319     bus-num = < 2 >; /* embeddedsw.dts.params.bus-num type NUMBER */
320     fifo-depth = < 128 >; /* embeddedsw.dts.params fifo-depth type NUMBER */
321     num-chipselect = < 4 >; /* embeddedsw.dts.params.num-chipselect type NUMBER */
322     bank-width = < 2 >;
323     device-width = < 1 >;
324     master-ref-clk = < 400000000 >; /* appended from boardinfo */
325     ext-decoder = < 0 >; /* appended from boardinfo */
326
327     flash0: n25q128@0 {
328         #address-cells = < 1 >; /* appended from boardinfo */
329         #size-cells = < 1 >; /* appended from boardinfo */
330         compatible = "n25q128"; /* appended from boardinfo */
331         reg = < 0 >; /* appended from boardinfo */
332         spi-max-frequency = < 100000000 >; /* appended from boardinfo */
333         page-size = < 256 >; /* appended from boardinfo */
334         block-size = < 16 >; /* appended from boardinfo */
335         quad = < 1 >; /* appended from boardinfo */
336         tshsl-nr = < 200 >; /* appended from boardinfo */
337         tsd2d-nr = < 255 >; /* appended from boardinfo */
338         tchsh-nr = < 20 >; /* appended from boardinfo */
339         tslch-nr = < 20 >; /* appended from boardinfo */
340
341         part0: partition@0 {
342             label = "Flash 0 Raw Data"; /* appended from boardinfo */
343         }; //end partition@0 (part0)
344
345         part1: partition@800000 {
346             label = "Flash 1 jffs2 Filesystem"; /* appended from boardinfo */
347         }; //end partition@800000 (part1)
348     }; //end n25q128@0 (flash0)
349 }; //end flash@0xff705000 (hps_0_qspi)
350
351 hps_0_sdmmc: flash@0xff704000 {
352     compatible = "snps,mmc-1.0", "snps,dw-mshc";
353     reg = < 0xFF704000 0x00001000 >;
354     interrupt-parent = < &hps_0_arm_gic_0 >;
355     interrupts = < 0 139 4 >;
356     fifo-depth = < 1024 >; /* embeddedsw.dts.params fifo-depth type NUMBER */
357     num-slots = < 1 >; /* embeddedsw.dts.params.num-slots type NUMBER */
358     bank-width = < 2 >;
359     device-width = < 1 >;
360     bus-hz = < 12500000 >; /* appended from boardinfo */
361     #address-cells = < 1 >; /* appended from boardinfo */
362     #size-cells = < 0 >; /* appended from boardinfo */
363     supports-highspeed; /* appended from boardinfo */
364     broken-cd; /* appended from boardinfo */
365
366     slot_0: slot@0 {
367         reg = < 0 >; /* appended from boardinfo */
368         bus-width = < 4 >; /* appended from boardinfo */
369     }; //end slot@0 (slot_0)
370 }; //end flash@0xff704000 (hps_0_sdmmc)
371
372 hps_0_usb0: usb@0xffb00000 {
373     compatible = "snps,dwc-otg-1.0", "snps,dwc-otg";
374     reg = < 0xFFB00000 0x00001000 >;
375     interrupt-parent = < &hps_0_arm_gic_0 >;
376     interrupts = < 0 125 4 >;
377     dev-nperio-tx-fifo-size = < 4096 >; /* embeddedsw.dts.params.dev-nperio-tx-fifo-size
378         type NUMBER */
379     dev-perio-tx-fifo-size = "<512 512 512 512 512 512 512 512 512 512 512 512 512
380         512>"; /* embeddedsw.dts.params.dev-perio-tx-fifo-size type STRING */
381     dev-tx-fifo-size = "<512 512 512 512 512 512 512 512 512 512 512 512 512 512>";
382         /* embeddedsw.dts.params.dev-tx-fifo-size type STRING */

```

```

380 dev_rx_fifo_size = < 512 >; /* embeddedsw.dts.params.dev_rx_fifo_size type NUMBER */
381 dma-mask = < 268435455 >; /* embeddedsw.dts.params.dma-mask type NUMBER */
382 host_rx_fifo_size = < 512 >; /* embeddedsw.dts.params.host_rx_fifo_size type NUMBER */
383 */
384 ulpi-ddr = < 0 >; /* embeddedsw.dts.params.ulpi-ddr type NUMBER */
385 voltage-switch = < 0 >; /* embeddedsw.dts.params.voltage-switch type NUMBER */
386 }; //end usb@0xffb00000 (hps_0_usb0)

387 hps_0_usb1: usb@0xffb40000 {
388     compatible = "snps,dwc-otg-1.0", "snps,dwc-otg";
389     reg = < 0xFFB40000 0x00001000 >;
390     interrupt-parent = < &hps_0_arm_gic_0 >;
391     interrupts = < 0 128 4 >;
392     dev-nperio-tx-fifo-size = < 4096 >; /* embeddedsw.dts.params.dev-nperio-tx-fifo-size
393         type NUMBER */
394     dev-perio-tx-fifo-size = "<512 512 512 512 512 512 512 512 512 512 512 512 512
395         512>"; /* embeddedsw.dts.params.dev-perio-tx-fifo-size type STRING */
396     dev-tx-fifo-size = "<512 512 512 512 512 512 512 512 512 512 512 512 512
397         512>"; /* embeddedsw.dts.params.dev-tx-fifo-size type STRING */
398     dev_rx_fifo_size = < 512 >; /* embeddedsw.dts.params.dev_rx_fifo_size type NUMBER */
399     dma-mask = < 268435455 >; /* embeddedsw.dts.params.dma-mask type NUMBER */
400     host_rx_fifo_size = < 512 >; /* embeddedsw.dts.params.host_rx_fifo_size type NUMBER */
401         */
402     ulpi-ddr = < 0 >; /* embeddedsw.dts.params.ulpi-ddr type NUMBER */
403     voltage-switch = < 0 >; /* embeddedsw.dts.params.voltage-switch type NUMBER */
404 }; //end usb@0xffb40000 (hps_0_usb1)

405 hps_0_gmac0: ethernet@0xff700000 {
406     compatible = "synopsys,dwmac-1.0", "altr,socfpga-stmmac", "snps,dwmac-3.70a", "snps,
407         dwmac";
408     reg = < 0xFF700000 0x00002000 >;
409     interrupt-parent = < &hps_0_arm_gic_0 >;
410     interrupts = < 0 115 4 >;
411     interrupt-names = "macirq"; /* embeddedsw.dts.params.interrupt-names type STRING */
412     mac-address = "[00 00 00 00 00 01]"; /* embeddedsw.dts.params.mac-address type
413         STRING */
414     address-bits = < 48 >;
415     max-frame-size = < 1518 >;
416     local-mac-address = [ 00 00 00 00 00 00 ];
417     status = "disabled"; /* appended from boardinfo */
418 }; //end ethernet@0xff700000 (hps_0_gmac0)

419 hps_0_gmac1: ethernet@0xff702000 {
420     compatible = "synopsys,dwmac-1.0", "altr,socfpga-stmmac", "snps,dwmac-3.70a", "snps,
421         dwmac";
422     reg = < 0xFF702000 0x00002000 >;
423     interrupt-parent = < &hps_0_arm_gic_0 >;
424     interrupts = < 0 120 4 >;
425     interrupt-names = "macirq"; /* embeddedsw.dts.params.interrupt-names type STRING */
426     mac-address = "[00 00 00 00 00 01]"; /* embeddedsw.dts.params.mac-address type
427         STRING */
428     address-bits = < 48 >;
429     max-frame-size = < 1518 >;
430     local-mac-address = [ 00 00 00 00 00 00 ];
431     phy-mode = "rgmii"; /* appended from boardinfo */
432     phy-addr = < 0xFFFFFFFF >; /* appended from boardinfo */
433 }; //end ethernet@0xff702000 (hps_0_gmac1)

434 hps_0_timer: timer@0xffffec600 {
435     compatible = "arm,cortex-a9-twd-timer-1.0", "arm,cortex-a9-twd-timer";
436     reg = < 0xFFFFEC600 0x00000100 >;
437     interrupt-parent = < &hps_0_arm_gic_0 >;
438     interrupts = < 1 13 3844 >;
439 }; //end timer@0xffffec600 (hps_0_timer)
440 }; //end socp@0

441 leds {
442     compatible = "gpio-leds";

```

```

439    fpga0 {
440        gpios = <&led_pio 0 1>;
441        label = "fpga_led0";
442    };
443    fpga1 {
444        gpios = <&led_pio 1 1>;
445        label = "fpga_led1";
446    };
447    fpga2 {
448        gpios = <&led_pio 2 1>;
449        label = "fpga_led2";
450    };
451    fpga3 {
452        gpios = <&led_pio 3 1>;
453        label = "fpga_led3";
454    };
455    hps0 {
456        gpios = <&hps_0_gpio1 15 1>;
457        label = "hps_led0";
458    };
459    hps1 {
460        gpios = <&hps_0_gpio1 14 1>;
461        label = "hps_led1";
462    };
463    hps2 {
464        gpios = <&hps_0_gpio1 13 1>;
465        label = "hps_led2";
466    };
467    hps3 {
468        gpios = <&hps_0_gpio1 12 1>;
469        label = "hps_led3";
470    };
471};
472chosen {
473    bootargs = "console=ttyS0 ,57600";
474}; //end chosen
475}; //end /

```

Listing 16.23: Modified Device Tree File (socfpga.dts)

16.4.8 SD Card Flash Script (make-sd.sh)

```

1 #!/bin/bash
2
3 set -v
4
5 cd ~/yocto-ghrd/build/tmp/deploy/images
6 sudo rm -rf rootfs
7 mkdir rootfs
8 cd rootfs
9 sudo tar xzf ../../altera-gsrd-image-socfpga_cyclone5.tar.gz
10 cd ..
11 rm -f sd_image_yocto.bin
12 sudo /opt/altera-linux-ghrd/bin/make_sdimage.sh -k uImage,socfpga.dtb -rp u-boot-spl-
    socfpga_cyclone5.bin -t ~/altera/13.0sp1/embedded/host_tools/altera/mkpimage/mkpimage
    -b u-boot-socfpga_cyclone5.img -r rootfs/ -o sd_image_yocto.bin

```

Listing 16.24: SD Card Flash Script (make-sd.sh)

16.5 Database - SQL Table Creation Code

```

1 IF object_id('Anomaly') is not null
2 BEGIN

```

```

3   DROP TABLE Anomaly
4 END
5
6 IF object_id('DetectionEvent') is not null
7 BEGIN
8   DROP TABLE DetectionEvent
9 END
10
11 IF object_id('CameraStats') is not null
12 BEGIN
13   DROP TABLE CameraStats
14 END
15
16 IF object_id('Camera') is not null
17 BEGIN
18   DROP TABLE Camera
19 END
20
21 IF object_id('KnownImg') is not null
22 BEGIN
23   DROP TABLE KnownImg
24 END
25
26 IF object_id('Individual') is not null
27 BEGIN
28   DROP TABLE Individual
29 END
30
31 CREATE TABLE Individual
32 (
33   ID Integer IDENTITY(1,1) Not Null PRIMARY KEY,
34   Name Varchar(40) Not Null ,
35   Details Varchar(100) Null
36 );
37
38 CREATE TABLE KnownImg
39 (
40   KnownImgID Integer IDENTITY(1,1) Not Null PRIMARY KEY,
41   IDImg Integer Not Null ,
42   Img Image Null ,
43   FOREIGN KEY(IDImg) REFERENCES Individual(ID) on delete cascade
44 );
45
46 CREATE TABLE Camera
47 (
48   CameraID SmallInt Not Null PRIMARY KEY,
49   CameraName Varchar(20) Not Null ,
50   CameraIP Varchar(40) Null ,
51   FPGAIP Varchar(40) Null ,
52   CameraLoc Varchar(40) Null ,
53   CameraDetails Varchar(100) Null ,
54   imgIndex smallint Null ,
55   ThumbPath Varchar(10) Null
56 );
57
58 CREATE TABLE CameraStats
59 (
60   TimeID SmallInt Not Null PRIMARY KEY,
61   CameraID SmallInt Not Null ,
62   LoTimeRange Time(2) Null ,
63   HiTimeRange Time(2) Null ,
64   Simga Float Null ,
65   Mean Float Null ,
66   NValue Integer Default(0) ,
67   FOREIGN KEY(CameraID) REFERENCES Camera(CameraID) on delete cascade
68 );
69
70 CREATE TABLE DetectionEvent

```

```

71 |
72 |     EventID Integer PRIMARY KEY Not Null ,
73 |     EventCamera SmallInt Not Null ,
74 |     EventTimeIn Time(2) Not Null ,
75 |     EventTimeOut Time(2) Null ,
76 |     isChecked Int default(0) ,
77 |     EntryType SmallInt Null ,
78 |     ExitType SmallInt Null ,
79 |     SuspectID Integer Not Null ,
80 |     Confidence Float Not Null ,
81 |     --ImgSize int Null ,
82 |     --ImgXPos int Null ,
83 |     --ImgYPos int Null ,
84 |     FOREIGN KEY(EventCamera) REFERENCES Camera(CameraID) on delete cascade ,
85 |     FOREIGN KEY(SuspectID) REFERENCES Individual(ID) on delete cascade
86 );
87
88 CREATE TABLE Anomaly
89 (
90 |     AnomalyID Integer Identity(1,1) PRIMARY KEY(AnomalyID) ,
91 |     EventID Integer Not Null ,
92 |     Severity Float default(0) ,
93 |     isChecked Smallint default(0) ,
94 |     AnomalyType SmallInt Null ,
95 |     FOREIGN KEY(EventID) REFERENCES DetectionEvent(EventID) on delete cascade
96 );

```

Listing 16.25: Code for Creation of the Database's Tables

16.6 Behavioral Analysis Engine - Primary File

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Data.Linq;
5  using System.Data.Linq.Mapping;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace SDBehavioralAnalysis
10 {
11     class Program
12     {
13         //SeniorDesignCamStatsDataContext SDDatabase;
14         static void Main(string[] args)
15         {
16             SeniorDesignCamStatsDataContext SDDatabase = new
17                 SeniorDesignCamStatsDataContext();
18             DateTime CurrentTime = new DateTime();
19
20             while (SDDatabase.DatabaseExists())
21             {
22                 CurrentTime = DateTime.Now;
23                 var NewInEvents = (from events in SDDatabase.GetTable<DetectionEvent>()
24                     where (events.ExitType == null && events.isChecked == 0) select events
25                     );
26                 var NewOutEvents = (from events in SDDatabase.GetTable<DetectionEvent>()
27                     where (events.ExitType != null && events.EventTimeOut != null &&
28                         events.isChecked == 0) select events);
29                 var CamStats = (from camtables in SDDatabase.GetTable<CameraStat>() where
30                     (camtables.HiTimeRange > CurrentTime.TimeOfDay && CurrentTime.
31                     TimeOfDay > camtables.LoTimeRange) select camtables);
32
33                 foreach (DetectionEvent Event in NewOutEvents)
34             }
35         }
36     }
37 }

```

```

28    {
29
30        if (Event.ExitType == 0 || Event.EntryType == 0)
31            ReportAnomaly(SDDatabase, Event);
32        foreach(CameraStat Camera in CamStats)
33        {
34            if (Camera.CameraID == Event.EventID)
35            {
36                double x = (((TimeSpan)Event.EventTimeOut).TotalMinutes -
37                            Event.EventTimeIn.TotalMinutes);
38                if (Camera.NValue > 36)
39                {
40                    if (Math.Abs(x) > (Camera.Mean + (Camera.Simga * 3)))
41                    {
42                        ReportAnomaly(SDDatabase, Event);
43                        continue;
44                    }
45                    if (checkZVal(Camera, x) > 1.96)
46                        ReportAnomaly(SDDatabase, Event);
47                }
48                UpdateStats(SDDatabase, Camera, Event, x);
49            }
50        } //end of foreach Out
51
52        foreach(DetectionEvent Event in NewInEvents)
53        {
54            foreach(CameraStat Camera in CamStats)
55            {
56                double x = (DateTime.Now.TimeOfDay.TotalMinutes - Event.
57                            EventTimeIn.TotalMinutes);
58                if (Camera.NValue > 36 && checkZVal(Camera, x) > 1.96)
59                {
60                    ReportAnomaly(SDDatabase, Event);
61                    Event.isChecked = 1;
62                    try
63                    {
64                        SDDatabase.SubmitChanges();
65                    }
66                    catch (Exception e)
67                    {
68                        Console.WriteLine("Database Connection Failed, Please
69                                      Check Connection and Restart Program. Details: " + e);
70                    }
71                }
72            }
73        } //end of while
74    } //end of main
75
76    private static double checkZVal(CameraStat CamInfo, double x)
77    {
78        double z = ((x - (double)CamInfo.Mean) / (double)CamInfo.Simga) * Math.Sqrt((
79                                double)CamInfo.NValue);
80        return z;
81    }
82
83    private static void UpdateStats(SeniorDesignCamStatsDataContext SDDatabase,
84                                    CameraStat CameraInfo, DetectionEvent Event, double x)
85    {
86        int NewNVal = (int)CameraInfo.NValue + 1;
87        double NewMean = (((double)CameraInfo.Mean * (NewNVal - 1)) + x) / NewNVal;
88        double NewSigma = Math.Sqrt(((NewNVal - 1) * Math.Pow((int)CameraInfo.Simga
89                                , 2.0)) + Math.Pow((x - NewMean), 2.0)) / NewNVal;
         CameraInfo.NValue = NewNVal;
         CameraInfo.Mean = NewMean;

```

```

90     CameraInfo.Simga = NewSigma;
91     Event.isChecked = 1;
92     try
93     {
94         SDDatabase.SubmitChanges();
95         Console.WriteLine("Camera " + CameraInfo.CameraID + ": Updated with new
96                         values!");
97     }
98     catch (Exception e)
99     {
100        Console.WriteLine("Database Connection Failed, Please Check Connection and
101                         Restart Program. Details: " + e);
102        return;
103    }
104
105    private static void ReportAnomaly(SeniorDesignCamStatsDataContext SDDatabase,
106                                      DetectionEvent Event)
107    {
108        Anomaly newAnom = new Anomaly
109        {
110            EventID = Event.EventID,
111            Severity = 0.0,
112            AnomalyType = null,
113            isChecked = 0
114        };
115
116        SDDatabase.Anomalies.InsertOnSubmit(newAnom);
117        Event.isChecked = 1;
118
119        try
120        {
121            SDDatabase.SubmitChanges();
122            Console.WriteLine("Anomaly Detected !!!!!");
123        }
124        catch (Exception e)
125        {
126            Console.WriteLine("Database Connection Failed, Please Check Connection and
127                         Restart Program. Details: " + e);
128            return;
129        }
130    }
131
132}

```

Listing 16.26: Primary File for the Behavioral Analysis Engine

16.7 Behavioral Analysis Engine - Database Configuration File

```

1 #pragma warning disable 1591
2 //-----
3 // <auto-generated>
4 //   This code was generated by a tool.
5 //   Runtime Version:4.0.30319.34014
6 //
7 //   Changes to this file may cause incorrect behavior and will be lost if
8 //   the code is regenerated.
9 // </auto-generated>
10 //-----
11
12 namespace SDBehavioralAnalysis
13 {

```

```

14  using System.Data.Linq;
15  using System.Data.Linq.Mapping;
16  using System.Data;
17  using System.Collections.Generic;
18  using System.Reflection;
19  using System.Linq;
20  using System.Linq.Expressions;
21  using System.ComponentModel;
22  using System;
23
24
25  [global::System.Data.Linq.Mapping.DatabaseAttribute(Name="SeniorDesignDB2")]
26  public partial class SeniorDesignCamStatsDataContext : System.Data.Linq.DataContext
27  {
28
29      private static System.Data.Linq.Mapping.MappingSource mappingSource = new
30          AttributeMappingSource();
31
32      #region Extensibility Method Definitions
33      partial void OnCreated();
34      partial void InsertCamera(Camera instance);
35      partial void UpdateCamera(Camera instance);
36      partial void DeleteCamera(Camera instance);
37      partial void InsertCameraStat(CameraStat instance);
38      partial void UpdateCameraStat(CameraStat instance);
39      partial void DeleteCameraStat(CameraStat instance);
40      partial void InsertDetectionEvent(DetectionEvent instance);
41      partial void UpdateDetectionEvent(DetectionEvent instance);
42      partial void DeleteDetectionEvent(DetectionEvent instance);
43      partial void InsertAnomaly(Anomaly instance);
44      partial void UpdateAnomaly(Anomaly instance);
45      partial void DeleteAnomaly(Anomaly instance);
46      #endregion
47
48      public SeniorDesignCamStatsDataContext() :
49          base(global::SDBehavioralAnalysis.Properties.Settings.Default.
50              SeniorDesignDB2ConnectionString, mappingSource)
51      {
52          OnCreated();
53      }
54
55      public SeniorDesignCamStatsDataContext(string connection) :
56          base(connection, mappingSource)
57      {
58          OnCreated();
59      }
60
61      public SeniorDesignCamStatsDataContext(System.Data.IDbConnection connection) :
62          base(connection, mappingSource)
63      {
64          OnCreated();
65      }
66
67      public SeniorDesignCamStatsDataContext(string connection, System.Data.Linq.Mapping.
68          MappingSource mappingSource) :
69          base(connection, mappingSource)
70      {
71          OnCreated();
72      }
73
74      public SeniorDesignCamStatsDataContext(System.Data.IDbConnection connection, System.
75          Data.Linq.Mapping.MappingSource mappingSource) :
76          base(connection, mappingSource)
77      {
78          OnCreated();
79      }
80
81      public System.Data.Linq.Table<Camera> Cameras

```

```

78    {
79        get
80        {
81            return this.GetTable<Camera>();
82        }
83    }
84
85    public System.Data.Linq.Table<CameraStat> CameraStats
86    {
87        get
88        {
89            return this.GetTable<CameraStat>();
90        }
91    }
92
93    public System.Data.Linq.Table<DetectionEvent> DetectionEvents
94    {
95        get
96        {
97            return this.GetTable<DetectionEvent>();
98        }
99    }
100
101   public System.Data.Linq.Table<Anomaly> Anomalies
102   {
103       get
104       {
105           return this.GetTable<Anomaly>();
106       }
107   }
108 }
109
110 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Camera")]
111 public partial class Camera : INotifyPropertyChanging, INotifyPropertyChanged
112 {
113
114     private static PropertyChangingEventArgs emptyChangingEventArgs = new
115         PropertyChangingEventArgs(String.Empty);
116
117     private short _CameraID;
118
119     private string _CameraName;
120
121     private string _CameraIP;
122
123     private string _FPGAIP;
124
125     private string _CameraLoc;
126
127     private string _CameraDetails;
128
129     private System.Nullable<short> _imgIndex;
130
131     private string _ThumbPath;
132
133     private EntitySet<CameraStat> _CameraStats;
134
135     private EntitySet<DetectionEvent> _DetectionEvents;
136
137     #region Extensibility Method Definitions
138     partial void OnLoaded();
139     partial void OnValidate(System.Data.Linq.ChangeAction action);
140     partial void OnCreated();
141     partial void OnCameraIDChanging(short value);
142     partial void OnCameraIDChanged();
143     partial void OnCameraNameChanging(string value);
144     partial void OnCameraNameChanged();
145     partial void OnCameraIPChanging(string value);

```

```

145    partial void OnCameraIPChanged();
146    partial void OnFPGAIPChanging(string value);
147    partial void OnFPGAIPChanged();
148    partial void OnCameraLocChanging(string value);
149    partial void OnCameraLocChanged();
150    partial void OnCameraDetailsChanging(string value);
151    partial void OnCameraDetailsChanged();
152    partial void OnimgIndexChanging(System.Nullable<short> value);
153    partial void OnimgIndexChanged();
154    partial void OnThumbPathChanging(string value);
155    partial void OnThumbPathChanged();
156 #endregion
157
158    public Camera()
159    {
160        this._CameraStats = new EntitySet<CameraStat>(new Action<CameraStat>(this.
161            attach_CameraStats), new Action<CameraStat>(this.detach_CameraStats));
162        this._DetectionEvents = new EntitySet<DetectionEvent>(new Action<DetectionEvent>(
163            this.attach_DetectionEvents), new Action<DetectionEvent>(this.
164            detach_DetectionEvents));
165        OnCreated();
166    }
167
168    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="`_CameraID", DbType="
169        SmallInt NOT NULL", IsPrimaryKey=true)]
170    public short CameraID
171    {
172        get
173        {
174            return this._CameraID;
175        }
176        set
177        {
178            if ((this._CameraID != value))
179            {
180                this.OnCameraIDChanging(value);
181                this.SendPropertyChanging();
182                this._CameraID = value;
183                this.SendPropertyChanged("CameraID");
184                this.OnCameraIDChanged();
185            }
186        }
187    }
188
189    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="`_CameraName", DbType="
190        VarChar(20) NOT NULL", CanBeNull=false)]
191    public string CameraName
192    {
193        get
194        {
195            return this._CameraName;
196        }
197        set
198        {
199            if ((this._CameraName != value))
200            {
201                this.OnCameraNameChanging(value);
202                this.SendPropertyChanging();
203                this._CameraName = value;
204                this.SendPropertyChanged("CameraName");
205                this.OnCameraNameChanged();
206            }
207        }
208    }
209
210    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="`_CameraIP", DbType="VarChar
211        (40)")]
212    public string CameraIP

```

```

207    {
208        get
209        {
210            return this._CameraIP;
211        }
212        set
213        {
214            if ((this._CameraIP != value))
215            {
216                this.OnCameraIPChanging(value);
217                this.SendPropertyChanging();
218                this._CameraIP = value;
219                this.SendPropertyChanged("CameraIP");
220                this.OnCameraIPChanged();
221            }
222        }
223    }
224
225    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FPGAIP", DbType="VarChar
226    (40)")]
226    public string FPGAIP
227    {
228        get
229        {
230            return this._FPGAIP;
231        }
232        set
233        {
234            if ((this._FPGAIP != value))
235            {
236                this.OnFPGAIPChanging(value);
237                this.SendPropertyChanging();
238                this._FPGAIP = value;
239                this.SendPropertyChanged("FPGAIP");
240                this.OnFPGAIPChanged();
241            }
242        }
243    }
244
245    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraLoc", DbType="
246    VarChar(40)")]
246    public string CameraLoc
247    {
248        get
249        {
250            return this._CameraLoc;
251        }
252        set
253        {
254            if ((this._CameraLoc != value))
255            {
256                this.OnCameraLocChanging(value);
257                this.SendPropertyChanging();
258                this._CameraLoc = value;
259                this.SendPropertyChanged("CameraLoc");
260                this.OnCameraLocChanged();
261            }
262        }
263    }
264
265    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraDetails", DbType="
266    VarChar(100)")]
266    public string CameraDetails
267    {
268        get
269        {
270            return this._CameraDetails;
271        }

```

```

272     set
273     {
274         if ((this._CameraDetails != value))
275         {
276             this.OnCameraDetailsChanging(value);
277             this.SendPropertyChanging();
278             this._CameraDetails = value;
279             this.SendPropertyChanged("CameraDetails");
280             this.OnCameraDetailsChanged();
281         }
282     }
283 }
284
285 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_imgIndex", DbType="SmallInt")]
286 public System.Nullable<short> imgIndex
287 {
288     get
289     {
290         return this._imgIndex;
291     }
292     set
293     {
294         if ((this._imgIndex != value))
295         {
296             this.OnimgIndexChanging(value);
297             this.SendPropertyChanging();
298             this._imgIndex = value;
299             this.SendPropertyChanged("imgIndex");
300             this.OnimgIndexChanged();
301         }
302     }
303 }
304
305 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ThumbPath", DbType="VarChar(10)")]
306 public string ThumbPath
307 {
308     get
309     {
310         return this._ThumbPath;
311     }
312     set
313     {
314         if ((this._ThumbPath != value))
315         {
316             this.OnThumbPathChanging(value);
317             this.SendPropertyChanging();
318             this._ThumbPath = value;
319             this.SendPropertyChanged("ThumbPath");
320             this.OnThumbPathChanged();
321         }
322     }
323 }
324
325 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat",
326             Storage="_CameraStats", ThisKey="CameraID", OtherKey="CameraID")]
327 public EntitySet<CameraStat> CameraStats
328 {
329     get
330     {
331         return this._CameraStats;
332     }
333     set
334     {
335         this._CameraStats.Assign(value);
336     }
337 }
```

```

337 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_DetectionEvent",
338     Storage="_DetectionEvents", ThisKey="CameraID", OtherKey="EventCamera")]
339 public EntitySet<DetectionEvent> DetectionEvents
340 {
341     get
342     {
343         return this._DetectionEvents;
344     }
345     set
346     {
347         this._DetectionEvents.Assign(value);
348     }
349 }
350
351 public event PropertyChangingEventHandler PropertyChanging;
352
353 public event PropertyChangedEventHandler PropertyChanged;
354
355 protected virtual void SendPropertyChanging()
356 {
357     if ((this.PropertyChanging != null))
358     {
359         this.PropertyChanging(this, emptyChangingEventArgs);
360     }
361 }
362
363 protected virtual void SendPropertyChanged(String propertyName)
364 {
365     if ((this.PropertyChanged != null))
366     {
367         this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
368     }
369 }
370
371 private void attach_CameraStats(CameraStat entity)
372 {
373     this.SendPropertyChanging();
374     entity.Camera = this;
375 }
376
377 private void detach_CameraStats(CameraStat entity)
378 {
379     this.SendPropertyChanging();
380     entity.Camera = null;
381 }
382
383 private void attach_DetectionEvents(DetectionEvent entity)
384 {
385     this.SendPropertyChanging();
386     entity.Camera = this;
387 }
388
389 private void detach_DetectionEvents(DetectionEvent entity)
390 {
391     this.SendPropertyChanging();
392     entity.Camera = null;
393 }
394 }

395 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.CameraStats")]
396 public partial class CameraStat : INotifyPropertyChanging, INotifyPropertyChanged
397 {
398
399     private static PropertyChangingEventArgs emptyChangingEventArgs = new
400         PropertyChangingEventArgs(String.Empty);
401
402     private short _TimeID;

```

```

403
404     private short _CameraID;
405
406     private System.Nullable<System.TimeSpan> _LoTimeRange;
407
408     private System.Nullable<System.TimeSpan> _HiTimeRange;
409
410     private System.Nullable<double> _Simga;
411
412     private System.Nullable<double> _Mean;
413
414     private System.Nullable<int> _NValue;
415
416     private EntityRef<Camera> _Camera;
417
418 #region Extensibility Method Definitions
419     partial void OnLoaded();
420     partial void OnValidate(System.Data.Linq.ChangeAction action);
421     partial void OnCreated();
422     partial void OnTimeIDChanging(short value);
423     partial void OnTimeIDChanged();
424     partial void OnCameraIDChanging(short value);
425     partial void OnCameraIDChanged();
426     partial void OnLoTimeRangeChanging(System.Nullable<System.TimeSpan> value);
427     partial void OnLoTimeRangeChanged();
428     partial void OnHiTimeRangeChanging(System.Nullable<System.TimeSpan> value);
429     partial void OnHiTimeRangeChanged();
430     partial void OnSimgaChanging(System.Nullable<double> value);
431     partial void OnSimgaChanged();
432     partial void OnMeanChanging(System.Nullable<double> value);
433     partial void OnMeanChanged();
434     partial void OnNValueChanging(System.Nullable<int> value);
435     partial void OnNValueChanged();
436 #endregion
437
438     public CameraStat()
439     {
440         this._Camera = default(EntityRef<Camera>);
441         OnCreated();
442     }
443
444 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="TimeID", DbType="SmallInt
445     NOT NULL", IsPrimaryKey=true)]
446     public short TimeID
447     {
448         get
449         {
450             return this._TimeID;
451         }
452         set
453         {
454             if ((this._TimeID != value))
455             {
456                 this.OnTimeIDChanging(value);
457                 this.SendPropertyChanging();
458                 this._TimeID = value;
459                 this.SendPropertyChanged("TimeID");
460                 this.OnTimeIDChanged();
461             }
462         }
463     }
464
465 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="CameraID", DbType="
466     SmallInt NOT NULL")]
467     public short CameraID
468     {
469         get
470         {

```

```

469     return this._CameraID;
470 }
471 set
472 {
473     if ((this._CameraID != value))
474     {
475         if (this.Camera.HasLoadedOrAssignedValue)
476         {
477             throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
478         }
479         this.OnCameraIDChanging(value);
480         this.SendPropertyChanging();
481         this._CameraID = value;
482         this.SendPropertyChanged("CameraID");
483         this.OnCameraIDChanged();
484     }
485 }
486 }
487
488 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="LoTimeRange", DbType="Time
    ")]
489 public System.Nullable<System.TimeSpan> LoTimeRange
490 {
491     get
492     {
493         return this._LoTimeRange;
494     }
495     set
496     {
497         if ((this._LoTimeRange != value))
498         {
499             this.OnLoTimeRangeChanging(value);
500             this.SendPropertyChanging();
501             this._LoTimeRange = value;
502             this.SendPropertyChanged("LoTimeRange");
503             this.OnLoTimeRangeChanged();
504         }
505     }
506 }
507
508 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="HiTimeRange", DbType="Time
    ")]
509 public System.Nullable<System.TimeSpan> HiTimeRange
510 {
511     get
512     {
513         return this._HiTimeRange;
514     }
515     set
516     {
517         if ((this._HiTimeRange != value))
518         {
519             this.OnHiTimeRangeChanging(value);
520             this.SendPropertyChanging();
521             this._HiTimeRange = value;
522             this.SendPropertyChanged("HiTimeRange");
523             this.OnHiTimeRangeChanged();
524         }
525     }
526 }
527
528 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="Simga", DbType="Float")]
529 public System.Nullable<double> Simga
530 {
531     get
532     {
533         return this._Simga;
534     }

```

```

535     set
536     {
537         if ((this._Simga != value))
538         {
539             this.OnSimgaChanging(value);
540             this.SendPropertyChanging();
541             this._Simga = value;
542             this.SendPropertyChanged("Simga");
543             this.OnSimgaChanged();
544         }
545     }
546 }
547
548 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="Mean", DbType="Float")]
549 public System.Nullable<double> Mean
550 {
551     get
552     {
553         return this._Mean;
554     }
555     set
556     {
557         if ((this._Mean != value))
558         {
559             this.OnMeanChanging(value);
560             this.SendPropertyChanging();
561             this._Mean = value;
562             this.SendPropertyChanged("Mean");
563             this.OnMeanChanged();
564         }
565     }
566 }
567
568 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="NValue", DbType="Int")]
569 public System.Nullable<int> NValue
570 {
571     get
572     {
573         return this._NValue;
574     }
575     set
576     {
577         if ((this._NValue != value))
578         {
579             this.OnNValueChanging(value);
580             this.SendPropertyChanging();
581             this._NValue = value;
582             this.SendPropertyChanged("NValue");
583             this.OnNValueChanged();
584         }
585     }
586 }
587
588 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat",
589             Storage="_Camera", ThisKey="CameraID", OtherKey="CameraID", IsForeignKey=true,
590             DeleteOnNull=true, DeleteRule="CASCADE")]
591 public Camera Camera
592 {
593     get
594     {
595         return this._Camera.Entity;
596     }
597     set
598     {
599         Camera previousValue = this._Camera.Entity;
600         if (((previousValue != value)
601             || (this._Camera.HasLoadedOrAssignedValue == false)))
602         {

```

```

601     this .SendPropertyChanging();
602     if ((previousValue != null))
603     {
604         this ._Camera.Entity = null;
605         previousValue.CameraStats.Remove(this );
606     }
607     this ._Camera.Entity = value;
608     if ((value != null))
609     {
610         value.CameraStats.Add(this );
611         this ._CameraID = value.CameraID;
612     }
613     else
614     {
615         this ._CameraID = default(short);
616     }
617     this .SendPropertyChanged("Camera");
618 }
619 }
620 }
621
622 public event PropertyChangingEventHandler PropertyChanging;
623
624 public event PropertyChangedEventHandler PropertyChanged;
625
626 protected virtual void SendPropertyChanging()
627 {
628     if ((this .PropertyChanging != null))
629     {
630         this .PropertyChanging(this , emptyChangingEventArgs);
631     }
632 }
633
634 protected virtual void SendPropertyChanged(String propertyName)
635 {
636     if ((this .PropertyChanged != null))
637     {
638         this .PropertyChanged(this , new PropertyChangedEventArgs(propertyName));
639     }
640 }
641 }
642
643 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.DetectionEvent")]
644 public partial class DetectionEvent : INotifyPropertyChanging, INotifyPropertyChanged
645 {
646
647     private static PropertyChangingEventArgs emptyChangingEventArgs = new
648     PropertyChangingEventArgs(String.Empty);
649
650     private int _EventID;
651
652     private short _EventCamera;
653
654     private System.TimeSpan _EventTimeIn;
655
656     private System.Nullable<System.TimeSpan> _EventTimeOut;
657
658     private System.Nullable<int> _isChecked;
659
660     private System.Nullable<short> _EntryType;
661
662     private System.Nullable<short> _ExitType;
663
664     private int _SuspectID;
665
666     private double _Confidence;
667
668     private EntitySet<Anomaly> _Anomalies;

```

```

668
669     private EntityRef<Camera> _Camera;
670
671 #region Extensibility Method Definitions
672     partial void OnLoaded();
673     partial void OnValidate(System.Data.Linq.ChangeAction action);
674     partial void OnCreated();
675     partial void OnEventIDChanging(int value);
676     partial void OnEventIDChanged();
677     partial void OnEventCameraChanging(short value);
678     partial void OnEventCameraChanged();
679     partial void OnEventTimeInChanging(System.TimeSpan value);
680     partial void OnEventTimeInChanged();
681     partial void OnEventTimeOutChanging(System.Nullable<System.TimeSpan> value);
682     partial void OnEventTimeOutChanged();
683     partial void OnIsCheckedChanging(System.Nullable<int> value);
684     partial void OnIsCheckedChanged();
685     partial void OnEntryTypeChanging(System.Nullable<short> value);
686     partial void OnEntryTypeChanged();
687     partial void OnExitTypeChanging(System.Nullable<short> value);
688     partial void OnExitTypeChanged();
689     partial void OnSuspectIDChanging(int value);
690     partial void OnSuspectIDChanged();
691     partial void OnConfidenceChanging(double value);
692     partial void OnConfidenceChanged();
693 #endregion
694
695     public DetectionEvent()
696     {
697         this._Anomalies = new EntitySet<Anomaly>(new Action<Anomaly>(this.attach_Anomalies),
698             new Action<Anomaly>(this.detach_Anomalies));
699         this._Camera = default(EntityRef<Camera>);
700         OnCreated();
701     }
702
703 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EventID", DbType="Int NOT
704     NULL", IsPrimaryKey=true)]
705     public int EventID
706     {
707         get
708         {
709             return this._EventID;
710         }
711         set
712         {
713             if ((this._EventID != value))
714             {
715                 this.OnEventIDChanging(value);
716                 this.SendPropertyChanging();
717                 this._EventID = value;
718                 this.SendPropertyChanged("EventID");
719                 this.OnEventIDChanged();
720             }
721         }
722
723 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EventCamera", DbType="
724     SmallInt NOT NULL")]
725     public short EventCamera
726     {
727         get
728         {
729             return this._EventCamera;
730         }
731         set
732         {
733             if ((this._EventCamera != value))
734             {

```

```

733         if (this._Camera.HasLoadedOrAssignedValue)
734         {
735             throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
736         }
737         this.OnEventCameraChanging(value);
738         this.SendPropertyChanging();
739         this._EventCamera = value;
740         this.SendPropertyChanged("EventCamera");
741         this.OnEventCameraChanged();
742     }
743 }
744 }
745
746 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EventTimeIn", DbType="Time
    NOT NULL")]
747 public System.TimeSpan EventTimeIn
748 {
749     get
750     {
751         return this._EventTimeIn;
752     }
753     set
754     {
755         if ((this._EventTimeIn != value))
756         {
757             this.OnEventTimeInChanging(value);
758             this.SendPropertyChanging();
759             this._EventTimeIn = value;
760             this.SendPropertyChanged("EventTimeIn");
761             this.OnEventTimeInChanged();
762         }
763     }
764 }
765
766 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EventTimeOut", DbType="
    Time")]
767 public System.Nullable<System.TimeSpan> EventTimeOut
768 {
769     get
770     {
771         return this._EventTimeOut;
772     }
773     set
774     {
775         if ((this._EventTimeOut != value))
776         {
777             this.OnEventTimeOutChanging(value);
778             this.SendPropertyChanging();
779             this._EventTimeOut = value;
780             this.SendPropertyChanged("EventTimeOut");
781             this.OnEventTimeOutChanged();
782         }
783     }
784 }
785
786 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_isChecked", DbType="Int")]
787 public System.Nullable<int> isChecked
788 {
789     get
790     {
791         return this._isChecked;
792     }
793     set
794     {
795         if ((this._isChecked != value))
796         {
797             this.OnisCheckedChanging(value);
798             this.SendPropertyChanging();

```

```

799         this._isChecked = value;
800         this.SendPropertyChanged("isChecked");
801         this.OnisCheckedChanged();
802     }
803 }
804 }
805
806 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EntryType", DbType="SmallInt")]
807 public System.Nullable<short> EntryType
808 {
809     get
810     {
811         return this._EntryType;
812     }
813     set
814     {
815         if ((this._EntryType != value))
816         {
817             this.OnEntryTypeChanging(value);
818             this.SendPropertyChanging();
819             this._EntryType = value;
820             this.SendPropertyChanged("EntryType");
821             this.OnEntryTypeChanged();
822         }
823     }
824 }
825
826 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="ExitType", DbType="SmallInt")]
827 public System.Nullable<short> ExitType
828 {
829     get
830     {
831         return this._ExitType;
832     }
833     set
834     {
835         if ((this._ExitType != value))
836         {
837             this.OnExitTypeChanging(value);
838             this.SendPropertyChanging();
839             this._ExitType = value;
840             this.SendPropertyChanged("ExitType");
841             this.OnExitTypeChanged();
842         }
843     }
844 }
845
846 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="SuspectID", DbType="Int NOT NULL")]
847 public int SuspectID
848 {
849     get
850     {
851         return this._SuspectID;
852     }
853     set
854     {
855         if ((this._SuspectID != value))
856         {
857             this.OnSuspectIDChanging(value);
858             this.SendPropertyChanging();
859             this._SuspectID = value;
860             this.SendPropertyChanged("SuspectID");
861             this.OnSuspectIDChanged();
862         }
863     }

```

```

864    }
865
866    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="`_Confidence`, DbType="Float
867      NOT NULL")]
867    public double Confidence
868    {
869      get
870      {
871        return this._Confidence;
872      }
873      set
874      {
875        if ((this._Confidence != value))
876        {
877          this.OnConfidenceChanging(value);
878          this.SendPropertyChanging();
879          this._Confidence = value;
880          this.SendPropertyChanged("Confidence");
881          this.OnConfidenceChanged();
882        }
883      }
884    }
885
886    [global::System.Data.Linq.Mapping.AssociationAttribute(Name="DetectionEvent_Anomaly",
887      Storage="_Anomalies", ThisKey="EventID", OtherKey="EventID")]
887    public EntitySet<Anomaly> Anomalies
888    {
889      get
890      {
891        return this._Anomalies;
892      }
893      set
894      {
895        this._Anomalies.Assign(value);
896      }
897    }
898
899    [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_DetectionEvent",
900      Storage="_Camera", ThisKey="EventCamera", OtherKey="CameraID", IsForeignKey=true,
901      DeleteOnNull=true, DeleteRule="CASCADE")]
900    public Camera Camera
901    {
902      get
903      {
904        return this._Camera.Entity;
905      }
906      set
907      {
908        Camera previousValue = this._Camera.Entity;
909        if (((previousValue != value)
910            || (this._Camera.HasLoadedOrAssignedValue == false)))
911        {
912          this.SendPropertyChanging();
913          if ((previousValue != null))
914          {
915            this._Camera.Entity = null;
916            previousValue.DetectionEvents.Remove(this);
917          }
918          this._Camera.Entity = value;
919          if ((value != null))
920          {
921            value.DetectionEvents.Add(this);
922            this._EventCamera = value.CameraID;
923          }
924        }
925        else
926        {
927          this._EventCamera = default(short);
928        }

```

```

928         this .SendPropertyChanged("Camera");
929     }
930   }
931 }
932
933 public event PropertyChangingEventHandler PropertyChanging;
934
935 public event PropertyChangedEventHandler PropertyChanged;
936
937 protected virtual void SendPropertyChanging()
938 {
939   if ((this .PropertyChanging != null))
940   {
941     this .PropertyChanging(this , emptyChangingEventArgs);
942   }
943 }
944
945 protected virtual void SendPropertyChanged(String propertyName)
946 {
947   if ((this .PropertyChanged != null))
948   {
949     this .PropertyChanged(this , new PropertyChangedEventArgs(propertyName));
950   }
951 }
952
953 private void attach_Anomalies(Anomaly entity)
954 {
955   this .SendPropertyChanging();
956   entity.DetectionEvent = this ;
957 }
958
959 private void detach_Anomalies(Anomaly entity)
960 {
961   this .SendPropertyChanging();
962   entity.DetectionEvent = null;
963 }
964 }
965
966 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Anomaly")]
967 public partial class Anomaly : INotifyPropertyChanging, INotifyPropertyChanged
968 {
969
970   private static PropertyChangingEventArgs emptyChangingEventArgs = new
971   PropertyChangingEventArgs(String.Empty);
972
973   private int _AnomalyID;
974
975   private int _EventID;
976
977   private System.Nullable<double> _Severity;
978
979   private System.Nullable<short> _isChecked;
980
981   private System.Nullable<short> _AnomalyType;
982
983   private EntityRef<DetectionEvent> _DetectionEvent;
984
985   #region Extensibility Method Definitions
986   partial void OnLoaded();
987   partial void OnValidate(System.Data.Linq.ChangeAction action);
988   partial void OnCreated();
989   partial void OnAnomalyIDChanging(int value);
990   partial void OnAnomalyIDChanged();
991   partial void OnEventIDChanging(int value);
992   partial void OnEventIDChanged();
993   partial void OnSeverityChanging(System.Nullable<double> value);
994   partial void OnSeverityChanged();
995   partial void OnisCheckedChanging(System.Nullable<short> value);

```

```

995     partial void OnIsCheckedChanged();
996     partial void OnAnomalyTypeChanging(System.Nullable<short> value);
997     partial void OnAnomalyTypeChanged();
998 #endregion
999
1000    public Anomaly()
1001    {
1002        this._DetectionEvent = default(EntityRef<DetectionEvent>);
1003        OnCreated();
1004    }
1005
1006    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="AnomalyID", AutoSync=
1007        AutoSync.OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true,
1008        IsDbGenerated=true)]
1009    public int AnomalyID
1010    {
1011        get
1012        {
1013            return this._AnomalyID;
1014        }
1015        set
1016        {
1017            if ((this._AnomalyID != value))
1018            {
1019                this.OnAnomalyIDChanging(value);
1020                this.SendPropertyChanging();
1021                this._AnomalyID = value;
1022                this.SendPropertyChanged("AnomalyID");
1023                this.OnAnomalyIDChanged();
1024            }
1025        }
1026
1027    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EventID", DbType="Int NOT
1028        NULL")]
1029    public int EventID
1030    {
1031        get
1032        {
1033            return this._EventID;
1034        }
1035        set
1036        {
1037            if ((this._EventID != value))
1038            {
1039                if (this._DetectionEvent.HasLoadedOrAssignedValue)
1040                {
1041                    throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
1042                }
1043                this.OnEventIDChanging(value);
1044                this.SendPropertyChanging();
1045                this._EventID = value;
1046                this.SendPropertyChanged("EventID");
1047                this.OnEventIDChanged();
1048            }
1049        }
1050
1051    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="Severity", DbType="Float")]
1052    public System.Nullable<double> Severity
1053    {
1054        get
1055        {
1056            return this._Severity;
1057        }
1058        set
1059        {
1060

```

```

1059     if ((this._Severity != value))
1060     {
1061         this.OnSeverityChanging(value);
1062         this.SendPropertyChanging();
1063         this._Severity = value;
1064         this.SendPropertyChanged("Severity");
1065         this.OnSeverityChanged();
1066     }
1067 }
1068 }
1069
1070 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_isChecked", DbType="SmallInt")]
1071 public System.Nullable<short> isChecked
1072 {
1073     get
1074     {
1075         return this._isChecked;
1076     }
1077     set
1078     {
1079         if ((this._isChecked != value))
1080         {
1081             this.OnisCheckedChanging(value);
1082             this.SendPropertyChanging();
1083             this._isChecked = value;
1084             this.SendPropertyChanged("isChecked");
1085             this.OnisCheckedChanged();
1086         }
1087     }
1088 }
1089
1090 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_AnomalyType", DbType="SmallInt")]
1091 public System.Nullable<short> AnomalyType
1092 {
1093     get
1094     {
1095         return this._AnomalyType;
1096     }
1097     set
1098     {
1099         if ((this._AnomalyType != value))
1100         {
1101             this.OnAnomalyTypeChanging(value);
1102             this.SendPropertyChanging();
1103             this._AnomalyType = value;
1104             this.SendPropertyChanged("AnomalyType");
1105             this.OnAnomalyTypeChanged();
1106         }
1107     }
1108 }
1109
1110 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="DetectionEvent_Anomaly", Storage="_DetectionEvent", ThisKey="EventID", OtherKey="EventID", IsForeignKey=true, DeleteOnNull=true, DeleteRule="CASCADE")]
1111 public DetectionEvent DetectionEvent
1112 {
1113     get
1114     {
1115         return this._DetectionEvent.Entity;
1116     }
1117     set
1118     {
1119         DetectionEvent previousValue = this._DetectionEvent.Entity;
1120         if (((previousValue != value)
1121             || (this._DetectionEvent.HasLoadedOrAssignedValue == false)))
1122         {

```

```

1123     this .SendPropertyChanging();
1124     if ((previousValue != null))
1125     {
1126         this ._DetectionEvent.Entity = null;
1127         previousValue.Anomalies.Remove(this );
1128     }
1129     this ._DetectionEvent.Entity = value;
1130     if ((value != null))
1131     {
1132         value.Anomalies.Add(this );
1133         this ._EventID = value.EventID;
1134     }
1135     else
1136     {
1137         this ._EventID = default(int);
1138     }
1139     this .SendPropertyChanged("DetectionEvent");
1140 }
1141 }
1142 }
1143
1144 public event PropertyChangingEventHandler PropertyChanging;
1145
1146 public event PropertyChangedEventHandler PropertyChanged;
1147
1148 protected virtual void SendPropertyChanging()
1149 {
1150     if ((this .PropertyChanging != null))
1151     {
1152         this .PropertyChanging(this , emptyChangingEventArgs);
1153     }
1154 }
1155
1156 protected virtual void SendPropertyChanged(String propertyName)
1157 {
1158     if ((this .PropertyChanged != null))
1159     {
1160         this .PropertyChanged(this , new PropertyChangedEventArgs(propertyName));
1161     }
1162 }
1163 }
1164 #pragma warning restore 1591

```

Listing 16.27: Database Configuration File

16.8 User Interface - Primary Form

```

1 using System;
2 using System.Collections.Generic;
3 using System.Drawing;
4 using System.Windows.Forms;
5 using System.Linq;
6 using System.Data.Linq;
7 using System.Data.Linq.Mapping;
8 using System.Net;
9 using System.IO;
10 using System.Threading;
11 using System.ComponentModel;
12
13
14 namespace Senior_Design_UI
15 {
16     public partial class Form1 : Form
17     {

```

```

18     Form StreamAddForm;
19     delegate void SetTextCallback(string text);
20     delegate void SetUICallback();
21     System.Threading.Thread DBThread;
22
23
24     /****** LOADING FUNCTIONS
25      *****/
26     public Form1()
27     {
28         InitializeComponent();
29     }
30
31     private void Form1_Load(object sender, EventArgs e)
32     {
33         /* Basic Options and links being established */
34         StreamListView.SmallImageList = Globals.StreamThumbsList;
35         ObjectListView.SmallImageList = Globals.EventThumbsList;
36         Globals.StreamThumbsList.ImageSize = new Size(95,60);
37         Globals.EventThumbsList.ImageSize = new Size(95, 60);
38         VideoStream.CtlVisible = false;
39         VideoStream.Update();
40         DBThread = new System.Threading.Thread(new System.Threading.ThreadStart(
41             StartDBCalls));
42         DBThread.Start();
43
44         /* Setting up resizing options and different screen options */
45         StreamListMenu.Size = new Size(StreamListMenu.Size.Width, (RightMenu.Size.
46             Height / 2) - 12);
47         ObjectList.Size = new Size(ObjectList.Size.Width, (RightMenu.Size.Height / 2
48             - 12));
49         NoteLog.Location = new Point(NoteLog.Location.X, (RightMenu.Size.Height +
50             menuStrip1.Size.Height - NoteLog.Size.Height) - 5);
51         VideoStream.Size = new Size( RightMenu.Location.X - 24, NoteLog.Location.Y -
52             12 - menuStrip1.Size.Height );
53         Identity.Location = new Point(NoteLog.Size.Width + NoteLog.Location.X + 12,
54             NoteLog.Location.Y);
55         if (RightMenu.Location.X > 800)
56         {
57             NoteLog.Size = new Size((RightMenu.Location.X - Identity.Size.Width),
58                 NoteLog.Size.Height);
59             Identity.Size = new Size((RightMenu.Location.X - NoteLog.Size.Width) - 36
60                 , Identity.Size.Height);
61             Identity.Location = new Point(NoteLog.Size.Width + NoteLog.Location.X +
62                 12, Identity.Location.Y);
63             Identity.Show();
64         }
65         else
66         {
67             NoteLog.Size = new Size(RightMenu.Location.X - 12, NoteLog.Size.Height);
68             Identity.Hide();
69         }
70         LoadDBCams();
71     }
72
73     private void Form1_FormClosing(object sender, FormClosingEventArgs e)
74     {
75         DBThread.Abort();
76     }
77
78     private void LoadDBCams()
79     {
80         SDDBDDataContext DB = new SDDBDDataContext();
81
82         var Cameras = (from cams in DB.GetTable<Camera>() select cams);
83
84         foreach (Camera DBCam in Cameras)
85         {

```

```

76         try
77     {
78         Image NewThumb = Image.FromFile(DBCam.ThumbPath);
79         Globals.StreamThumbsList.Images.Add(NewThumb);
80     }
81     catch
82     {
83         Image FailedImg = Image.FromFile("C:/Users/nfox/Source/Repos/
84             SmartCameras/Senior Design UI/Senior Design UI/noImageAvailable.
85             jpg");
86         Globals.StreamThumbsList.Images.Add(FailedImg);
87     }
88
89     RSTPStream CamFromDB = new RSTPStream(DBCam.CameraIP, DBCam.CameraDetails,
90                                         DBCam.CameraLoc, Globals.StreamList.Count + 1);
91     Globals.StreamList.Add(CamFromDB);
92
93     UpdateStreamListBox();
94 }
95
96
97 /* **** Event Functions **** */
98
99 private void addStreamToolStripMenuItem_Click(object sender, EventArgs e)
100 {
101     StreamAddForm = new AddStream();
102     StreamAddForm.ShowDialog();
103     if (StreamAddForm.DialogResult == DialogResult.OK)
104     {
105         label1.Text = "Start" + Globals.StreamList.Count + " : ";
106         StreamThumbRefresh(Globals.StreamList[Globals.StreamList.Count - 1]);
107     }
108 }
109
110 private void StreamThumbRefresh(RSTPStream thisStream)
111 {
112
113     //Image Newthumb = Image.FromFile();
114     //Globals.StreamThumbsList.Images[thisStream.get_imgIndex()] = NewThumb;
115
116     /* Not Working Yet
117     Uri ThumbUrl = new Uri(thisStream.get_url() + "/dsm?");
118     WebRequest ThumbNailRequest = WebRequest.Create(ThumbUrl);
119     WebResponse ThumbNailReponse = ThumbNailRequest.GetResponse();
120
121     if (ThumbNailReponse.ResponseUri.AbsoluteUri != null)
122     {
123         Image NewThumb = Image.FromStream(ThumbNailReponse.GetResponseStream());
124         Globals.StreamThumbsList.Images[thisStream.get_imgIndex()] = NewThumb;
125     }*/
126
127     UpdateStreamListBox();
128 }
129
130 private void UpdateStreamListBox()
131 {
132     for (int i = 0; i < Globals.StreamList.Count; i++)
133     {
134         ListViewItem thisColumn = new ListViewItem("", i);
135         thisColumn.SubItems.Add(Globals.StreamList[i].get_loc());
136         thisColumn.SubItems.Add(Globals.StreamList[i].get_details());
137         try
138         {
139             StreamListView.Items[i] = thisColumn;
140         }
141         catch (ArgumentOutOfRangeException e)
142         {
143             StreamListView.Items.Add(thisColumn);
144         }
145     }
146 }

```

```

140     }
141 }
142 }
143
144 private void Form1_ResizeBegin(object sender, EventArgs e)
145 {
146     StreamListMenu.Size = new Size(StreamListMenu.Size.Width, (RightMenu.Size.
147         Height / 2) - 12);
148     ObjectList.Size = new Size(ObjectList.Size.Width, (RightMenu.Size.Height / 2
149         - 12));
150     NoteLog.Location = new Point(NoteLog.Location.X, (RightMenu.Size.Height +
151         menuStrip1.Size.Height - NoteLog.Size.Height) - 5);
152     if (RightMenu.Location.X > 800)
153     {
154         NoteLog.Size = new Size((RightMenu.Location.X - Identity.Size.Width) - 36,
155             NoteLog.Size.Height);
156         Identity.Size = new Size(300, Identity.Size.Height);
157         Identity.Location = new Point(NoteLog.Size.Width + NoteLog.Location.X +
158             12, Identity.Location.Y);
159         Identity.Show();
160     }
161 else
162 {
163     NoteLog.Size = new Size(RightMenu.Location.X - 12, NoteLog.Size.Height);
164     Identity.Hide();
165 }
166 }
167
168
169 private void refreshIconsToolStripMenuItem_Click(object sender, EventArgs e)
170 {
171     UpdateStreamListBox();
172 }
173
174
175 private void StreamListView_SelectedIndexChanged(object sender, EventArgs e)
176 {
177     ListView.SelectedListViewItemCollection indexes = this.StreamListView.
178         SelectedItems;
179     int index = 0;
180
181     foreach (ListViewItem item in indexes)
182     {
183         if (item.Selected)
184         {
185             index = item.Index;
186             break;
187         }
188
189         String VideoUrl = "rtsp://" + Globals.StreamList[index].get_url() + "/live1.
190             sdp";
191         VideoStream.playlist.add(VideoUrl);
192         VideoStream.playlist.next();
193         VideoStream.playlist.play();
194     }
195
196
197 /**
198 **** DATABASE FUNCTIONS ****
199 */
199 // Database Continous Calls checking for anomalies
200 private void StartDBCalls()
201 {
202     SDDBDataContext SDDatabase = new SDDBDataContext();
203     DateTime CurrTime = DateTime.Now;
204
205     while (SDDatabase.DatabaseExists())
206     {
207         if (CurrTime <= DateTime.Now)
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999

```

```

200    {
201        // SafeSetText(DateTime.Now.TimeOfDay + "Queries started");
202        UpdateAnoms(SDDatabase);
203        UpdateEvents(SDDatabase);
204        CurrTime = CurrTime.AddMilliseconds(1500);
205        // SafeSetText( DateTime.Now.TimeOfDay + "Queries completed");
206    }
207    else
208    {
209        System.Threading.Thread.Sleep(500);
210    }
211}
212}
213
214 private void UpdateAnoms(SDDBDataContext SDDB)
215{
216    var newAnomalies = (from anom in SDDB.GetTable<Anomaly>() where (anom.
217        isChecked == 0) select anom);
218    String LogOutput = "";
219
220    foreach (Anomaly anom in newAnomalies)
221    {
222        try
223        {
224            anom.isChecked = 1;
225            SDDB.SubmitChanges();
226            var eventID = SDDB.DetectionEvents.Single(x => x.EventID == anom.
227                EventID);
228            LogOutput = DateTime.Now.TimeOfDay + ": Anomaly Detected in Camera: "
229                + SDDB.Cameras.Single( x => x.CameraID == eventID.Camera.CameraID)
230                .CameraLoc + " with individual: " + SDDB.Individuals.Single(x => x
231                .ID == eventID.SuspectID).Name + "\n";
232        }
233        catch (Exception e)
234        {
235            LogOutput = DateTime.Now.TimeOfDay + ": Database Connection Failed ,
236                Please Check Connection. Details: " + e + "\n";
237        }
238        SafeSetText(LogOutput);
239    }
240}
241
242 private void UpdateEvents(SDDBDataContext SDDB)
243{
244    var OngoingEvents = (from events in SDDB.GetTable<DetectionEvent>() where (
245        events.EventTimeIn != null && events.EventTimeOut == null) select events);
246    bool UpdateFlag = false;
247
248    foreach(DetectionEvent events in OngoingEvents)
249    {
250        string IDName = SDDB.Individuals.Single(v => v.ID == events.SuspectID).
251            Name;
252        if (Globals.EventList.Exists(x => x.EventID == events.EventID))
253            continue;
254        UpdateFlag = true;
255        EventObject newEvent = new EventObject(IDName, (Int32)events.EventCamera,
256            Globals.EventList.Count, (Int32)events.EventID, 0, 0);
257        Globals.EventList.Add(newEvent);
258        Image FailedImg = Image.FromFile("C:/Users/nfox/Source/Repos/SmartCameras/
259            Senior Design UI/Senior Design UI/noImageAvailable.jpg");
260        Globals.EventThumbsList.Images.Add(FailedImg);
261    }
262    if (UpdateFlag)
263        UpdateUISafe();
264}
265
266 private void UpdateUISafe()
267{

```

```

258     if (this.ObjectListView.InvokeRequired)
259     {
260         SetUICallback d = new SetUICallback(UpdateUISafe);
261         this.Invoke(d, new object[] {});
262     }
263     else
264     {
265         for (int i = 0; i < Globals.EventList.Count; i++)
266         {
267             ListViewItem thisColumn = new ListViewItem("", i);
268             thisColumn.SubItems.Add(Globals.EventList[i].Individ);
269             thisColumn.SubItems.Add(Globals.StreamList[(Globals.EventList[i].
270                 CameraID - 1)].get_loc());
271             thisColumn.SubItems.Add(Globals.EventList[i].X.ToString());
272             thisColumn.SubItems.Add(Globals.EventList[i].Y.ToString());
273             thisColumn.SubItems.Add("0");
274             try
275             {
276                 ObjectListView.Items[i] = thisColumn;
277             }
278             catch (ArgumentOutOfRangeException e)
279             {
280                 ObjectListView.Items.Add(thisColumn);
281             }
282         }
283     }
284 }
285
286 private void SafeSetText(String SetText)
287 {
288     if (this.NotificationLog.InvokeRequired)
289     {
290         SetTextCallback d = new SetTextCallback(SafeSetText);
291         this.Invoke(d, new object[] { SetText });
292     }
293     else
294         this.NotificationLog.AppendText(SetText + "\n");
295 }
296
297 private void textBox1_Enter(object sender, EventArgs e)
298 {
299
300     NotificationLog.AppendText( DateTime.Now.TimeOfDay + ":" + textEnterLine.Text
301         + "\n");
302     textEnterLine.Text = "";
303 }
304
305 private void NotificationLog_TextChanged(object sender, EventArgs e)
306 {
307     SDDBDataContext SDDB = new SDDBDataContext();
308     var Array = NotificationLog.Text.Split('\n');
309     if (SDDB.Log == null)
310         return;
311     try
312     {
313         SDDB.Log.NewLine = Array[Array.Length - 1];
314     }
315     catch
316     {
317         SDDB.Log.NewLine = "Bug in TextLog, revisit later";
318     }
319 }
320
321 }
322
323 public class Globals

```

```

324    {
325        internal static List<RSTPStream> StreamList = new List<RSTPStream>();
326        public static ImageList StreamThumbsList = new ImageList();
327        internal static List<EventObject> EventList = new List<EventObject>();
328        public static ImageList EventThumbsList = new ImageList();
329    }
330}
331}

```

Listing 16.28: Code for the Primary Form of the User Interface

16.9 User Interface - Primary Form Designer File

```

1  namespace Senior_Design_UI
2  {
3      partial class Form1
4      {
5          /// <summary>
6          /// Required designer variable.
7          /// </summary>
8          private System.ComponentModel.IContainer components = null;
9
10         /// <summary>
11         /// Clean up any resources being used.
12         /// </summary>
13         /// <param name="disposing">true if managed resources should be disposed;
14         /// otherwise, false.</param>
15         protected override void Dispose(bool disposing)
16         {
17             if (disposing && (components != null))
18             {
19                 components.Dispose();
20             }
21             base.Dispose(disposing);
22         }
23
24         #region Windows Form Designer generated code
25
26         /// <summary>
27         /// Required method for Designer support – do not modify
28         /// the contents of this method with the code editor.
29         /// </summary>
30         private void InitializeComponent()
31         {
32             this.components = new System.ComponentModel.Container();
33             System.ComponentModel.ComponentResourceManager resources = new System.
34                 ComponentModel.ComponentResourceManager(typeof(Form1));
35             this.NoteLog = new System.Windows.Forms.GroupBox();
36             this.NotificationLog = new System.Windows.Forms.RichTextBox();
37             this.textEnterLine = new System.Windows.Forms.TextBox();
38             this.label1 = new System.Windows.Forms.Label();
39             this.VideoStream = new AxAXVLC.AxVLCPlugin2();
40             this.menuStrip1 = new System.Windows.Forms.MenuStrip();
41             this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
42             this.streamsToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
43             this.addStreamToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
44             this.refreshIconsToolStripMenuItem = new System.Windows.Forms.
45                 ToolStripMenuItem();
46             this.imageList1 = new System.Windows.Forms.ImageList(this.components);
47             this.listView1 = new System.Windows.Forms.ListView();
48             this.columnHeader6 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
49                 Forms.ColumnHeader()));
50             this.columnHeader7 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
51                 Forms.ColumnHeader()));
52
53             this.menuStrip1.SuspendLayout();
54             this.SuspendLayout();
55
56             // NoteLog
57             this.NoteLog.Location = new System.Drawing.Point(12, 12);
58             this.NoteLog.Name = "NoteLog";
59             this.NoteLog.Size = new System.Drawing.Size(250, 150);
60             this.NoteLog.TabIndex = 0;
61             this.NoteLog.Text = "Note Log";
62
63             // NotificationLog
64             this.NotificationLog.Location = new System.Drawing.Point(12, 168);
65             this.NotificationLog.Name = "NotificationLog";
66             this.NotificationLog.Size = new System.Drawing.Size(250, 150);
67             this.NotificationLog.TabIndex = 1;
68             this.NotificationLog.Text = "Notification Log";
69
70             // textEnterLine
71             this.textEnterLine.Location = new System.Drawing.Point(12, 320);
72             this.textEnterLine.Name = "textEnterLine";
73             this.textEnterLine.Size = new System.Drawing.Size(250, 150);
74             this.textEnterLine.TabIndex = 2;
75             this.textEnterLine.Text = "Enter text here";
76
77             // label1
78             this.label1.Location = new System.Drawing.Point(12, 480);
79             this.label1.Name = "label1";
80             this.label1.Size = new System.Drawing.Size(250, 150);
81             this.label1.TabIndex = 3;
82             this.label1.Text = "Label 1";
83
84             // VideoStream
85             this.VideoStream.Location = new System.Drawing.Point(270, 12);
86             this.VideoStream.Name = "VideoStream";
87             this.VideoStream.Size = new System.Drawing.Size(300, 200);
88             this.VideoStream.TabIndex = 4;
89             this.VideoStream.Text = "Video Stream";
90
91             // menuStrip1
92             this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
93                 this.fileToolStripMenuItem,
94                 this.streamsToolStripMenuItem,
95                 this.addStreamToolStripMenuItem,
96                 this.refreshIconsToolStripMenuItem });
97             this.menuStrip1.Location = new System.Drawing.Point(0, 0);
98             this.menuStrip1.Name = "menuStrip1";
99             this.menuStrip1.Size = new System.Drawing.Size(300, 22);
100            this.menuStrip1.TabIndex = 5;
101            this.menuStrip1.Text = "menuStrip1";
102
103            // fileToolStripMenuItem
104            this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
105            this.fileToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
106            this.fileToolStripMenuItem.Text = "File";
107
108            // streamsToolStripMenuItem
109            this.streamsToolStripMenuItem.Name = "streamsToolStripMenuItem";
110            this.streamsToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
111            this.streamsToolStripMenuItem.Text = "Streams";
112
113            // addStreamToolStripMenuItem
114            this.addStreamToolStripMenuItem.Name = "addStreamToolStripMenuItem";
115            this.addStreamToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
116            this.addStreamToolStripMenuItem.Text = "Add Stream";
117
118            // refreshIconsToolStripMenuItem
119            this.refreshIconsToolStripMenuItem.Name = "refreshIconsToolStripMenuItem";
120            this.refreshIconsToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
121            this.refreshIconsToolStripMenuItem.Text = "Refresh Icons";
122
123            // imageList1
124            this.imageList1.ImageStream = ((System.Windows.Forms.ImageListStreamer)(this.
125                components.GetImageList("imageList1")));
126            this.imageList1.TransparentColor = System.Drawing.Color.Transparent;
127
128            // listView1
129            this.listView1.Columns.Add(this.columnHeader6);
130            this.listView1.Columns.Add(this.columnHeader7);
131            this.listView1.Location = new System.Drawing.Point(270, 168);
132            this.listView1.Name = "listView1";
133            this.listView1.Size = new System.Drawing.Size(300, 200);
134            this.listView1.TabIndex = 6;
135            this.listView1.UseCompatibleStateImageBehavior = false;
136
137            // columnHeader6
138            this.columnHeader6.Text = "Column 6";
139            this.columnHeader6.Width = 150;
140
141            // columnHeader7
142            this.columnHeader7.Text = "Column 7";
143            this.columnHeader7.Width = 150;
144
145            // 
146            this.ResumeLayout(false);
147        }
148    }
149}

```

```

47     this.columnHeader8 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
48         Forms.ColumnHeader()));
49     this.StreamListMenu = new System.Windows.Forms.GroupBox();
50     this.ObjectList = new System.Windows.Forms.GroupBox();
51     this.ObjectListView = new System.Windows.Forms.ListView();
52     this.columnHeader1 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
53         Forms.ColumnHeader()));
54     this.columnHeader2 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
55         Forms.ColumnHeader()));
56     this.columnHeader3 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
57         Forms.ColumnHeader()));
58     this.columnHeader4 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
59         Forms.ColumnHeader()));
60     this.columnHeader5 = ((System.Windows.Forms.ColumnHeader)(new System.Windows.
61         Forms.ColumnHeader()));
62     this.RightMenu = new System.Windows.Forms.GroupBox();
63     this.pictureBox1 = new System.Windows.Forms.PictureBox();
64     this.Identity = new System.Windows.Forms.GroupBox();
65     this.NoteLog.SuspendLayout();
66     ((System.ComponentModel.ISupportInitialize)(this.VideoStream)).BeginInit();
67     this.menuStrip1.SuspendLayout();
68     this.StreamListMenu.SuspendLayout();
69     this.ObjectList.SuspendLayout();
70     this.RightMenu.SuspendLayout();
71     ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
72     this.Identity.SuspendLayout();
73     this.NoteLog.SuspendLayout();
74     // NoteLog
75     // 
76     this.NoteLog.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.
77         Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Left)
78         | System.Windows.Forms.AnchorStyles.Right)));
79     this.NoteLog.AutoSize = true;
80     this.NoteLog.Controls.Add(this.NotificationLog);
81     this.NoteLog.Controls.Add(this.textEnterLine);
82     this.NoteLog.Location = new System.Drawing.Point(9, 596);
83     this.NoteLog.Margin = new System.Windows.Forms.Padding(2);
84     this.NoteLog.Name = "NoteLog";
85     this.NoteLog.Padding = new System.Windows.Forms.Padding(2);
86     this.NoteLog.Size = new System.Drawing.Size(721, 193);
87     this.NoteLog.TabIndex = 2;
88     this.NoteLog.TabStop = false;
89     this.NoteLog.Text = "Notification Log";
90     // NotificationLog
91     // 
92     this.NotificationLog.Dock = System.Windows.Forms.DockStyle.Top;
93     this.NotificationLog.Location = new System.Drawing.Point(2, 15);
94     this.NotificationLog.Margin = new System.Windows.Forms.Padding(2);
95     this.NotificationLog.MaxLength = 50000;
96     this.NotificationLog.Name = "NotificationLog";
97     this.NotificationLog.ReadOnly = true;
98     this.NotificationLog.ScrollBars = System.Windows.Forms.RichTextBoxScrollBars.
99     ForcedVertical;
100    this.NotificationLog.Size = new System.Drawing.Size(717, 154);
101    this.NotificationLog.TabIndex = 1;
102    this.NotificationLog.Text = "";
103    this.NotificationLog.TextChanged += new System.EventHandler(this.
104        NotificationLog_TextChanged);
105    // textEnterLine
106    // 
107    this.textEnterLine.Dock = System.Windows.Forms.DockStyle.Bottom;
108    this.textEnterLine.Location = new System.Drawing.Point(2, 171);
109    this.textEnterLine.Margin = new System.Windows.Forms.Padding(2);

```

```

105    this.textEnterLine.Name = "textEnterLine";
106    this.textEnterLine.Size = new System.Drawing.Size(717, 20);
107    this.textEnterLine.TabIndex = 0;
108    this.textEnterLine.Leave += new System.EventHandler(this.textBox1_Enter);
109    //
110    // label1
111    //
112    this.label1.AutoSize = true;
113    this.label1.Location = new System.Drawing.Point(18, 25);
114    this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
115    this.label1.Name = "label1";
116    this.label1.Size = new System.Drawing.Size(56, 13);
117    this.label1.TabIndex = 5;
118    this.label1.Text = "Testlabel1";
119    this.label1.Visible = false;
120    //
121    // VideoStream
122    //
123    this.VideoStream.Anchor = ((System.Windows.Forms.AnchorStyles)((System.
124        Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)
125        | System.Windows.Forms.AnchorStyles.Left)
126        | System.Windows.Forms.AnchorStyles.Right));
127    this.VideoStream.Enabled = true;
128    this.VideoStream.Location = new System.Drawing.Point(11, 27);
129    this.VideoStream.Margin = new System.Windows.Forms.Padding(2);
130    this.VideoStream.Name = "VideoStream";
131    this.VideoStream.OcxState = ((System.Windows.Forms.AxHost.State)(resources.
132        GetObject("VideoStream.OcxState")));
133    this.VideoStream.Size = new System.Drawing.Size(1441, 700);
134    this.VideoStream.TabIndex = 8;
135    //
136    // menuStrip1
137    //
138    this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
139        this.fileToolStripMenuItem,
140        this.streamsToolStripMenuItem});
141    this.menuStrip1.Location = new System.Drawing.Point(0, 0);
142    this.menuStrip1.Name = "menuStrip1";
143    this.menuStrip1.Padding = new System.Windows.Forms.Padding(4, 2, 0, 2);
144    this.menuStrip1.Size = new System.Drawing.Size(1426, 24);
145    this.menuStrip1.TabIndex = 10;
146    this.menuStrip1.Text = "menuStrip1";
147    //
148    // fileToolStripMenuItem
149    //
150    this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
151    this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);
152    this.fileToolStripMenuItem.Text = "File";
153    //
154    // streamsToolStripMenuItem
155    //
156    this.streamsToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
157        this.addStreamToolStripMenuItem,
158        this.refreshIconsToolStripMenuItem});
159    this.streamsToolStripMenuItem.Name = "streamsToolStripMenuItem";
160    this.streamsToolStripMenuItem.Size = new System.Drawing.Size(61, 20);
161    this.streamsToolStripMenuItem.Text = "Streams";
162    //
163    // addStreamToolStripMenuItem
164    //
165    this.addStreamToolStripMenuItem.Name = "addStreamToolStripMenuItem";
166    this.addStreamToolStripMenuItem.Size = new System.Drawing.Size(144, 22);
167    this.addStreamToolStripMenuItem.Text = "Add Stream";
168    this.addStreamToolStripMenuItem.Click += new System.EventHandler(this.
        addStreamToolStripMenuItem_Click);
169    //
170    // refreshIconsToolStripMenuItem

```

```

169 // 
170 this.refreshIconsToolStripMenuItem.Name = "refreshIconsToolStripMenuItem";
171 this.refreshIconsToolStripMenuItem.Size = new System.Drawing.Size(144, 22);
172 this.refreshIconsToolStripMenuItem.Text = "Refresh Icons";
173 this.refreshIconsToolStripMenuItem.Click += new System.EventHandler(this.
    refreshIconsToolStripMenuItem_Click);
174 // 
175 // imageList1
176 // 
177 this.imageList1.ColorDepth = System.Windows.Forms.ColorDepth.Depth8Bit;
178 this.imageList1.ImageSize = new System.Drawing.Size(80, 80);
179 this.imageList1.TransparentColor = System.Drawing.Color.Transparent;
180 // 
181 // StreamListView
182 // 
183 this.StreamListView.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] {
184     this.columnHeader6,
185     this.columnHeader7,
186     this.columnHeader8});
187 this.StreamListView.Dock = System.Windows.Forms.DockStyle.Fill;
188 this.StreamListView.FullRowSelect = true;
189 this.StreamListView.GridLines = true;
190 this.StreamListView.Location = new System.Drawing.Point(2, 15);
191 this.StreamListView.Margin = new System.Windows.Forms.Padding(2);
192 this.StreamListView.MultiSelect = false;
193 this.StreamListView.Name = "StreamListView";
194 this.StreamListView.Size = new System.Drawing.Size(420, 347);
195 this.StreamListView.SmallImageList = this.imageList1;
196 this.StreamListView.TabIndex = 11;
197 this.StreamListView.TileSize = new System.Drawing.Size(400, 65);
198 this.StreamListView.UseCompatibleStateImageBehavior = false;
199 this.StreamListView.View = System.Windows.Forms.View.Details;
200 this.StreamListView.ItemActivate += new System.EventHandler(this.
    StreamListView_SelectedIndexChanged);
201 this.StreamListView.SelectedIndexChanged += new System.EventHandler(this.
    StreamListView_SelectedIndexChanged);
202 this.StreamListView.Click += new System.EventHandler(this.
    StreamListView_SelectedIndexChanged);
203 // 
204 // columnHeader6
205 // 
206 this.columnHeader6.Text = "Thumb Nail";
207 this.columnHeader6.Width = 100;
208 // 
209 // columnHeader7
210 // 
211 this.columnHeader7.Text = "Stream Name";
212 this.columnHeader7.Width = 100;
213 // 
214 // columnHeader8
215 // 
216 this.columnHeader8.Text = "Details";
217 this.columnHeader8.Width = 216;
218 // 
219 // StreamListMenu
220 // 
221 this.StreamListMenu.Controls.Add(this.StreamListView);
222 this.StreamListMenu.Dock = System.Windows.Forms.DockStyle.Top;
223 this.StreamListMenu.Location = new System.Drawing.Point(2, 15);
224 this.StreamListMenu.Margin = new System.Windows.Forms.Padding(2);
225 this.StreamListMenu.Name = "StreamListMenu";
226 this.StreamListMenu.Padding = new System.Windows.Forms.Padding(2);
227 this.StreamListMenu.Size = new System.Drawing.Size(424, 364);
228 this.StreamListMenu.TabIndex = 12;
229 this.StreamListMenu.TabStop = false;
230 this.StreamListMenu.Text = "Stream List";
231 // 
232 // ObjectList

```

```

233 // 
234 this.ObjectList.Controls.Add(this.ObjectListView);
235 this.ObjectList.Dock = System.Windows.Forms.DockStyle.Bottom;
236 this.ObjectList.Location = new System.Drawing.Point(2, 382);
237 this.ObjectList.Margin = new System.Windows.Forms.Padding(2);
238 this.ObjectList.Name = "ObjectList";
239 this.ObjectList.Padding = new System.Windows.Forms.Padding(2);
240 this.ObjectList.Size = new System.Drawing.Size(424, 387);
241 this.ObjectList.TabIndex = 13;
242 this.ObjectList.TabStop = false;
243 this.ObjectList.Text = "Object List";
244 // 
245 // ObjectListView
246 // 
247 this.ObjectListView.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] {
248 this.columnHeader1,
249 this.columnHeader2,
250 this.columnHeader1,
251 this.columnHeader3,
252 this.columnHeader4,
253 this.columnHeader5});
254 this.ObjectListView.Dock = System.Windows.Forms.DockStyle.Fill;
255 this.ObjectListView.FullRowSelect = true;
256 this.ObjectListView.GridLines = true;
257 this.ObjectListView.Location = new System.Drawing.Point(2, 15);
258 this.ObjectListView.Margin = new System.Windows.Forms.Padding(2);
259 this.ObjectListView.Name = "ObjectListView";
260 this.ObjectListView.Size = new System.Drawing.Size(420, 370);
261 this.ObjectListView.SmallImageList = this.imageList1;
262 this.ObjectListView.TabIndex = 11;
263 this.ObjectListView.TileSize = new System.Drawing.Size(400, 100);
264 this.ObjectListView.UseCompatibleStateImageBehavior = false;
265 this.ObjectListView.View = System.Windows.Forms.View.Details;
266 // 
267 // columnHeader1
268 // 
269 this.columnHeader1.Text = "Thumbnail";
270 this.columnHeader1.Width = 100;
271 // 
272 // columnHeader2
273 // 
274 this.columnHeader2.Text = "Name";
275 this.columnHeader2.Width = 100;
276 // 
277 // columnHeader1
278 // 
279 this.columnHeader1.Text = "Stream";
280 this.columnHeader1.Width = 90;
281 // 
282 // columnHeader3
283 // 
284 this.columnHeader3.Text = "X";
285 this.columnHeader3.Width = 20;
286 // 
287 // columnHeader4
288 // 
289 this.columnHeader4.Text = "Y";
290 this.columnHeader4.Width = 20;
291 // 
292 // columnHeader5
293 // 
294 this.columnHeader5.Text = "Last Seen";
295 this.columnHeader5.Width = 80;
296 // 
297 // RightMenu
298 // 
299 this.RightMenu.Controls.Add(this.StreamListMenu);
300 this.RightMenu.Controls.Add(this.ObjectList);

```

```

301     this.RightMenu.Dock = System.Windows.Forms.DockStyle.Right;
302     this.RightMenu.Location = new System.Drawing.Point(998, 24);
303     this.RightMenu.Margin = new System.Windows.Forms.Padding(2);
304     this.RightMenu.Name = "RightMenu";
305     this.RightMenu.Padding = new System.Windows.Forms.Padding(2);
306     this.RightMenu.Size = new System.Drawing.Size(428, 771);
307     this.RightMenu.TabIndex = 14;
308     this.RightMenu.TabStop = false;
309     this.RightMenu.Text = "Side Menu";
310     //
311     // pictureBox1
312     //
313     this.pictureBox1.BackgroundImage = global::Senior_Design_UI.Properties.
314         Resources.scuseal;
314     this.pictureBox1.ImageLayout = System.Windows.Forms.ImageLayout.Stretch;
315     this.pictureBox1.Dock = System.Windows.Forms.DockStyle.Fill;
316     this.pictureBox1.Location = new System.Drawing.Point(2, 15);
317     this.pictureBox1.Margin = new System.Windows.Forms.Padding(2);
318     this.pictureBox1.Name = "pictureBox1";
319     this.pictureBox1.Size = new System.Drawing.Size(351, 178);
320     this.pictureBox1.TabIndex = 15;
321     this.pictureBox1.TabStop = false;
322     //
323     // Identity
324     //
325     this.Identity.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.
326         Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Left)
327             | System.Windows.Forms.AnchorStyles.Right));
327     this.Identity.Controls.Add(this.label1);
328     this.Identity.Controls.Add(this.pictureBox1);
329     this.Identity.Location = new System.Drawing.Point(734, 596);
330     this.Identity.Margin = new System.Windows.Forms.Padding(2);
331     this.Identity.Name = "Identity";
332     this.Identity.Padding = new System.Windows.Forms.Padding(2);
333     this.Identity.Size = new System.Drawing.Size(355, 195);
334     this.Identity.TabIndex = 16;
335     this.Identity.TabStop = false;
336     this.Identity.Text = "Placeholder";
337     //
338     // Form1
339     //
340     this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
341     this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
342     this.BackColor = System.Drawing.Color.MintCream;
343     this.ClientSize = new System.Drawing.Size(1426, 795);
344     this.Controls.Add(this.Identity);
345     this.Controls.Add(this.RightMenu);
346     this.Controls.Add(this.VideoStream);
347     this.Controls.Add(this.NoteLog);
348     this.Controls.Add(this.menuStrip1);
349     this.MainMenuStrip = this.menuStrip1;
350     this.Margin = new System.Windows.Forms.Padding(2);
351     this.MaximizeBox = false;
352     this.MinimizeBox = false;
353     this.Name = "Form1";
354     this.Text = "Senior Design UI";
355     this.WindowState = System.Windows.Forms.FormWindowState.Maximized;
356     this.FormClosing += new System.Windows.Forms.FormClosingEventHandler(this.
357         Form1_FormClosing);
357     this.Load += new System.EventHandler(this.Form1_Load);
358     this.ResizeBegin += new System.EventHandler(this.Form1_ResizeBegin);
359     this.ResizeEnd += new System.EventHandler(this.Form1_ResizeBegin);
360     this.Resize += new System.EventHandler(this.Form1_ResizeBegin);
361     this.NoteLog.ResumeLayout(false);
362     this.NoteLog.PerformLayout();
363     ((System.ComponentModel.ISupportInitialize)(this.VideoStream)).EndInit();
364     this.menuStrip1.ResumeLayout(false);

```

```

365     this.menuStrip1.PerformLayout();
366     this.StreamListMenu.ResumeLayout(false);
367     this.ObjectList.ResumeLayout(false);
368     this.RightMenu.ResumeLayout(false);
369     ((System.ComponentModel.ISupportInitialize) this.pictureBox1).EndInit();
370     this.Identity.ResumeLayout(false);
371     this.Identity.PerformLayout();
372     this.ResumeLayout(false);
373     this.PerformLayout();
374 }
375
376 #endregion
377
378 private System.Windows.Forms.GroupBox NoteLog;
379 private System.Windows.Forms.RichTextBox NotificationLog;
380 private System.Windows.Forms.TextBox textEnterLine;
381 private System.Windows.Forms.Label label1;
382 private AxAXVLC.AxVLCPPlugin2 VideoStream;
383 private System.Windows.Forms.MenuStrip menuStrip1;
384 private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;
385 private System.Windows.Forms.ToolStripMenuItem streamsToolStripMenuItem;
386 private System.Windows.Forms.ToolStripMenuItem addStreamToolStripMenuItem;
387 private System.Windows.Forms.ImageList imageList1;
388 private System.Windows.Forms.ListView StreamListView;
389 private System.Windows.Forms.GroupBox StreamListMenu;
390 private System.Windows.Forms.GroupBox ObjectList;
391 private System.Windows.Forms.ListView ObjectListView;
392 private System.Windows.Forms.ColumnHeader columnHeader1;
393 private System.Windows.Forms.ColumnHeader columnHeader2;
394 private System.Windows.Forms.ColumnHeader columnHeader1;
395 private System.Windows.Forms.ColumnHeader columnHeader3;
396 private System.Windows.Forms.ColumnHeader columnHeader4;
397 private System.Windows.Forms.ColumnHeader columnHeader5;
398 private System.Windows.Forms.ColumnHeader columnHeader6;
399 private System.Windows.Forms.ColumnHeader columnHeader7;
400 private System.Windows.Forms.ColumnHeader columnHeader8;
401 private System.Windows.Forms.GroupBox RightMenu;
402 private System.Windows.Forms.PictureBox pictureBox1;
403 private System.Windows.Forms.GroupBox Identity;
404 private System.Windows.Forms.ToolStripMenuItem refreshIconsToolStripMenuItem;
405 }
406 }
407 }
```

Listing 16.29: Code for the Designer of Primary Form of the User Interface

16.10 User Interface - Add Stream Interface

```

1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.Net;
11
12 namespace Senior_Design_UI
13 {
14     public partial class AddStream : Form
15     {
16
17         public AddStream()
```

```

18     {
19         InitializeComponent();
20     }
21
22     private void StreamAddCancel(object sender, EventArgs e)
23     {
24         this.Close();
25     }
26
27     private void StreamAddAccept(object sender, EventArgs e)
28     {
29
30         if (ThumbnailText.Text != null || ThumbnailText.Text != "")
31     {
32         Image NewThumb = Image.FromFile(ThumbnailText.Text);
33         Globals.StreamThumbsList.Images.Add(NewThumb);
34     }
35     //else
36     // Globals.StreamThumbsList.Images.Add();
37
38     RSTPStream NewStream = new RSTPStream(urlTextBox.Text, detTextBox.Text,
39                                         locTextBox.Text, Globals.StreamList.Count + 1);
40     Globals.StreamList.Add(NewStream);
41
42
43     /* Thumbnail WEB Request Code ***NOT WORKING*** */
44     Uri ThumbUrl = new Uri(NewStream.get_url() + "/dsm?");
45     WebRequest ThumbNailRequest = WebRequest.Create("http://192.168.11.32/dsm?");
46     WebResponse ThumbNailReponse = ThumbNailRequest.GetResponse();
47
48     if (ThumbNailReponse.ResponseUri.AbsoluteUri != null)
49     {
50         Image NewThumb = Image.FromStream(ThumbNailReponse.GetResponseStream());
51         Globals.StreamThumbsList.Images.Add(NewThumb);
52     } */
53
54     this.Close();
55 }
56
57     private void button1_Click(object sender, EventArgs e)
58     {
59         //ez creat a invis player
60         //if encountered error
61         //then change text
62
63         labelTestResults.Text = "Test Reuslts : Not Implemented";
64
65     }
66
67     private void button3_Click(object sender, EventArgs e)
68     {
69         OpenFileDialog ImgLoc = new OpenFileDialog();
70         if (ImgLoc.ShowDialog() == DialogResult.OK)
71         {
72             ThumbnailText.Text = ImgLoc.FileName;
73         }
74     }
75 }
76
77 }

```

Listing 16.30: Code for the Add Stream Interface

16.11 User Interface - Add Stream Interface Designer File

```

1  namespace Senior_Design_UI
2  {
3      partial class AddStream
4      {
5          /// <summary>
6          /// Required designer variable.
7          /// </summary>
8          private System.ComponentModel.IContainer components = null;
9
10         /// <summary>
11         /// Clean up any resources being used.
12         /// </summary>
13         /// <param name="disposing">true if managed resources should be disposed;
14         /// otherwise, false.</param>
15         protected override void Dispose(bool disposing)
16         {
17             if (disposing && (components != null))
18             {
19                 components.Dispose();
20             }
21             base.Dispose(disposing);
22         }
23
24         #region Windows Form Designer generated code
25
26         /// <summary>
27         /// Required method for Designer support - do not modify
28         /// the contents of this method with the code editor.
29         /// </summary>
30         private void InitializeComponent()
31         {
32             this.button1 = new System.Windows.Forms.Button();
33             this.label1 = new System.Windows.Forms.Label();
34             this.label2 = new System.Windows.Forms.Label();
35             this.urlTextBox = new System.Windows.Forms.TextBox();
36             this.locTextBox = new System.Windows.Forms.TextBox();
37             this.detTextBox = new System.Windows.Forms.TextBox();
38             this.labelTestResults = new System.Windows.Forms.Label();
39             this.label4 = new System.Windows.Forms.Label();
40             this.button2 = new System.Windows.Forms.Button();
41             this.cancelButton = new System.Windows.Forms.Button();
42             this.label5 = new System.Windows.Forms.Label();
43             this.nameTextBox = new System.Windows.Forms.TextBox();
44             this.button3 = new System.Windows.Forms.Button();
45             this.ThumbnailText = new System.Windows.Forms.TextBox();
46             this.SuspendLayout();
47
48             // button1
49             //
50             this.button1.Location = new System.Drawing.Point(11, 139);
51             this.button1.Margin = new System.Windows.Forms.Padding(2);
52             this.button1.Name = "button1";
53             this.button1.Size = new System.Drawing.Size(74, 19);
54             this.button1.TabIndex = 0;
55             this.button1.Text = "Test Stream";
56             this.button1.UseVisualStyleBackColor = true;
57             this.button1.Visible = false;
58             this.button1.Click += new System.EventHandler(this.button1_Click);
59
60             // label1
61             //
62             this.label1.AutoSize = true;
63             this.label1.Location = new System.Drawing.Point(8, 37);
64             this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
65             this.label1.Name = "label1";
66             this.label1.Size = new System.Drawing.Size(65, 13);
67             this.label1.TabIndex = 1;

```

```

67     this.label1.Text = "Camera IP : ";
68     // 
69     // label2
70     // 
71     this.label2.AutoSize = true;
72     this.label2.Location = new System.Drawing.Point(8, 63);
73     this.label2.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
74     this.label2.Name = "label2";
75     this.label2.Size = new System.Drawing.Size(57, 13);
76     this.label2.TabIndex = 2;
77     this.label2.Text = "Location : ";
78     // 
79     // urlTextBox
80     // 
81     this.urlTextBox.Location = new System.Drawing.Point(77, 34);
82     this.urlTextBox.Margin = new System.Windows.Forms.Padding(2);
83     this.urlTextBox.Name = "urlTextBox";
84     this.urlTextBox.Size = new System.Drawing.Size(162, 20);
85     this.urlTextBox.TabIndex = 3;
86     this.urlTextBox.Text = "192.168.11.32";
87     // 
88     // locTextBox
89     // 
90     this.locTextBox.Location = new System.Drawing.Point(69, 60);
91     this.locTextBox.Margin = new System.Windows.Forms.Padding(2);
92     this.locTextBox.Name = "locTextBox";
93     this.locTextBox.Size = new System.Drawing.Size(170, 20);
94     this.locTextBox.TabIndex = 4;
95     this.locTextBox.Text = "608C";
96     // 
97     // detTextBox
98     // 
99     this.detTextBox.Location = new System.Drawing.Point(60, 85);
100    this.detTextBox.Margin = new System.Windows.Forms.Padding(2);
101    this.detTextBox.Multiline = true;
102    this.detTextBox.Name = "detTextBox";
103    this.detTextBox.Size = new System.Drawing.Size(179, 49);
104    this.detTextBox.TabIndex = 5;
105    this.detTextBox.Text = "Bannon Engineering Outside facing benson quad";
106    // 
107    // labelTestResults
108    // 
109    this.labelTestResults.AutoSize = true;
110    this.labelTestResults.Location = new System.Drawing.Point(91, 142);
111    this.labelTestResults.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
112    this.labelTestResults.Name = "labelTestResults";
113    this.labelTestResults.Size = new System.Drawing.Size(118, 13);
114    this.labelTestResults.TabIndex = 6;
115    this.labelTestResults.Text = "Test Results : Untested";
116    this.labelTestResults.Visible = false;
117    // 
118    // label4
119    // 
120    this.label4.AutoSize = true;
121    this.label4.Location = new System.Drawing.Point(8, 88);
122    this.label4.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
123    this.label4.Name = "label4";
124    this.label4.Size = new System.Drawing.Size(48, 13);
125    this.label4.TabIndex = 7;
126    this.label4.Text = "Details : ";
127    // 
128    // button2
129    // 
130    this.button2.DialogResult = System.Windows.Forms.DialogResult.OK;
131    this.button2.Location = new System.Drawing.Point(34, 211);
132    this.button2.Margin = new System.Windows.Forms.Padding(2);
133    this.button2.Name = "button2";
134    this.button2.Size = new System.Drawing.Size(74, 19);

```

```

135     this.button2.TabIndex = 8;
136     this.button2.Text = "Add Stream";
137     this.button2.UseVisualStyleBackColor = true;
138     this.button2.Click += new System.EventHandler(this.StreamAddAccept);
139     //
140     // cancelButton
141     //
142     this.cancelButton.DialogResult = System.Windows.Forms.DialogResult.Cancel;
143     this.cancelButton.Location = new System.Drawing.Point(136, 211);
144     this.cancelButton.Margin = new System.Windows.Forms.Padding(2);
145     this.cancelButton.Name = "cancelButton";
146     this.cancelButton.Size = new System.Drawing.Size(65, 19);
147     this.cancelButton.TabIndex = 9;
148     this.cancelButton.Text = "Cancel";
149     this.cancelButton.UseVisualStyleBackColor = true;
150     this.cancelButton.Click += new System.EventHandler(this.StreamAddCancel);
151     //
152     // label5
153     //
154     this.label5.AutoSize = true;
155     this.label5.Location = new System.Drawing.Point(8, 12);
156     this.label5.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
157     this.label5.Name = "label5";
158     this.label5.Size = new System.Drawing.Size(80, 13);
159     this.label5.TabIndex = 10;
160     this.label5.Text = "Stream Name : ";
161     //
162     // nameTextBox
163     //
164     this.nameTextBox.Location = new System.Drawing.Point(94, 9);
165     this.nameTextBox.Margin = new System.Windows.Forms.Padding(2);
166     this.nameTextBox.Name = "nameTextBox";
167     this.nameTextBox.Size = new System.Drawing.Size(145, 20);
168     this.nameTextBox.TabIndex = 11;
169     this.nameTextBox.Text = "Home";
170     //
171     // button3
172     //
173     this.button3.Location = new System.Drawing.Point(11, 163);
174     this.button3.Name = "button3";
175     this.button3.Size = new System.Drawing.Size(228, 19);
176     this.button3.TabIndex = 12;
177     this.button3.Text = "Choose Thumbnail";
178     this.button3.UseVisualStyleBackColor = true;
179     this.button3.Click += new System.EventHandler(this.button3_Click);
180     //
181     // ThumbnailText
182     //
183     this.ThumbnailText.Location = new System.Drawing.Point(11, 187);
184     this.ThumbnailText.Name = "ThumbnailText";
185     this.ThumbnailText.ReadOnly = true;
186     this.ThumbnailText.Size = new System.Drawing.Size(228, 20);
187     this.ThumbnailText.TabIndex = 13;
188     //
189     // AddStream
190     //
191     this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
192     this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
193     this.CancelButton = this.cancelButton;
194     this.ClientSize = new System.Drawing.Size(247, 234);
195     this.Controls.Add(this.ThumbnailText);
196     this.Controls.Add(this.button3);
197     this.Controls.Add(this.nameTextBox);
198     this.Controls.Add(this.label5);
199     this.Controls.Add(this.cancelButton);
200     this.Controls.Add(this.button2);
201     this.Controls.Add(this.label14);
202     this.Controls.Add(this.labelTestResults);

```

```

203     this.Controls.Add(this.detTextBox);
204     this.Controls.Add(this.locTextBox);
205     this.Controls.Add(this.urlTextBox);
206     this.Controls.Add(this.label12);
207     this.Controls.Add(this.label11);
208     this.Controls.Add(this.button1);
209     this.Margin = new System.Windows.Forms.Padding(2);
210     this.MaximizeBox = false;
211     this.MinimizeBox = false;
212     this.Name = "AddStream";
213     this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
214     this.Text = "AddStream";
215     this.ResumeLayout(false);
216     this.PerformLayout();
217 }
218 }
219 #endregion
220
221 private System.Windows.Forms.Button button1;
222 private System.Windows.Forms.Label label11;
223 private System.Windows.Forms.Label label12;
224 private System.Windows.Forms.TextBox urlTextBox;
225 private System.Windows.Forms.TextBox locTextBox;
226 private System.Windows.Forms.TextBox detTextBox;
227 private System.Windows.Forms.Label labelTestResults;
228 private System.Windows.Forms.Label label4;
229 private System.Windows.Forms.Button button2;
230 private System.Windows.Forms.Button cancelButton;
231 private System.Windows.Forms.Label label5;
232 private System.Windows.Forms.TextBox nameTextBox;
233 private System.Windows.Forms.Button button3;
234 private System.Windows.Forms.TextBox thumbnailText;
235 }
236 }
237 }
```

Listing 16.31: Code for the Designer of Add Stream Interface

16.12 User Interface - Event Object Class

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Senior_Design_UI
8 {
9     class EventObject
10    {
11        public EventObject()
12        {
13            individual = "";
14            imgindex = 0;
15            camera = 0;
16            Xpos = 0;
17            Ypos = 0;
18        }
19
20        public EventObject( String individ , Int32 cameraid , Int32 img , Int32 eventID ,
21                           Int32 x, Int32 y )
22        {
23            individual = individ;
24            camera = cameraid;
25            imgindex = img;
```

```

25         eventid = eventID;
26         Xpos = x;
27         Ypos = y;
28     }
29
30     public String Individ
31     {
32         get { return individual; }
33         set { individual = value; }
34     }
35
36     public Int32 ImgIndex
37     {
38         get { return imgindex; }
39         set { imgindex = value; }
40     }
41
42     public Int32 CameraID
43     {
44         get { return camera; }
45         set { camera = value; }
46     }
47
48     public Int32 EventID
49     {
50         get { return eventid; }
51         set { eventid = value; }
52     }
53
54     public Int32 X
55     {
56         get { return Xpos; }
57         set { Xpos = value; }
58     }
59
60     public Int32 Y
61     {
62         get { return Ypos; }
63         set { Ypos = value; }
64     }
65
66     // Variables
67     String individual;
68     Int32 imgindex, eventid, camera, Xpos, Ypos;
69 }
70 }
```

Listing 16.32: Code for the Event Object Class

16.13 User Interface - RSTPStream Class

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Senior_Design_UI
8  {
9      class RSTPStream
10     {
11         public RSTPStream()
12         {
13             url = "";
14             details = "";
```

```

15     loc = "";
16     imgIndex = 0;
17     //options = ""; //VLCplugin issues with working
18 }
19
20 public RSTPStream( String urls , String dis , String locs , Int32 name )
21 {
22     url = urls;
23     details = dis;
24     loc = locs;
25     imgIndex = name;
26     //options = ":showdisplay=false"; //Deprecated
27 }
28
29 public string get_url()
30 {
31     return url;
32 }
33
34 public string get_details()
35 {
36     return details;
37 }
38
39 public string get_loc()
40 {
41     return loc;
42 }
43
44 public Int32 get_imgIndex()
45 {
46     return imgIndex;
47 }
48
49 public void set_url(String urls)
50 {
51     url = urls;
52 }
53
54 public void set_details(String dets)
55 {
56     details = dets;
57 }
58
59 public void set_loc(String locs)
60 {
61     loc = locs;
62 }
63
64 public void set_imgIndex(Int32 img)
65 {
66     imgIndex = img;
67 }
68
69 // Variables
70 String url , name , details , loc ;
71 Int32 imgIndex ;
72 }
73 }
```

Listing 16.33: Code for the RSTPStream Class

16.14 User Interface - Database Configuration File

```
| #pragma warning disable 1591 |
```

```

2 //-----
3 // <auto-generated>
4 //   This code was generated by a tool.
5 //   Runtime Version:4.0.30319.34014
6 //
7 //   Changes to this file may cause incorrect behavior and will be lost if
8 //   the code is regenerated.
9 // </auto-generated>
10 //-----
11
12 namespace Senior_Design_UI
13 {
14     using System.Data.Linq;
15     using System.Data.Linq.Mapping;
16     using System.Data;
17     using System.Collections.Generic;
18     using System.Reflection;
19     using System.Linq;
20     using System.Linq.Expressions;
21     using System.ComponentModel;
22     using System;
23
24
25 [global::System.Data.Linq.Mapping.DatabaseAttribute(Name="SeniorDesignDB2")]
26 public partial class SDDBDATAContext : System.Data.Linq.DataContext
27 {
28
29     private static System.Data.Linq.Mapping.MappingSource mappingSource = new
30         AttributeMappingSource();
31
32     #region Extensibility Method Definitions
33     partial void OnCreated();
34     partial void InsertCamera(Camera instance);
35     partial void UpdateCamera(Camera instance);
36     partial void DeleteCamera(Camera instance);
37     partial void InsertCameraStat(CameraStat instance);
38     partial void UpdateCameraStat(CameraStat instance);
39     partial void DeleteCameraStat(CameraStat instance);
40     partial void InsertDetectionEvent(DetectionEvent instance);
41     partial void UpdateDetectionEvent(DetectionEvent instance);
42     partial void DeleteDetectionEvent(DetectionEvent instance);
43     partial void InsertIndividual(Individual instance);
44     partial void UpdateIndividual(Individual instance);
45     partial void DeleteIndividual(Individual instance);
46     partial void InsertKnownImg(KnownImg instance);
47     partial void UpdateKnownImg(KnownImg instance);
48     partial void DeleteKnownImg(KnownImg instance);
49     partial void InsertAnomaly(Anomaly instance);
50     partial void UpdateAnomaly(Anomaly instance);
51     partial void DeleteAnomaly(Anomaly instance);
52     #endregion
53
54     public SDDBDATAContext() :
55         base(global::Senior_Design_UI.Properties.Settings.Default.
56               SeniorDesignDB2ConnectionString, mappingSource)
57     {
58         OnCreated();
59     }
60
61     public SDDBDATAContext(string connection) :
62         base(connection, mappingSource)
63     {
64         OnCreated();
65     }
66
67     public SDDBDATAContext(System.Data.IDbConnection connection) :
68         base(connection, mappingSource)
69     {

```

```

68     OnCreated();
69 }
70
71 public SDDBDataContext(string connection, System.Data.Linq.Mapping.MappingSource
72     mappingSource) :
73     base(connection, mappingSource)
74 {
75     OnCreated();
76 }
77
78 public SDDBDataContext(System.Data.IDbConnection connection, System.Data.Linq.Mapping.
79     MappingSource mappingSource) :
80     base(connection, mappingSource)
81 {
82     OnCreated();
83 }
84
85 public System.Data.Linq.Table<Camera> Cameras
86 {
87     get
88     {
89         return this.GetTable<Camera>();
90     }
91 }
92
93 public System.Data.Linq.Table<CameraStat> CameraStats
94 {
95     get
96     {
97         return this.GetTable<CameraStat>();
98     }
99 }
100
101 public System.Data.Linq.Table<DetectionEvent> DetectionEvents
102 {
103     get
104     {
105         return this.GetTable<DetectionEvent>();
106     }
107 }
108
109 public System.Data.Linq.Table<Individual> Individuals
110 {
111     get
112     {
113         return this.GetTable<Individual>();
114     }
115 }
116
117 public System.Data.Linq.Table<KnownImg> KnownImgs
118 {
119     get
120     {
121         return this.GetTable<KnownImg>();
122     }
123 }
124
125 public System.Data.Linq.Table<Anomaly> Anomalies
126 {
127     get
128     {
129         return this.GetTable<Anomaly>();
130     }
131 }
132 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Camera")]
133 public partial class Camera : INotifyPropertyChanging, INotifyPropertyChanged

```

```

134    {
135
136    private static PropertyChangingEventArgs emptyChangingEventArgs = new
137        PropertyChangingEventArgs(String.Empty);
138
139    private short _CameraID;
140
141    private string _CameraName;
142
143    private string _CameraIP;
144
145    private string _FPGAIP;
146
147    private string _CameraLoc;
148
149    private string _CameraDetails;
150
151    private System.Nullable<short> _imgIndex;
152
153    private string _ThumbPath;
154
155    private EntitySet<CameraStat> _CameraStats;
156
157    private EntitySet<DetectionEvent> _DetectionEvents;
158
159    #region Extensibility Method Definitions
160    partial void OnLoaded();
161    partial void OnValidate(System.Data.Linq.ChangeAction action);
162    partial void OnCreated();
163    partial void OnCameraIDChanging(short value);
164    partial void OnCameraIDChanged();
165    partial void OnCameraNameChanging(string value);
166    partial void OnCameraNameChanged();
167    partial void OnCameraIPChanging(string value);
168    partial void OnCameraIPChanged();
169    partial void OnFPGAIPChanging(string value);
170    partial void OnFPGAIPChanged();
171    partial void OnCameraLocChanging(string value);
172    partial void OnCameraLocChanged();
173    partial void OnCameraDetailsChanging(string value);
174    partial void OnCameraDetailsChanged();
175    partial void OnimgIndexChanging(System.Nullable<short> value);
176    partial void OnimgIndexChanged();
177    partial void OnThumbPathChanging(string value);
178    partial void OnThumbPathChanged();
179    #endregion
180
181    public Camera()
182    {
183        this._CameraStats = new EntitySet<CameraStat>(new Action<CameraStat>(this.
184            attach_CameraStats), new Action<CameraStat>(this.detach_CameraStats));
185        this._DetectionEvents = new EntitySet<DetectionEvent>(new Action<DetectionEvent>(
186            this.attach_DetectionEvents), new Action<DetectionEvent>(this.
187            detach_DetectionEvents));
188        OnCreated();
189    }
190
191    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _CameraID", DbType="
192        SmallInt NOT NULL", IsPrimaryKey=true)]
193    public short CameraID
194    {
195        get
196        {
197            return this._CameraID;
198        }
199        set
200        {
201            if ((this._CameraID != value))

```

```

197     {
198         this.OnCameraIDChanging(value);
199         this.SendPropertyChanging();
200         this._CameraID = value;
201         this.SendPropertyChanged("CameraID");
202         this.OnCameraIDChanged();
203     }
204 }
205 }
206
207 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraName", DbType="
208     VarChar(20) NOT NULL", CanBeNull=false)]
209 public string CameraName
210 {
211     get
212     {
213         return this._CameraName;
214     }
215     set
216     {
217         if ((this._CameraName != value))
218         {
219             this.OnCameraNameChanging(value);
220             this.SendPropertyChanging();
221             this._CameraName = value;
222             this.SendPropertyChanged("CameraName");
223             this.OnCameraNameChanged();
224         }
225     }
226 }
227
228 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraIP", DbType="VarChar
229     (40)")]
230 public string CameraIP
231 {
232     get
233     {
234         return this._CameraIP;
235     }
236     set
237     {
238         if ((this._CameraIP != value))
239         {
240             this.OnCameraIPChanging(value);
241             this.SendPropertyChanging();
242             this._CameraIP = value;
243             this.SendPropertyChanged("CameraIP");
244             this.OnCameraIPChanged();
245         }
246     }
247 }
248
249 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FPGAIP", DbType="VarChar
250     (40)")]
251 public string FPGAIP
252 {
253     get
254     {
255         return this._FPGAIP;
256     }
257     set
258     {
259         if ((this._FPGAIP != value))
260         {
261             this.OnFPGAIPChanging(value);
262             this.SendPropertyChanging();
263             this._FPGAIP = value;
264             this.SendPropertyChanged("FPGAIP");
265         }
266     }

```

```

262         this.OnFPGAIPChanged();
263     }
264 }
265 }
266
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _CameraLoc", DbType="
    VarChar(40)")]
268 public string CameraLoc
269 {
270     get
271     {
272         return this._CameraLoc;
273     }
274     set
275     {
276         if ((this._CameraLoc != value))
277         {
278             this.OnCameraLocChanging(value);
279             this.SendPropertyChanging();
280             this._CameraLoc = value;
281             this.SendPropertyChanged("CameraLoc");
282             this.OnCameraLocChanged();
283         }
284     }
285 }
286
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _CameraDetails", DbType="
    VarChar(100)")]
288 public string CameraDetails
289 {
290     get
291     {
292         return this._CameraDetails;
293     }
294     set
295     {
296         if ((this._CameraDetails != value))
297         {
298             this.OnCameraDetailsChanging(value);
299             this.SendPropertyChanging();
300             this._CameraDetails = value;
301             this.SendPropertyChanged("CameraDetails");
302             this.OnCameraDetailsChanged();
303         }
304     }
305 }
306
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _imgIndex", DbType="
    SmallInt")]
308 public System.Nullable<short> imgIndex
309 {
310     get
311     {
312         return this._imgIndex;
313     }
314     set
315     {
316         if ((this._imgIndex != value))
317         {
318             this.OnimgIndexChanging(value);
319             this.SendPropertyChanging();
320             this._imgIndex = value;
321             this.SendPropertyChanged("imgIndex");
322             this.OnimgIndexChanged();
323         }
324     }
325 }
326

```

```

327 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="ThumbPath", DbType="
328     VarChar(10)")]
329 public string ThumbPath
330 {
331     get
332     {
333         return this._ThumbPath;
334     }
335     set
336     {
337         if ((this._ThumbPath != value))
338         {
339             this.OnThumbPathChanging(value);
340             this.SendPropertyChanging();
341             this._ThumbPath = value;
342             this.SendPropertyChanged("ThumbPath");
343             this.OnThumbPathChanged();
344         }
345     }
346 }
347 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat",
348     Storage="_CameraStats", ThisKey="CameraID", OtherKey="CameraID")]
349 public EntitySet<CameraStat> CameraStats
350 {
351     get
352     {
353         return this._CameraStats;
354     }
355     set
356     {
357         this._CameraStats.Assign(value);
358     }
359 }
360 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_DetectionEvent",
361     Storage="_DetectionEvents", ThisKey="CameraID", OtherKey="EventCamera")]
362 public EntitySet<DetectionEvent> DetectionEvents
363 {
364     get
365     {
366         return this._DetectionEvents;
367     }
368     set
369     {
370         this._DetectionEvents.Assign(value);
371     }
372 }
373 public event PropertyChangedEventHandler PropertyChanged;
374
375 protected virtual void SendPropertyChanging()
376 {
377     if ((this.PropertyChanging != null))
378     {
379         this.PropertyChanging(this, emptyChangingEventArgs);
380     }
381 }
382
383 protected virtual void SendPropertyChanged(String propertyName)
384 {
385     if ((this.PropertyChanged != null))
386     {
387         this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
388     }
389 }
390 }
391 
```

```

392
393     private void attach_CameraStats(CameraStat entity)
394     {
395         this .SendPropertyChanging();
396         entity .Camera = this ;
397     }
398
399     private void detach_CameraStats(CameraStat entity)
400     {
401         this .SendPropertyChanging();
402         entity .Camera = null;
403     }
404
405     private void attach_DetectionEvents(DetectionEvent entity)
406     {
407         this .SendPropertyChanging();
408         entity .Camera = this ;
409     }
410
411     private void detach_DetectionEvents(DetectionEvent entity)
412     {
413         this .SendPropertyChanging();
414         entity .Camera = null;
415     }
416 }
417
418 [global :: System .Data .Linq .Mapping .TableAttribute (Name="dbo . CameraStats")]
419 public partial class CameraStat : INotifyPropertyChanging , INotifyPropertyChanged
420 {
421
422     private static PropertyChangingEventArgs emptyChangingEventArgs = new
423         PropertyChangingEventArgs (String .Empty);
424
425     private short _TimeID;
426
427     private short _CameraID;
428
429     private System .Nullable <System .TimeSpan> _LoTimeRange;
430
431     private System .Nullable <System .TimeSpan> _HiTimeRange;
432
433     private System .Nullable <double> _Simga;
434
435     private System .Nullable <double> _Mean;
436
437     private System .Nullable <int> _NValue;
438
439     private EntityRef <Camera> _Camera;
440
441 #region Extensibility Method Definitions
442     partial void OnLoaded();
443     partial void OnValidate (System .Data .Linq .ChangeAction action);
444     partial void OnCreated();
445     partial void OnTimeIDChanging (short value);
446     partial void OnTimeIDChanged();
447     partial void OnCameraIDChanging (short value);
448     partial void OnCameraIDChanged();
449     partial void OnLoTimeRangeChanging (System .Nullable <System .TimeSpan> value);
450     partial void OnLoTimeRangeChanged();
451     partial void OnHiTimeRangeChanging (System .Nullable <System .TimeSpan> value);
452     partial void OnHiTimeRangeChanged();
453     partial void OnSimgaChanging (System .Nullable <double> value);
454     partial void OnSimgaChanged();
455     partial void OnMeanChanging (System .Nullable <double> value);
456     partial void OnMeanChanged();
457     partial void OnNValueChanging (System .Nullable <int> value);
458     partial void OnNValueChanged();
459 #endregion

```

```

459
460     public CameraStat()
461     {
462         this._Camera = default(EntityRef<Camera>);
463         OnCreated();
464     }
465
466     [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="TimeID", DbType="SmallInt
467         NOT NULL", IsPrimaryKey=true)]
468     public short TimeID
469     {
470         get
471         {
472             return this._TimeID;
473         }
474         set
475         {
476             if ((this._TimeID != value))
477             {
478                 this.OnTimeIDChanging(value);
479                 this.SendPropertyChanging();
480                 this._TimeID = value;
481                 this.SendPropertyChanged("TimeID");
482                 this.OnTimeIDChanged();
483             }
484         }
485     }
486
487     [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="CameraID", DbType="
488         SmallInt NOT NULL")]
489     public short CameraID
490     {
491         get
492         {
493             return this._CameraID;
494         }
495         set
496         {
497             if ((this._CameraID != value))
498             {
499                 if (this._Camera.HasLoadedOrAssignedValue)
500                 {
501                     throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
502                 }
503                 this.OnCameraIDChanging(value);
504                 this.SendPropertyChanging();
505                 this._CameraID = value;
506                 this.SendPropertyChanged("CameraID");
507                 this.OnCameraIDChanged();
508             }
509         }
510     }
511
512     [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="LoTimeRange", DbType="Time
513         "")]
514     public System.Nullable<System.TimeSpan> LoTimeRange
515     {
516         get
517         {
518             return this._LoTimeRange;
519         }
520         set
521         {
522             if ((this._LoTimeRange != value))
523             {
524                 this.OnLoTimeRangeChanging(value);
525                 this.SendPropertyChanging();
526                 this._LoTimeRange = value;
527             }
528         }
529     }

```

```

524         this .SendPropertyChanged("LoTimeRange");
525         this .OnLoTimeRangeChanged();
526     }
527 }
528 }
529
530 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _HiTimeRange" , DbType="Time
531 ")]
532 public System.Nullable<System.TimeSpan> HiTimeRange
533 {
534     get
535     {
536         return this ._HiTimeRange;
537     }
538     set
539     {
540         if ((this ._HiTimeRange != value))
541         {
542             this .OnHiTimeRangeChanging(value);
543             this .SendPropertyChanging();
544             this ._HiTimeRange = value;
545             this .SendPropertyChanged("HiTimeRange");
546             this .OnHiTimeRangeChanged();
547         }
548     }
549
550 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _Simga" , DbType="Float")]
551 public System.Nullable<double> Simga
552 {
553     get
554     {
555         return this ._Simga;
556     }
557     set
558     {
559         if ((this ._Simga != value))
560         {
561             this .OnSimgaChanging(value);
562             this .SendPropertyChanging();
563             this ._Simga = value;
564             this .SendPropertyChanged("Simga");
565             this .OnSimgaChanged();
566         }
567     }
568 }
569
570 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _Mean" , DbType="Float")]
571 public System.Nullable<double> Mean
572 {
573     get
574     {
575         return this ._Mean;
576     }
577     set
578     {
579         if ((this ._Mean != value))
580         {
581             this .OnMeanChanging(value);
582             this .SendPropertyChanging();
583             this ._Mean = value;
584             this .SendPropertyChanged("Mean");
585             this .OnMeanChanged();
586         }
587     }
588 }
589
590 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _NValue" , DbType="Int")]

```

```

591     public System.Nullable<int> NValue
592     {
593         get
594         {
595             return this._NValue;
596         }
597         set
598         {
599             if ((this._NValue != value))
600             {
601                 this.OnNValueChanging(value);
602                 this.SendPropertyChanging();
603                 this._NValue = value;
604                 this.SendPropertyChanged("NValue");
605                 this.OnNValueChanged();
606             }
607         }
608     }
609
610     [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat",
611         Storage="_Camera", ThisKey="CameraID", OtherKey="CameraID", IsForeignKey=true,
612         DeleteOnNull=true, DeleteRule="CASCADE")]
613     public Camera Camera
614     {
615         get
616         {
617             return this._Camera.Entity;
618         }
619         set
620         {
621             Camera previousValue = this._Camera.Entity;
622             if (((previousValue != value)
623                  || (this._Camera.HasLoadedOrAssignedValue == false)))
624             {
625                 this.SendPropertyChanging();
626                 if ((previousValue != null))
627                 {
628                     this._Camera.Entity = null;
629                     previousValue.CameraStats.Remove(this);
630                 }
631                 this._Camera.Entity = value;
632                 if ((value != null))
633                 {
634                     value.CameraStats.Add(this);
635                     this._CameraID = value.CameraID;
636                 }
637                 else
638                 {
639                     this._CameraID = default(short);
640                 }
641                 this.SendPropertyChanged("Camera");
642             }
643         }
644     }
645
646     public event PropertyChangingEventHandler PropertyChanging;
647
648     protected virtual void SendPropertyChanging()
649     {
650         if ((this.PropertyChanging != null))
651         {
652             this.PropertyChanging(this, emptyChangingEventArgs);
653         }
654     }
655
656     protected virtual void SendPropertyChanged(String propertyName)

```

```

657     {
658         if ((this.PropertyChanged != null))
659         {
660             this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
661         }
662     }
663 }
664
665 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.DetectionEvent")]
666 public partial class DetectionEvent : INotifyPropertyChanging, INotifyPropertyChanged
667 {
668
669     private static PropertyChangingEventArgs emptyChangingEventArgs = new
670         PropertyChangingEventArgs(String.Empty);
671
672     private int _EventID;
673
674     private short _EventCamera;
675
676     private System.TimeSpan _EventTimeIn;
677
678     private System.Nullable<System.TimeSpan> _EventTimeOut;
679
680     private System.Nullable<int> _isChecked;
681
682     private System.Nullable<short> _EntryType;
683
684     private System.Nullable<short> _ExitType;
685
686     private int _SuspectID;
687
688     private double _Confidence;
689
690     private EntitySet<Anomaly> _Anomalies;
691
692     private EntityRef<Camera> _Camera;
693
694     private EntityRef<Individual> _Individual;
695
696     #region Extensibility Method Definitions
697     partial void OnLoaded();
698     partial void OnValidate(System.Data.Linq.ChangeAction action);
699     partial void OnCreated();
700     partial void OnEventIDChanging(int value);
701     partial void OnEventIDChanged();
702     partial void OnEventCameraChanging(short value);
703     partial void OnEventCameraChanged();
704     partial void OnEventTimeInChanging(System.TimeSpan value);
705     partial void OnEventTimeInChanged();
706     partial void OnEventTimeOutChanging(System.Nullable<System.TimeSpan> value);
707     partial void OnEventTimeOutChanged();
708     partial void OnisCheckedChanging(System.Nullable<int> value);
709     partial void OnisCheckedChanged();
710     partial void OnEntryTypeChanging(System.Nullable<short> value);
711     partial void OnEntryTypeChanged();
712     partial void OnExitTypeChanging(System.Nullable<short> value);
713     partial void OnExitTypeChanged();
714     partial void OnSuspectIDChanging(int value);
715     partial void OnSuspectIDChanged();
716     partial void OnConfidenceChanging(double value);
717     partial void OnConfidenceChanged();
718     #endregion
719
720     public DetectionEvent()
721     {
722         this._Anomalies = new EntitySet<Anomaly>(new Action<Anomaly>(this.attach_Anomalies),
723             new Action<Anomaly>(this.detach_Anomalies));
724         this._Camera = default(EntityRef<Camera>);

```

```

723     this._Individual = default(EntityRef<Individual>);
724     OnCreated();
725 }
726
727 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EventID", DbType="Int NOT
728     NULL", IsPrimaryKey=true)]
729 public int EventID
730 {
731     get
732     {
733         return this._EventID;
734     }
735     set
736     {
737         if ((this._EventID != value))
738         {
739             this.OnEventIDChanging(value);
740             this.SendPropertyChanging();
741             this._EventID = value;
742             this.SendPropertyChanged("EventID");
743             this.OnEventIDChanged();
744         }
745     }
746 }
747
748 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EventCamera", DbType="
749     SmallInt NOT NULL")]
750 public short EventCamera
751 {
752     get
753     {
754         return this._EventCamera;
755     }
756     set
757     {
758         if ((this._EventCamera != value))
759         {
760             if (this._Camera.HasLoadedOrAssignedValue)
761             {
762                 throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
763             }
764             this.OnEventCameraChanging(value);
765             this.SendPropertyChanging();
766             this._EventCamera = value;
767             this.SendPropertyChanged("EventCamera");
768             this.OnEventCameraChanged();
769         }
770     }
771 }
772
773 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EventTimeIn", DbType="Time
774     NOT NULL")]
775 public System.TimeSpan EventTimeIn
776 {
777     get
778     {
779         return this._EventTimeIn;
780     }
781     set
782     {
783         if ((this._EventTimeIn != value))
784         {
785             this.OnEventTimeInChanging(value);
786             this.SendPropertyChanging();
787             this._EventTimeIn = value;
788             this.SendPropertyChanged("EventTimeIn");
789             this.OnEventTimeInChanged();
790         }
791     }
792 }

```

```

788     }
789 }
790
791 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=""_EventTimeOut", DbType=""Time")]
792 public System.Nullable<System.TimeSpan> EventTimeOut
793 {
794     get
795     {
796         return this._EventTimeOut;
797     }
798     set
799     {
800         if ((this._EventTimeOut != value))
801         {
802             this.OnEventTimeOutChanging(value);
803             this.SendPropertyChanging();
804             this._EventTimeOut = value;
805             this.SendPropertyChanged("EventTimeOut");
806             this.OnEventTimeOutChanged();
807         }
808     }
809 }
810
811 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=""_isChecked", DbType=""Int")]
812 public System.Nullable<int> isChecked
813 {
814     get
815     {
816         return this._isChecked;
817     }
818     set
819     {
820         if ((this._isChecked != value))
821         {
822             this.OnisCheckedChanging(value);
823             this.SendPropertyChanging();
824             this._isChecked = value;
825             this.SendPropertyChanged("isChecked");
826             this.OnisCheckedChanged();
827         }
828     }
829 }
830
831 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=""_EntryType", DbType=""SmallInt")]
832 public System.Nullable<short> EntryType
833 {
834     get
835     {
836         return this._EntryType;
837     }
838     set
839     {
840         if ((this._EntryType != value))
841         {
842             this.OnEntryTypeChanging(value);
843             this.SendPropertyChanging();
844             this._EntryType = value;
845             this.SendPropertyChanged("EntryType");
846             this.OnEntryTypeChanged();
847         }
848     }
849 }
850
851 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=""_ExitType", DbType=""SmallInt")]
852 public System.Nullable<short> ExitType

```

```

853    {
854        get
855        {
856            return this._ExitType;
857        }
858        set
859        {
860            if ((this._ExitType != value))
861            {
862                this.OnExitTypeChanging(value);
863                this.SendPropertyChanging();
864                this._ExitType = value;
865                this.SendPropertyChanged("ExitType");
866                this.OnExitTypeChanged();
867            }
868        }
869    }
870
871    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_SuspectID", DbType="Int
872    NOT NULL")]
872    public int SuspectID
873    {
874        get
875        {
876            return this._SuspectID;
877        }
878        set
879        {
880            if ((this._SuspectID != value))
881            {
882                if (this._Individual.HasLoadedOrAssignedValue)
883                {
884                    throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
885                }
886                this.OnSuspectIDChanging(value);
887                this.SendPropertyChanging();
888                this._SuspectID = value;
889                this.SendPropertyChanged("SuspectID");
890                this.OnSuspectIDChanged();
891            }
892        }
893    }
894
895    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Confidence", DbType="Float
896    NOT NULL")]
896    public double Confidence
897    {
898        get
899        {
900            return this._Confidence;
901        }
902        set
903        {
904            if ((this._Confidence != value))
905            {
906                this.OnConfidenceChanging(value);
907                this.SendPropertyChanging();
908                this._Confidence = value;
909                this.SendPropertyChanged("Confidence");
910                this.OnConfidenceChanged();
911            }
912        }
913    }
914
915    [global::System.Data.Linq.Mapping.AssociationAttribute(Name="DetectionEvent_Anomaly",
916    Storage="_Anomalies", ThisKey="EventID", OtherKey="EventID")]
916    public EntitySet<Anomaly> Anomalies
917    {

```

```

918     get
919     {
920         return this._Anomalies;
921     }
922     set
923     {
924         this._Anomalies.Assign(value);
925     }
926 }
927
928 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_DetectionEvent",
929     Storage="_Camera", ThisKey="EventCamera", OtherKey="CameraID", IsForeignKey=true,
930     DeleteOnNull=true, DeleteRule="CASCADE")]
931 public Camera Camera
932 {
933     get
934     {
935         return this._Camera.Entity;
936     }
937     set
938     {
939         Camera previousValue = this._Camera.Entity;
940         if (((previousValue != value)
941             || (this._Camera.HasLoadedOrAssignedValue == false)))
942         {
943             this.SendPropertyChanging();
944             if ((previousValue != null))
945             {
946                 this._Camera.Entity = null;
947                 previousValue.DetectionEvents.Remove(this);
948             }
949             this._Camera.Entity = value;
950             if ((value != null))
951             {
952                 value.DetectionEvents.Add(this);
953                 this._EventCamera = value.CameraID;
954             }
955             else
956             {
957                 this._EventCamera = default(short);
958             }
959             this.SendPropertyChanged("Camera");
960         }
961     }
962
963 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Individual_DetectionEvent",
964     Storage="_Individual", ThisKey="SuspectID", OtherKey="ID", IsForeignKey=true,
965     DeleteOnNull=true, DeleteRule="CASCADE")]
966 public Individual Individual
967 {
968     get
969     {
970         return this._Individual.Entity;
971     }
972     set
973     {
974         Individual previousValue = this._Individual.Entity;
975         if (((previousValue != value)
976             || (this._Individual.HasLoadedOrAssignedValue == false)))
977         {
978             this.SendPropertyChanging();
979             if ((previousValue != null))
980             {
981                 this._Individual.Entity = null;
982                 previousValue.DetectionEvents.Remove(this);
983             }
984             this._Individual.Entity = value;

```

```

982         if ((value != null))
983     {
984         value.DetectionEvents.Add(this);
985         this._SuspectID = value.ID;
986     }
987     else
988     {
989         this._SuspectID = default(int);
990     }
991     this.SendPropertyChanged("Individual");
992 }
993 }
994 }
995
996 public event PropertyChangingEventHandler PropertyChanging;
997
998 public event PropertyChangedEventHandler PropertyChanged;
999
1000 protected virtual void SendPropertyChanging()
1001 {
1002     if ((this.PropertyChanging != null))
1003     {
1004         this.PropertyChanging(this, emptyChangingEventArgs);
1005     }
1006 }
1007
1008 protected virtual void SendPropertyChanged(String propertyName)
1009 {
1010     if ((this.PropertyChanged != null))
1011     {
1012         this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
1013     }
1014 }
1015
1016 private void attach_Anomalies(Anomaly entity)
1017 {
1018     this.SendPropertyChanging();
1019     entity.DetectionEvent = this;
1020 }
1021
1022 private void detach_Anomalies(Anomaly entity)
1023 {
1024     this.SendPropertyChanging();
1025     entity.DetectionEvent = null;
1026 }
1027 }
1028
1029 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Individual")]
1030 public partial class Individual : INotifyPropertyChanging, INotifyPropertyChanged
1031 {
1032
1033     private static PropertyChangingEventArgs emptyChangingEventArgs = new
1034         PropertyChangingEventArgs(String.Empty);
1035
1036     private int _ID;
1037
1038     private string _Name;
1039
1040     private string _Details;
1041
1042     private EntitySet<DetectionEvent> _DetectionEvents;
1043
1044     private EntitySet<KnownImg> _KnownImgs;
1045
1046     #region Extensibility Method Definitions
1047     partial void OnLoaded();
1048     partial void OnValidate(System.Data.Linq.ChangeAction action);
1049     partial void OnCreated();

```

```

1049 partial void OnIDChanging(int value);
1050 partial void OnIDChanged();
1051 partial void OnNameChanging(string value);
1052 partial void OnNameChanged();
1053 partial void OnDetailsChanging(string value);
1054 partial void OnDetailsChanged();
1055 #endregion
1056
1057 public Individual()
1058 {
1059     this._DetectionEvents = new EntitySet<DetectionEvent>(new Action<DetectionEvent>(
1060         this.attach_DetectionEvents), new Action<DetectionEvent>(this.
1061         detach_DetectionEvents));
1062     this._KnownImgs = new EntitySet<KnownImg>(new Action<KnownImg>(this.attach_KnownImgs
1063         ), new Action<KnownImg>(this.detach_KnownImgs));
1064     OnCreated();
1065 }
1066
1067 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID", AutoSync=AutoSync.
1068     OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true, IsDbGenerated=true)]
1069 public int ID
1070 {
1071     get
1072     {
1073         return this._ID;
1074     }
1075     set
1076     {
1077         if ((this._ID != value))
1078         {
1079             this.OnIDChanging(value);
1080             this.SendPropertyChanging();
1081             this._ID = value;
1082             this.SendPropertyChanged("ID");
1083             this.OnIDChanged();
1084         }
1085     }
1086 }
1087
1088 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Name", DbType="VarChar(40)
1089     NOT NULL", CanBeNull=false)]
1090 public string Name
1091 {
1092     get
1093     {
1094         return this._Name;
1095     }
1096     set
1097     {
1098         if ((this._Name != value))
1099         {
1100             this.OnNameChanging(value);
1101             this.SendPropertyChanging();
1102             this._Name = value;
1103             this.SendPropertyChanged("Name");
1104             this.OnNameChanged();
1105         }
1106     }
1107 }
1108
1109 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Details", DbType="VarChar
1110     (100)")]
1111 public string Details
1112 {
1113     get
1114     {
1115         return this._Details;
1116     }

```

```

1111     set
1112     {
1113         if ((this._Details != value))
1114         {
1115             this.OnDetailsChanging(value);
1116             this.SendPropertyChanging();
1117             this._Details = value;
1118             this.SendPropertyChanged("Details");
1119             this.OnDetailsChanged();
1120         }
1121     }
1122 }
1123
1124 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Individual_DetectionEvent",
1125   Storage="_DetectionEvents", ThisKey="ID", OtherKey="SuspectID")]
1126 public EntitySet<DetectionEvent> DetectionEvents
1127 {
1128     get
1129     {
1130         return this._DetectionEvents;
1131     }
1132     set
1133     {
1134         this._DetectionEvents.Assign(value);
1135     }
1136 }
1137
1138 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Individual_KnownImg",
1139   Storage="_KnownImg", ThisKey="ID", OtherKey="IDImg")]
1140 public EntitySet<KnownImg> KnownImg
1141 {
1142     get
1143     {
1144         return this._KnownImg;
1145     }
1146     set
1147     {
1148         this._KnownImg.Assign(value);
1149     }
1150 }
1151
1152 public event PropertyChangingEventHandler PropertyChanging;
1153
1154 protected virtual void SendPropertyChanging()
1155 {
1156     if ((this.PropertyChanging != null))
1157     {
1158         this.PropertyChanging(this, emptyChangingEventArgs);
1159     }
1160 }
1161
1162 protected virtual void SendPropertyChanged(String propertyName)
1163 {
1164     if ((this.PropertyChanged != null))
1165     {
1166         this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
1167     }
1168 }
1169
1170 private void attach_DetectionEvents(DetectionEvent entity)
1171 {
1172     this.SendPropertyChanging();
1173     entity.Individual = this;
1174 }
1175
1176 private void detach_DetectionEvents(DetectionEvent entity)

```

```

1177    {
1178        this .SendPropertyChanging();
1179        entity .Individual = null;
1180    }
1181
1182    private void attach_KnownImg (KnownImg entity )
1183    {
1184        this .SendPropertyChanging();
1185        entity .Individual = this ;
1186    }
1187
1188    private void detach_KnownImg (KnownImg entity )
1189    {
1190        this .SendPropertyChanging();
1191        entity .Individual = null;
1192    }
1193}
1194
1195 [global::System.Data.Linq.Mapping.TableAttribute (Name="dbo.KnownImg")]
1196 public partial class KnownImg : INotifyPropertyChanging , INotifyPropertyChanged
1197 {
1198
1199     private static PropertyChangingEventArgs emptyChangingEventArgs = new
1200         PropertyChangingEventArgs (String .Empty);
1201
1202     private int _KnownImgID;
1203
1204     private int _IDImg;
1205
1206     private System .Data .Linq .Binary _Img;
1207
1208     private EntityRef<Individual> _Individual;
1209
1210 #region Extensibility Method Definitions
1211     partial void OnLoaded();
1212     partial void OnValidate (System .Data .Linq .ChangeAction action);
1213     partial void OnCreated();
1214     partial void OnKnownImgIDChanging (int value);
1215     partial void OnKnownImgIDChanged();
1216     partial void OnIDImgChanging (int value);
1217     partial void OnIDImgChanged();
1218     partial void OnImgChanging (System .Data .Linq .Binary value);
1219     partial void OnImgChanged();
1220 #endregion
1221
1222     public KnownImg()
1223     {
1224         this ._Individual = default (EntityRef<Individual>);
1225         OnCreated();
1226     }
1227
1228 [global::System.Data.Linq.Mapping.ColumnAttribute (Storage="_KnownImgID" , AutoSync=
1229     AutoSync.OnInsert , DbType="Int NOT NULL IDENTITY" , IsPrimaryKey=true ,
1230     IsDbGenerated=true)]
1231     public int KnownImgID
1232     {
1233         get
1234         {
1235             return this ._KnownImgID;
1236         }
1237         set
1238         {
1239             if ((this ._KnownImgID != value))
1240             {
1241                 this .OnKnownImgIDChanging (value);
1242                 this .SendPropertyChanging();
1243                 this ._KnownImgID = value;
1244                 this .SendPropertyChanged ("KnownImgID");

```

```

1242         this.OnKnownImgIDChanged();
1243     }
1244 }
1245 }
1246 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _IDImg", DbType=" Int NOT
1247      NULL")]
1248 public int IDImg
1249 {
1250     get
1251     {
1252         return this._IDImg;
1253     }
1254     set
1255     {
1256         if ((this._IDImg != value))
1257         {
1258             if (this._Individual.HasLoadedOrAssignedValue)
1259             {
1260                 throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
1261             }
1262             this.OnIDImgChanging(value);
1263             this.SendPropertyChanging();
1264             this._IDImg = value;
1265             this.SendPropertyChanged("IDImg");
1266             this.OnIDImgChanged();
1267         }
1268     }
1269 }
1270
1271 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage=" _Img", DbType="Image",
1272      UpdateCheck=UpdateCheck.Never)]
1272 public System.Data.Linq.Binary Img
1273 {
1274     get
1275     {
1276         return this._Img;
1277     }
1278     set
1279     {
1280         if ((this._Img != value))
1281         {
1282             this.OnImgChanging(value);
1283             this.SendPropertyChanging();
1284             this._Img = value;
1285             this.SendPropertyChanged("Img");
1286             this.OnImgChanged();
1287         }
1288     }
1289 }
1290
1291 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Individual_KnownImg",
1292      Storage=" _Individual", ThisKey="IDImg", OtherKey="ID", IsForeignKey=true ,
1293      DeleteOnNull=true , DeleteRule="CASCADE")]
1294 public Individual Individual
1295 {
1296     get
1297     {
1298         return this._Individual.Entity;
1299     }
1300     set
1301     {
1302         Individual previousValue = this._Individual.Entity;
1303         if (((previousValue != value)
1304             || (this._Individual.HasLoadedOrAssignedValue == false)))
1305         {
1306             this.SendPropertyChanging();
1307             if ((previousValue != null))

```

```

1306     {
1307         this._Individual.Entity = null;
1308         previousValue.KnownImgs.Remove(this);
1309     }
1310     this._Individual.Entity = value;
1311     if ((value != null))
1312     {
1313         value.KnownImgs.Add(this);
1314         this._IDImg = value.ID;
1315     }
1316     else
1317     {
1318         this._IDImg = default(int);
1319     }
1320     this.SendPropertyChanged("Individual");
1321 }
1322 }
1323 }
1324
1325 public event PropertyChangingEventHandler PropertyChanging;
1326
1327 public event PropertyChangedEventHandler PropertyChanged;
1328
1329 protected virtual void SendPropertyChanging()
1330 {
1331     if ((this.PropertyChanging != null))
1332     {
1333         this.PropertyChanging(this, emptyChangingEventArgs);
1334     }
1335 }
1336
1337 protected virtual void SendPropertyChanged(String propertyName)
1338 {
1339     if ((this.PropertyChanged != null))
1340     {
1341         this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
1342     }
1343 }
1344 }
1345
1346 [global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Anomaly")]
1347 public partial class Anomaly : INotifyPropertyChanging, INotifyPropertyChanged
1348 {
1349
1350     private static PropertyChangingEventArgs emptyChangingEventArgs = new
1351             PropertyChangingEventArgs(String.Empty);
1352
1353     private int _AnomalyID;
1354
1355     private int _EventID;
1356
1357     private System.Nullable<double> _Severity;
1358
1359     private System.Nullable<short> _isChecked;
1360
1361     private System.Nullable<short> _AnomalyType;
1362
1363     private EntityRef<DetectionEvent> _DetectionEvent;
1364
1365     #region Extensibility Method Definitions
1366     partial void OnLoaded();
1367     partial void OnValidate(System.Data.Linq.ChangeAction action);
1368     partial void OnCreated();
1369     partial void OnAnomalyIDChanging(int value);
1370     partial void OnAnomalyIDChanged();
1371     partial void OnEventIDChanging(int value);
1372     partial void OnEventIDChanged();
1373     partial void OnSeverityChanging(System.Nullable<double> value);

```

```

1373 partial void OnSeverityChanged();
1374 partial void OnIsCheckedChanging(System.Nullable<short> value);
1375 partial void OnIsCheckedChanged();
1376 partial void OnAnomalyTypeChanging(System.Nullable<short> value);
1377 partial void OnAnomalyTypeChanged();
1378 #endregion
1379
1380 public Anomaly()
1381 {
1382     this._DetectionEvent = default(EntityRef<DetectionEvent>);
1383     OnCreated();
1384 }
1385
1386 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="AnomalyID", AutoSync=
    AutoSync.OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true,
    IsDbGenerated=true)]
1387 public int AnomalyID
1388 {
1389     get
1390     {
1391         return this._AnomalyID;
1392     }
1393     set
1394     {
1395         if ((this._AnomalyID != value))
1396         {
1397             this.OnAnomalyIDChanging(value);
1398             this.SendPropertyChanging();
1399             this._AnomalyID = value;
1400             this.SendPropertyChanged("AnomalyID");
1401             this.OnAnomalyIDChanged();
1402         }
1403     }
1404 }
1405
1406 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="EventID", DbType="Int NOT
    NULL")]
1407 public int EventID
1408 {
1409     get
1410     {
1411         return this._EventID;
1412     }
1413     set
1414     {
1415         if ((this._EventID != value))
1416         {
1417             if (this._DetectionEvent.HasLoadedOrAssignedValue)
1418             {
1419                 throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
1420             }
1421             this.OnEventIDChanging(value);
1422             this.SendPropertyChanging();
1423             this._EventID = value;
1424             this.SendPropertyChanged("EventID");
1425             this.OnEventIDChanged();
1426         }
1427     }
1428 }
1429
1430 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="Severity", DbType="Float")]
1431 public System.Nullable<double> Severity
1432 {
1433     get
1434     {
1435         return this._Severity;
1436     }

```

```

1437     set
1438     {
1439         if ((this._Severity != value))
1440         {
1441             this.OnSeverityChanging(value);
1442             this.SendPropertyChanging();
1443             this._Severity = value;
1444             this.SendPropertyChanged("Severity");
1445             this.OnSeverityChanged();
1446         }
1447     }
1448 }
1449
1450 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_isChecked", DbType="SmallInt")]
1451 public System.Nullable<short> isChecked
1452 {
1453     get
1454     {
1455         return this._isChecked;
1456     }
1457     set
1458     {
1459         if ((this._isChecked != value))
1460         {
1461             this.OnisCheckedChanging(value);
1462             this.SendPropertyChanging();
1463             this._isChecked = value;
1464             this.SendPropertyChanged("isChecked");
1465             this.OnisCheckedChanged();
1466         }
1467     }
1468 }
1469
1470 [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_AnomalyType", DbType="SmallInt")]
1471 public System.Nullable<short> AnomalyType
1472 {
1473     get
1474     {
1475         return this._AnomalyType;
1476     }
1477     set
1478     {
1479         if ((this._AnomalyType != value))
1480         {
1481             this.OnAnomalyTypeChanging(value);
1482             this.SendPropertyChanging();
1483             this._AnomalyType = value;
1484             this.SendPropertyChanged("AnomalyType");
1485             this.OnAnomalyTypeChanged();
1486         }
1487     }
1488 }
1489
1490 [global::System.Data.Linq.Mapping.AssociationAttribute(Name="DetectionEvent_Anomaly",
1491     Storage="_DetectionEvent", ThisKey="EventID", OtherKey="EventID", IsForeignKey=
1492     true, DeleteOnNull=true, DeleteRule="CASCADE")]
1493 public DetectionEvent DetectionEvent
1494 {
1495     get
1496     {
1497         return this._DetectionEvent.Entity;
1498     }
1499     set
1500     {
1501         DetectionEvent previousValue = this._DetectionEvent.Entity;
1502         if (((previousValue != value)

```

```

1501     || (_this._DetectionEvent.HasLoadedOrAssignedValue == false)))
1502     {
1503         _this.SendPropertyChanging();
1504         if ((previousValue != null))
1505         {
1506             _this._DetectionEvent.Entity = null;
1507             previousValue.Anomalies.Remove(_this);
1508         }
1509         _this._DetectionEvent.Entity = value;
1510         if ((value != null))
1511         {
1512             value.Anomalies.Add(_this);
1513             _this._EventID = value.EventID;
1514         }
1515         else
1516         {
1517             _this._EventID = default(int);
1518         }
1519         _this.SendPropertyChanged("DetectionEvent");
1520     }
1521 }
1522 }
1523 public event PropertyChangingEventHandler PropertyChanging;
1524
1525 public event PropertyChangedEventHandler PropertyChanged;
1526
1527 protected virtual void SendPropertyChanging()
1528 {
1529     if ((this.PropertyChanging != null))
1530     {
1531         this.PropertyChanging(this, emptyChangingEventArgs);
1532     }
1533 }
1534
1535 protected virtual void SendPropertyChanged(String propertyName)
1536 {
1537     if ((this.PropertyChanged != null))
1538     {
1539         this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
1540     }
1541 }
1542 }
1543 }
1544 }
1545 #pragma warning restore 1591

```

Listing 16.34: Code for the Database Configuration File of the User Interface

16.15 Identity Engine - Matlab Demo Code

```

1 function [name, cam, frameNum, action] = identity_engine(frame, camera, xpos, ypos, h, w,
2 img)
3 % input = struct {
4 % int objID,
5 % int frame,
6 % int camera,
7 % int xpos,
8 % int ypos,
9 % int h,
10 % int w,
11 % image img }
12
13 persistent prevIDs;
14 persistent lastCam;
15 persistent lastFrame;

```

```

15 persistent lastPos;
16 delay = 6;
17
18 % read image
19 figure
20 subplot(2,2,1)
imshow(img);
21
22 % create grayscale image
23 g_img = rgb2gray(img);
24 subplot(2,2,2)
imshow(g_img);
25
26
27 % edge detection
28 edgy_img = edge(g_img, 'canny');
29 subplot(2,2,3)
imshow(edgy_img);
30
31
32 % find target area
33 x = w;
34 y = h;
35
36 min_x = x;
37 min_y = y;
38 max_x = 1;
39 max_y = 1;
40
41
42 % find max and min x and y coordinates
43 for j = 1:y
44     for i = 1:x
45         if edgy_img(i,j) == 1
46             if i < min_x
47                 min_x = i;
48             elseif i > max_x
49                 max_x = i;
50             end
51
52             if j < min_y
53                 min_y = j;
54             elseif j > max_y
55                 max_y = j;
56             end
57         end
58     end
59 end
60
61 % MATLAB demo only
62 demo_img = img;
63
64 % find outer radius
65 r = (abs(max_x-min_x)/2 + abs(max_y-min_y)/2)/2;
66 r_out = 0.69 * r;
67 r_mid_out = 0.49 * r;
68 r_mid_in = 0.45 * r;
69 r_in = 0.25 * r;
70
71 for i = 1:1:x
72     for j = 1:y
73         p = sqrt((i-(x/2))^2 + (j-(y/2))^2);
74         if p <= (r_out * 1.01) && p >= (r_out * 0.99)
75             demo_img(i, j, 1) = 255;
76             demo_img(i, j, 2) = 140;
77             demo_img(i, j, 3) = 0;
78         elseif p <= (r_mid_out * 1.01) && p >= (r_mid_out * 0.99)
79             demo_img(i, j, 1) = 255;
80             demo_img(i, j, 2) = 140;
81             demo_img(i, j, 3) = 0;
82         elseif p <= (r_mid_in * 1.01) && p >= (r_mid_in * 0.99)

```

```

83     demo_img(i, j, 1) = 255;
84     demo_img(i, j, 2) = 140;
85     demo_img(i, j, 3) = 0;
86 elseif p <= (r_in * 1.01) && p >= (r_in * 0.99)
87     demo_img(i, j, 1) = 255;
88     demo_img(i, j, 2) = 140;
89     demo_img(i, j, 3) = 0;
90 end
91 end
92
93 subplot(2,2,4)
94 imshow(demo_img);
95
96 % ID color bands
97 % vars for each band: pixel count, sum, avg
98 band1_pix = 0;
99 band1_sum = [0, 0, 0];
100 band1_avg = [0, 0, 0];
101
102 band2_pix = 0;
103 band2_sum = [0, 0, 0];
104 band2_avg = [0, 0, 0];
105
106 for j = 1:y
107     for i = 1:x
108         p = sqrt((i-(x/2))^2 + (j-(y/2))^2);
109         % inside band 1
110         if p <= r_out && p >= r_mid_out
111             band1_pix = band1_pix + 1;
112             band1_sum(1) = band1_sum(1) + cast(img(i,j,1), 'uint32');
113             band1_sum(2) = band1_sum(2) + cast(img(i,j,2), 'uint32');
114             band1_sum(3) = band1_sum(3) + cast(img(i,j,3), 'uint32');
115         % inside band 2
116         elseif p <= r_mid_in && p >= r_in
117             band2_pix = band2_pix + 1;
118             band2_sum(1) = band2_sum(1) + cast(img(i,j,1), 'uint32');
119             band2_sum(2) = band2_sum(2) + cast(img(i,j,2), 'uint32');
120             band2_sum(3) = band2_sum(3) + cast(img(i,j,3), 'uint32');
121         end
122     end
123 end
124
125 band1_avg(1) = cast(ceil(band1_sum(1) / band1_pix), 'uint8');
126 band1_avg(2) = cast(ceil(band1_sum(2) / band1_pix), 'uint8');
127 band1_avg(3) = cast(ceil(band1_sum(3) / band1_pix), 'uint8');
128 band2_avg(1) = cast(ceil(band2_sum(1) / band2_pix), 'uint8');
129 band2_avg(2) = cast(ceil(band2_sum(2) / band2_pix), 'uint8');
130 band2_avg(3) = cast(ceil(band2_sum(3) / band2_pix), 'uint8');
131
132 band1 = [0, 0, 0];
133 band2 = [0, 0, 0];
134
135 % threshold
136 for i = 1:3
137     if band1_avg(i) <= 32
138         band1(i) = 0;
139     elseif band1_avg(i) > 32 && band1_avg(i) <= 96
140         band1(i) = 64;
141     elseif band1_avg(i) > 96 && band1_avg(i) <= 160
142         band1(i) = 128;
143     elseif band1_avg(i) > 160 && band1_avg(i) <= 224
144         band1(i) = 192;
145     elseif band1_avg(i) > 224
146         band1(i) = 255;
147     end
148
149     if band2_avg(i) <= 32

```

```

151         band2(i) = 0;
152     elseif band2_avg(i) > 32 && band2_avg(i) <= 96
153         band2(i) = 64;
154     elseif band2_avg(i) > 96 && band2_avg(i) <= 160
155         band2(i) = 128;
156     elseif band2_avg(i) > 160 && band2_avg(i) <= 224
157         band2(i) = 192;
158     elseif band2_avg(i) > 224
159         band2(i) = 255;
160     end
161 end
162
163 name = strcat(dec2hex(band1(1)), dec2hex(band1(2)), dec2hex(band1(3)), dec2hex(band2
164     (1)), dec2hex(band2(2)), dec2hex(band2(3)));
165 cam(1) = camera;
166 frameNum(1) = frame;
167
168 prev = 0;
169 for i = 1:size(prevIDs)
170     if strcmp(name, prevIDs(i)) == 1
171         prev = 1;
172         lastFrame(i) = frame;
173         lastPos(i) = [xpos, ypos];
174     end
175 end
176 if prev == 0
177     prevIDs = [prevIDs, cellstr(name)];
178     lastCam = [lastCam, camera];
179     lastFrame = [lastFrame, frame];
180     lastPos = [lastPos, [xpos, ypos]];
181 end
182
183 frame_size = [1280, 800];
184 % 0 = entered normally
185 % 1 = exited normally
186 % 2 = exited abnormally
187 % 3 = entered abnormally
188 if (xpos <= frame_size(1) * 0.15 || xpos >= frame_size(1) * 0.85) && prev == 0
189     action(1) = 0;
190 elseif (xpos >= frame_size(1) * 0.15 || xpos <= frame_size(1) * 0.85) && prev == 0
191     action(1) = 3;
192 end
193
194 for i = 1:size(prevIDs)
195     if (lastPos(i,1) <= frame_size(1) * 0.15 || lastPos(i,1) >= frame_size(1) * 0.85)
196         && frame == lastFrame(i) + delay
197         name = [name, cellstr(prevIDs(i))];
198         cam = [cam, lastCam(i)];
199         frameNum = [frameNum, lastFrame(i)];
200         action = [action, 1];
201     elseif (lastPos(i,1) >= frame_size(1) * 0.15 || lastPos(i,1) <= frame_size(1) *
202         0.85) && frame == lastFrame(i) + delay
203         name = [name, cellstr(prevIDs(i))];
204         cam = [cam, lastCam(i)];
205         frameNum = [frameNum, lastFrame(i)];
206         action = [action, 2];
207     end
208 end
209 end

```

Listing 16.35: Object Identification Demo

Chapter 17

Appendix - FPGA Lab Notebook

FPGA Flow

Thursday, December 26, 2013 3:30 PM

Device Tree (From Quartus) specifies memory maps and which drivers to use
In example altera-gpio and legs-gpio are already packaged with linux shource AND NOT in a bitbake
recepie
Bitbake takes the device tree -> /meta-altera/recipes-gsrd/linux/files/socfpga.dts
The "compatable" attribute sets driver compatibility
Shared Memory (FPGA Memory Accessed over h2f_axi) does not appear to be included in device
tree (no driver needed)?
Looks like you can add driver using bitbake recipie

Notation:

H2f_lw_axi - HPS to FPGA Lightweight AXI
H2f_axi - HPS to FPGA (High Bandwidth) AXI

From the description in the GHRD Description, the onchip ram is used as an "scratch pad" and serves
only as a reference on how to use the high bandwith bridge and how to use FPGA RAM, it is directly
accessible to the processor as "
memory map of soft IP peripherals, as viewed by the microprocessor unit (MPU), starts at HPS-to-
FPGAAddress offset 0xC000_0000"

From <<http://rocketboards.org/foswiki/Documentation/GSRDGhrd>>

The HPS therefore, does not have access to the entire memory space of the FPGA ram (only 3FFF_FFFF

Investigate

Thursday, December 26, 2013 3:31 PM

Decode image with linux.
Copy to memory which is shared with FPGA
FPGA fetches images from memory
FPGA puts result in shared memory

See if we can use DMA controller

Linux Libraries

Thursday, December 26, 2013 3:33 PM

FFMPEG - supports H.264 and M-JPEG. Supports RTSP

Libav - supports H.264 and M-JPEG. Supports RTSP
Has OpenEmbedded recipe -> migrated to OpenEmbedded core
Mentioned in <http://libav.org/download.html>

Formats supported in <http://libav.org/general.html>

VLC uses Libav

Ubuntu has switched to Libav

Both have very similar suites of programs

Libav is a fork of FFMPEG

Will test them in Ubuntu VM with camera

Command Info:

<http://libav.org/avplay.html#rtsp> - contains both avconv and avplay commands

	h.264	Mjpeg
Libav	<pre>christopher@seniorDesignUbuntuLTS:~/linux-altera-3.7/drivers/leds\$ avplay -max_delay 500000 -rtsp_transport udp rtsp://192.168.1.132:554/live1.sdp avplay version 0.8.9-4:0.8.9-0ubuntu0.12.04.1, Copyright (c) 2003-2013 the Libav developers built on Nov 9 2013 19:08:00 with gcc 4.6.3 [rtsp @ 0x7f95580008c0] Estimating duration from bitrate, this may be inaccurate Input #0, rtsp, from 'rtsp://192.168.1.132:554/live1.sdp': Metadata: title : RTSP/RTP stream 1 from DCS-2310L comment : live1.sdp with v2.0 Duration: N/A, start: 0.066667, bitrate: N/A Stream #0.0: Video: h264 (High), yuvj420p, 1280x800, 15 fps, 1k tbr, 90k tbn, 30 tbc Stream #0.1: Audio: pcm_mulaw, 8000 Hz, 1 channels, s16, 64 kb/s [avsink @ 0x7f954c001020] auto-inserting filter 'auto-inserted scaler 0' between the filter 'src' and the filter 'out' [scale @ 0x7f954c001700] w:1280 h:800 fmt:yuvj420p -> w:1280 h:800 fmt:yuv420p flags:0x4 46.26 A-V: -0.067 s:0.0 aq= 26KB vq= 1077KB sq= 0B f=0/0 f=0/0 christopher@seniorDesignUbuntuLTS:~/linux-altera-3.7/drivers/leds\$</pre>	ERROR - expects 8 bit, is given 16 bit UNSUPPORTED
FFMPEG	Ubuntu hard wired ffmpeg to Libav	

Libav can transcode RTSP to image files!!!! - including uncompressed using avconv
Filename needs to be in printf format!!! If encoding to images
<http://stackoverflow.com/questions/9271357/ffmpeg-on-fb0-from-nexus-galaxy-error-could-not-get-frame-filename-number-2>

-vframes sets number of frames to convert
-vcodec codec force video codec ('copy' to copy stream)
-f fmt force format
-an (output)
Disable audio recording.
-threads sets the number of threads to be used

To fix bad frames being written, <http://www.eonlinegratis.com/2013/avconv-ffmpeg-on-raspberry-pi-often-corrupts-extracted-video-still-images/#sthash.q3OIJjKI.dpbs>
-fflags discardcorrupt

```
avconv -max_delay 500 -rtsp_transport udp -fflags discardcorrupt -i  
rtsp://192.168.1.132:554/live1.sdp -fflags discardcorrupt -an testimg_%010d.bmp
```

You can extract images from a video, or create a video from many images:

For extracting images from a video:

```
avconv -i foo.avi -r 1 -s WxH -f image2 foo-%03d.jpeg
```

This will extract one video frame per second from the video and will output them in files named foo-001.jpeg, foo-002.jpeg, etc. Images will be rescaled to fit the new WxH values.

If you want to extract just a limited number of frames, you can use the above command in combination with the -vframes or -t option, or in combination with -ss to start extracting from a certain point in time.

For creating a video from many images:

```
avconv -f image2 -i foo-%03d.jpeg -r 12 -s WxH foo.avi
```

The syntax "foo-%03d.jpeg" specifies to use a decimal number composed of three digits padded with zeroes to express the sequence number. It is the same syntax supported by the C printf function, but only formats accepting a normal integer are suitable.

image2

Image file muxer.

The image file muxer writes video frames to image files.

The output filenames are specified by a pattern, which can be used to produce sequentially numbered series of files. The pattern may contain the string "%d" or "%0Nd", this string specifies the position of the characters representing a numbering in the filenames. If the form "%0Nd" is used, the string representing the number in each filename is 0-padded to N digits. The literal character '%' can be specified in the pattern with the string "%%".

If the pattern contains "%d" or "%0Nd", the first filename of the file list specified will contain the

number 1, all the following numbers will be sequential.

The pattern may contain a suffix which is used to automatically determine the format of the image files to write.

For example the pattern "img-%03d.bmp" will specify a sequence of filenames of the form img-001.bmp, img-002.bmp, ..., img-010.bmp, etc. The pattern "img%-%d.jpg" will specify a sequence of filenames of the form img%-1.jpg, img%-2.jpg, ..., img%-10.jpg, etc.

The following example shows how to use avconv for creating a sequence of files img-001.jpeg, img-002.jpeg, ..., taking one image every second from the input video:

```
avconv -i in.avi -vsync 1 -r 1 -f image2 'img-%03d.jpeg'
```

Note that with avconv, if the format is not specified with the "-f" option and the output filename specifies an image file format, the image2 muxer is automatically selected, so the previous command can be written as:

```
avconv -i in.avi -vsync 1 -r 1 'img-%03d.jpeg'
```

Note also that the pattern must not necessarily contain "%d" or "%0Nd", for example to create a single image file img.jpeg from the input video you can employ the command:

```
avconv -i in.avi -f image2 -frames:v 1 img.jpeg
```

Good Command

Friday, December 27, 2013 11:01 PM

```
avconv -max_delay 500 -rtsp_transport tcp -fflags discardcorrupt -i rtsp://192.168.1.132:554/live2.sdp -  
fflags discardcorrupt -an testimg_%010d.bmp
```

-threads sets the number of threads to be used

This work for the most part because feed 2 was set to 15 frames/sec. If higher than that, libav cannot keep up and there are errors. UDP is often responsible for lost packets (and therefore corruption). This is ok for humans but it is bad for computer vision

HOWEVER UDP has less latency! And that may be important for us

To record H.264 video in same format (no transcoding) (NO ERRORS b/c TCP):

Errors can occur if UDP is used (due to packet loss) Error appears in playback - it tries its best to
recover

```
avconv -max_delay 500 -rtsp_transport tcp -fflags discardcorrupt -i rtsp://192.168.1.132:554/live1.sdp -  
fflags discardcorrupt -an -vcodec copy test.mov
```

To convert H.264 video to bmps. NO ERRORS!!!!!!

```
avconv -fflags discardcorrupt -i ./test.mov -fflags discardcorrupt -an testimg_%010d.bmp
```

Playing video

```
avplay test.mov
```

Gstremaer

Thursday, December 26, 2013 10:47 PM

<http://stackoverflow.com/questions/4192871/playing-rtsp-with-python-gstreamer>

Does not work to my knowlege

Using MJPG

Monday, January 6, 2014 11:35 PM

SO, After days of digging, it was revealed in an e-mail that the RTSP/RTP demuxer does not support MJPG!!!!!!

Why can VLC play MJPEG over RTP/RTSP you may ask, because it uses a library called live555.

Live555 is capable of capturing the MJPEG stream and recording it (very clearly I may add - better than even libav

I have tried it in ubuntu and not on the board yet. However, not all is lost! You still need libav to PLAY the recorded MJPEG which avplay does quite well if it has the mjpg extension.

I had to force RTSP/RTP over TCP because of the school firewall. Apparently, this was causing the zero-size files I was seeing.

The commands are as follows: (the -t forces TCP, -d specifies how long to record)

```
openRTSP -t -d 10 rtsp://172.16.122.113:554/live2.sdp  
mv video-JPEG-1 video-JPEG-1.mjpg  
avplay video-JPEG-1.mjpg
```

Can only record video using

```
openRTSP -t -v -d 10 -F mjpg-video rtsp://172.16.122.113:554/live2.sdp > mjpeg-vid.mjpg
```

I will recompile and try it on the board

Live555 is a library and I should be able to extract raw data from it

<http://live555.com/liveMedia/> is an example media player

Live555 has an OE recipie that is in the meta-oe repo. There is also a recipie in the oe-classic repo

<http://layers.openembedded.org/layerindex/recipe/1176/>

It work on the board!!!!!!

Is able to record MJPG at 30 FPS!!!

However, FPS drops the longer the capture runs. But no errors. Unlike libav which corrupts images in an attempt to catch up, live555 drops frames to keep up

I think this can be fixed by writing custom program that only uses ram. I think that the bottleneck is writing file to SDcard

-m option saves each frame as a separate file!!!!

Yacto:

Thursday, December 26, 2013 4:16 PM

How to include a recipe [https://wiki.yoctoproject.org/wiki/How do I#Q:
How do I put my recipe into Yocto.3F](https://wiki.yoctoproject.org/wiki/How_do_I#Q:_How_do_I_put_my_recipe_into_Yocto.3F)

Patches: [https://wiki.yoctoproject.org/wiki/Recipe %26 Patch Style Guide](https://wiki.yoctoproject.org/wiki/Recipe_%26_Patch_Style_Guide)

[https://wiki.yoctoproject.org/wiki/Recipe %26 Patch Style Guide](https://wiki.yoctoproject.org/wiki/Recipe_%26_Patch_Style_Guide)

Yacto has a <http://en.wikipedia.org/wiki/GStreamer> recipie which relies on FFMPEG libs

oe-core has libav

Need to find how to import oe-core layer into yacto
[http://www.openembedded.org/wiki/Getting started](http://www.openembedded.org/wiki/Getting_started)

Error with license

<https://community.freescale.com/thread/316329>

Steps:

1. Cd ~
2. Mkdir oe
3. Cd oe
4. git clone git://git.openembedded.org/openembedded-core oe-core
5. Copy oe/oe-core/meta/recipes-multimedia/libav to yocto-ghrd/meta/recipes-multimedia
6. Add libav to ~/yocto-ghrd/meta-altera/recipes-core/images/altera-gsrd-image.bb
7. Will get license error
8. Append LICENSE_FLAGS_WHITELIST = "commercial" to ~/yocto-ghrd/build/conf/local.conf
(<https://community.freescale.com/thread/316329>)
9. There will be a dependence error with yasm
10. Copy ~/oe/oe-core/meta/recipes-devtools/yasm to ~/yocto-ghrd/meta/recipes-devtools
(<http://layers.openembedded.org/layerindex/recipe/5638/>)
11. Will get dependence error with x264
12. Copy ~/oe/oe-core/meta/recipes-multimedia/x264 to ~/yocto-ghrd/meta/recipes-multimedia
(<http://layers.openembedded.org/layerindex/recipe/5642/>)
13. source /home/christopher/yocto-ghrd/altera-init /home/christopher/yocto-ghrd/build
14. bitbake virtual/bootloader virtual/kernel altera-gsrd-image

Adding live555 library

Live555 has a recipie in meta

1. Cd ~/oe
2. git clone git://git.openembedded.org/meta-openembedded meta-oe
3. cd ~/yocto-ghrd/
4. Mkdir meta-oe
5. Cd meta-oe
6. cp -r ~/oe/meta-oe/meta-oe/conf/ .
7. cp -r ~/oe/meta-oe/meta-oe/licenses/ .
8. cp -r ~/oe/meta-oe/meta-oe/COPYING.MIT .
9. cp -r ~/oe/meta-oe/meta-oe/README .
10. mkdir recipes-multimedia
11. Cd recipes-multimedia

12. cp -r ~/oe/meta-oe/meta-oe/recipes-multimedia/live555/ .
13. Insert /home/christopher/yocto-ghrd/meta-oe \ into ~/yocto-ghrd/build/conf/bblayers.conf so that the file looks like:

```
/home/christopher/yocto-ghrd/meta-oe \  
/home/christopher/yocto-ghrd/meta \  
/home/christopher/yocto-ghrd/meta-yocto \  

```

14. In ~/yocto-ghrd/meta-oe/conf/layer.conf , modify the priority value to 1 (we want these packages to have the lowest precedent since we are integrating them into yacto and some may be incompatible)
15. Add live555 live555-openrtsp live555-mediaserver to ~/yocto-ghrd/meta-altera/recipes-core/images/altera-gsrd-image.bb
16. source /home/christopher/yocto-ghrd/altera-init /home/christopher/yocto-ghrd/build
17. bitbake virtual/bootloader virtual/kernel altera-gsrd-image
18. NOTE: I modified the Live555 bb recipie to get the current version as it is what the current example files are based on.

Camera Feed

Friday, December 27, 2013 10:33 PM

RTP over RTSP

Bitbake Add Layer

Saturday, December 28, 2013 1:25 AM

Layers are listed in `~/yocto-ghrd/build/conf/bblayer.conf`

http://www.openembedded.org/wiki/Creating_a_new_Layer

<http://www.yoctoproject.org/docs/current/dev-manual/dev-manual.html#prioritizing-your-layer>

`~/yocto-ghrd/build/conf/local.conf` has other parameters

How many threads bitbake uses

The machine (fpga type)

Distro name

Image config options

Bitbake clean

Saturday, December 28, 2013 3:48 PM

```
bitbake -c clean TARGET
```

Making SD Card

Saturday, December 28, 2013 5:16 PM

Since Using older version of Yocto, instructions on
<http://www.rocketboards.org/foswiki/Documentation/GSRDSdCard> are not quite right
The help text given is not the same as the help text with the SD Card image script in the program

Use instead!!!!

<http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGettingStarted>

```
$ cd ~/yocto-ghrd/build/tmp/deploy/images
$ sudo rm -rf rootfs
$ mkdir rootfs
$ cd rootfs
$ sudo tar xzf ../altera-gsrd-image-socfpga_cyclone5.tar.gz
$ cd ..
$ rm sd_image_yocto.bin
$ sudo /opt/altera-linux-ghrd/bin/make_sdimage.sh \
-k ulmage,socfpga.dtb \
-rp u-boot-spl-socfpga_cyclone5.bin \
-t ~/altera/13.0sp1/embedded/host_tools/altera/mkpimage/mkpimage \
-b u-boot-socfpga_cyclone5.img \
-r rootfs/ \
-o sd_image_yocto.bin
```

From <<http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGettingStarted>>

Documentation to flash image to disk is at
<http://www.rocketboards.org/foswiki/Documentation/GSRDBootLinuxSd>

Instead of downloading image file, copy the built one from ~/yocto-
ghrd/build/tmp/deploy/images/altera-gsrd-image-socfpga_cyclone5.tar.gz

```
//to list devices
cat /proc/partitions

sudo dd if=sd_image_yocto.bin of=/dev/sdb bs=1M
sudo sync
```

Printout

Saturday, December 28, 2013 6:56 PM

Linux Getting Started on Altera SoC Development Board - Using Yocto Source Package

Last modified by [John Carter](#) on 26 Nov 2013 - 00:39 - 19
Tagged [Kernel](#), [Linux](#), [Yocto](#)

Table of Contents

Introduction	Obtaining Yocto Source Package
Host Setup	Setting Up Yocto
CentOS 6.3	Building U-Boot/Kernel/Rootfs
Ubuntu 12.04	Creating SD Card Image
Ubuntu 13.10 - Saucy Salamander	Using SD Card Image
SoC EDS	Partially Updating SD Card

Introduction

This page presents the instructions on how to rebuild Linux by using the Yocto Source Package.

The Yocto Source package is an installer file provided by Altera that contains the Yocto build system, Yocto recipes and also the necessary dependencies to compile the Altera Linux bootloader, kernel and root filesystem.

Note: the Yocto build system and recipes for Altera Linux can be also obtained from the Git trees. See [Linux - Getting Started Using Git Trees](#) section for details on that. If the Git tree option is used, all the necessary dependencies would be downloaded from Internet.

For more details about Yocto see [Altera Yocto Project User Manual \(Danny\)](#).

Host Setup

The Altera Linux Yocto Source package was tested to build correctly with CentOS 6.3 and Ubuntu 12.04. It may also work with other distributions. For a list of distributions against which Yocto project was tested see [https://wiki.yoctoproject.org/wiki/Distribution_Support](#). This section presents the packages that need to be installed on the host PC to allow the Yocto Source Package to be built. Since each machine may have been installed in a different way, the packages listed here are not an exhaustive list. Add packages as necessary if you encounter issues.

CentOS 6.3

These are the required packages that need to be installed:

```
$ sudo yum update
$ sudo yum groupinstall "Development Tools"
$ sudo yum install texi2html texinfo glibc-devel chrpath
```

If the host machine runs the 64bit version of the OS, then the following additional packages need to be installed:

```
$ sudo yum install glibc.i686 libgcc.i686 libstdc++.i686 glibc-devel.i686 ncurses-libs.i686 zlib.i686
```

Ubuntu 12.04

These are the required packages that need to be installed:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install sed wget cvs subversion git-core coreutils unzip texi2html texinfo libsdll1.2-dev docbook-ut
ils gawk python-pyslqi* diffstat help2man make gcc build-essential g++ desktop-file-utils chrpath libgl1-mesa-de
v libglu1-mesa-dev mercurial autoconf groff libtool xterm
```

If the host machine runs the 64bit version of the OS, then the following additional packages need to be installed:

```
$ sudo apt-get install ia32-libs
```

Ubuntu 13.10 - Saucy Salamander

As per 12.04 above, except ia32-libs no longer exists in Saucy.

```
sudo apt-get install libqt4-gui:i386 ia32-libs lib32-dev:i386 libgomp1:i386 libqtwebkit4:i386 libtbb2:i386 tcl8.5-lib:i386 tcl8.5-dev:i386
```

The above list was guessed from the output of ldd and apt-file

SoC EDS

The Altera SoC EDS toolset needs to be installed, since it provides the mkpimage tool that adds the BootROM signature on top of the Preloader image.

The SoC EDS can be obtained from [https://www.altera.com/download/software/soc-eds](#). If you do not have a license, follow the instructions from the mentioned web page to obtain the free **Web Edition** license.

Obtaining Yocto Source Package

The Altera Yocto Source Package can be downloaded from the Altera website. The following commands can be used to download the package and mark it as executable:

↑

<http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGettingStarted> 12/28/2013

```
$ wget http://download.altera.com/akdlm/software/acdsinst/13.0/156/ib_installers/linux-socfpga-13.02-RC10-src.bsx
$ chmod +x linux-socfpga-13.02-RC10-src.bsx
```

Setting Up Yocto

The Yocto Source Package may be installed in a publicly accessible location, as this step can be shared by all the users on the system (or on the filesystem if your company uses a network share). The default location is `/opt/altera-linux`. If you wish to use this location, you will likely need root access in order to access this directory. This is why the command shown below is ran using `sudo`.

```
$ sudo ./linux-socfpga-13.02-RC10-src.bsx
```

The next step is to install a local set of Yocto recipes. This could be done in a shared location, but if someone wants or needs to modify them they should have their own version. The default install location for this is within your home directory:

```
$ /opt/altera-linux/bin/install_altera_socfpga_src.sh ~/yocto
```

Last step is to create a build directory. By keeping this separate from your yocto source you can erase your entire build without fear of deleting your yocto sources. Also, you can have several build directories, each with its own configuration, all based on the same yocto source. The script serves 2 purposes. First, it creates the new build directory using Altera's default configuration. Secondly, it sets some shell variables that are required for building.

```
$ source ~/yocto/altera-init ~/yocto/build
```

Note: If you start a new shell you will need to run the above command to set the shell variables again.

Building U-Boot/Kernel/Rootfs

In order to build u-boot from within the build directory you need to execute this command:

```
$ bitbake virtual/bootloader
```

or the equivalent:

```
$ bitbake u-boot
```

In order to build linux from within the build directory:

```
$ bitbake virtual/kernel
```

or the equivalent:

```
$ bitbake linux-altera
```

In order to build the root filesystem:

```
$ bitbake altera-image
```

If you wish to build a much smaller filesystem for small boot mediums:

```
$ bitbake altera-image-minimal
```

The first time you build the Yocto Source Package it may take up to several hours depending on your host machine.

Once finished, all images should be generated in `~/build/tmp/deploy/images`. Some of the most important files created are:

File	Description
u-boot-spl-socfpga_cyclone5	SPL Preloader ELF file
u-boot-spl-socfpga_cyclone5.bin	SPL Preloader Binary
u-boot-socfpga_cyclone5	U-Boot ELF file
u-boot-socfpga_cyclone5.bin	U-Boot binary file
u-boot-socfpga_cyclone5.img	U-Boot image
socfpga_cyclone5.dtb	Device Tree Binary
vmlinu	Kernel ELF file
uimage	Kernel Image
altera-image-socfpga_cyclone5.cpio	Root filesystem in cpio archive format
altera-image-socfpga_cyclone5.ext3	Root filesystem as ext3 image
altera-image-socfpga_cyclone5.jffs2	Root filesystem as jffs2 image
altera-image-minimal-socfpga_cyclone5.jffs2	Minimal root filesystem as jffs2 image
altera-image-socfpga_cyclone5.tar.gz	Root filesystem in tar gzip archive format

↑

<http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGettingStarted> 12/28/2013

Linux Getting Started on Altera SoC Development Board - Using Yocto Source Package... Page 3 of 4

The above files are actually links to the actual build resulted files. Each build file has a timestamp attached to its name and each time it is rebuilt, Yocto updates the link to point to the latest file.

Creating SD Card Image

To boot the linux images on SoC FPGA development kit, you need to write the images you just built with Yocto into one of the two Flash devices: SDMMC or QSPI. For this guide, we will use SDMMC due to its easy detachability. For SDMMC boot, all boot images will be located inside SD/MMC card. A script is provided with the release that will create an SD card image, ready to be deployed.

The script relies on a tool, named `mkpimage`, which creates the correct preloader image that the SoC FPGA Boot ROM accepts. This tool is provided with the SoC EDS release. See [SoC EDS](#) section for instructions.

After SoC EDS is installed, run the following commands to locate `mkpimage`:

```
$ ~/altera/13.0/embedded/embedded_command_shell.sh  
$ which mkpimage
```

The following files resulted from the Yocto build will be used, from `~/yocto/build/tmp/deploy/images`:

File	Description
<code>u-boot-socfpga_cyclone5.bin</code>	SPL Preloader Binary
<code>u-boot-socfpga_cyclone5.img</code>	U-Boot Image
<code>socfpga_cyclone5.dtb</code>	Device Tree Binary
<code>ulmage</code>	Kernel Image
<code>altera-image-socfpga_cyclone5.tar.gz</code>	Root File System

The following commands can be used to create the SD card image:

```
$ cd ~/yocto/build/tmp/deploy/images  
$ cp -f socfpga_cyclone5.dtb socfpga.dtb  
$ mkdir rootfs  
$ cd rootfs  
$ sudo tar xzf ../../altera-image-socfpga_cyclone5.tar.gz  
$ cd  
$ sudo /opt/altera-linux/bin/make_sdimage.sh  
-k ulmage,socfpga.dtb  
-rp u-boot-spl-socfpga_cyclone5.bin  
-t /home/<user>/altera/13.0/embedded/host_tools/altera/mkpimage/mkpimage  
-b u-boot-socfpga_cyclone5.img  
-r rootfs/ \  
-o sd_image_yocto.bin
```

The above commands do the following:

- Make a copy of the `socfpga_cyclone5.dtb` as `socfpga.dtb` because that is the name that U-boot expects to see
- Extracts the root file system into a folder, using `sudo` to be able to create the device nodes
- Invokes the `make_sdimage.sh` script to create the SD card image named `sd_image_yocto.bin`. This is also executed using `sudo` because it uses system utilities.

Notes:

- The above commands assume default installation paths were used for all tools. Update accordingly if non-default paths were used.
- Please replace `<user>` with your user name.

You can also run `make_sdimage.sh` as follows, without parameters, to obtain more information about the tool:

```
$ /opt/altera-linux/bin/make_sdimage.sh
```

Using SD Card Image

See [Linux Getting Started on Altera SoC Development Board - Using SD Card Image](#) for instructions on how to use the SD card image to boot Linux. Simply use the newly created SD card image instead of the precompiled one.

Partially Updating SD Card

This section presents how to update various parts on the SD card, without having to re-create and write to the SD card the whole image. You will first need to have a working SD card before being able to run the commands presented in this section.

The commands presented assume the SD card device is `/dev/sdx`. Refer to [Linux Getting Started on Altera SoC Development Board - Using SD Card Image](#) for instructions on how to determine which device corresponds to the SD card reader on your host PC.

Updating SPL Preloader:

```
$ sudo dd if=u-boot-spl-socfpga_cyclone5.img of=/dev/sdx3 bs=64k seek=0
```

Updating U-boot Image

```
$ sudo dd if=u-boot-socfpga_cyclone5.img of=/dev/sdx3 bs=64K seek=4
```

↑

<http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGettingStarted> 12/28/2013

Linux Getting Started on Altera SoC Development Board - Using Yocto Source Package... Page 4 of 4

Updating Device Tree Binary:

```
$ sudo mkdir sdcard  
$ sudo mount /dev/sdx1 sdcard/  
$ sudo cp socfpga_cyclone5.dtb sdcard/socfpga.dtb  
$ sudo umount sdcard/
```

Updating Kernel Image:

```
$ sudo mkdir sdcard  
$ sudo mount /dev/sdx1 sdcard/  
$ sudo cp uimage-socfpga_cyclone5.bin sdcard/uImage  
$ sudo umount sdcard/
```

Updating Root Filesystem:

```
$ sudo dd if=altera-image-socfpga_cyclone5.ext3 of=/dev/sdx2
```

↑

<http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGettingStarted> 12/28/2013

Results on Altera Board

Sunday, December 29, 2013 1:05 AM

H.264 Recording works with aconv!!!!

Get many glitches in using 30 pfs stream (feed1)

Using feed 2 at 15 fps is much better (but still has some gliches)

Suggest using only 15 fps stream

BMP transcode did not work well even at 15 fps (or with 2 threads)

JPG transcode a bit better but still incredibly error prone (9 fps max and routiennly goes to 6)

Problem is that FPS being transcoded is slower than incoming resulting in a backlog

Will NEED to write software using the libav libraries that decodes individual frames and places them in shared mem (hopefully not having to re-encode will make things faster)

Able to avoid temporarily by encoding to jpg and having strem capped at 4 fps

Decoding of pre-recorded h.264 to jpg ranged from 6 to 10 fps

Investigate saving raw image (avoid re-encoding)

Recording raw MJPEG works well with Live555, even at 30 FPS

Avplay can be used to play and avconv can be used to transcode to bitmaps. The speed of transcoding to BMP degrades over time (down to 4 FPS)

The speed of transcoding to JPG is stuck at 9 FPS

I believe this is due to a bottleneck in writing to the SD Card. For only 3 sec of video, transcoding to BMP takes ~300 MB!!! Each BMP is 4.6 MB

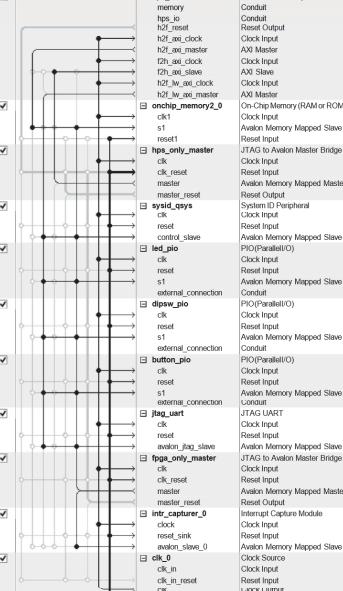
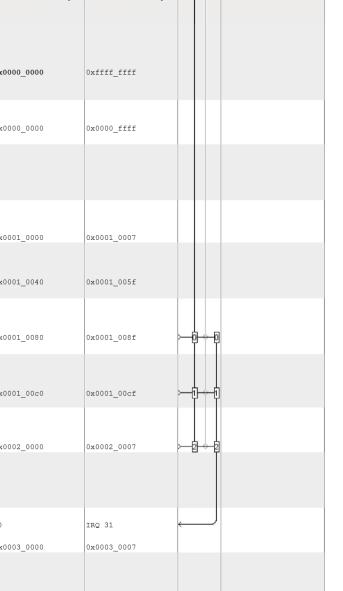
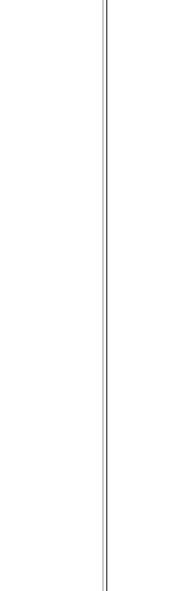
Is able to record MJPG at 30 FPS!!!

However, FPS drops the longer the capture runs. :(But no errors. Unlike libav which corrupts images in an attempt to catch up, live555 drops frames to keep up

I think this can be fixed by writing custom program that only uses ram. I think that the bottleneck is writing file to SDcard

Printout

Friday, January 3, 2014 3:53 PM

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Opcode Name
<input checked="" type="checkbox"/>		hps_0	Hard Processor System						
		memory	Conduit						
		hps_0_axi_reset	Reset Output						
		h2t_axi_clock	Clock Input						
		l2h_axi_clock	Clock Input						
		l2h_axi_slave	Axi Slave						
		l2h_axi_clock	Clock Output						
		h2t_axi_master	Axi Master						
<input checked="" type="checkbox"/>		onchip_memory2_0	On-Chip Memory(RAM or ROM)						
		clk1	Clock Input						
		s1	Avalon Memory Mapped Slave						
		reset1	Reset Input						
<input checked="" type="checkbox"/>		hps_only_master	JTAG to Avalon Master Bridge						
		clk	Clock Input						
		clk_reset	Reset Input						
		master	Avalon Memory Mapped Master						
		master_reset	Reset Output						
<input checked="" type="checkbox"/>		sysid_psys	System ID Peripheral						
		clk	Clock Input						
		reset	Reset Input						
		s1	Avalon Memory Mapped Slave						
<input checked="" type="checkbox"/>		led_pio	PIO(ParallelIO)						
		clk	Clock Input						
		reset	Reset Input						
		s1	Avalon Memory Mapped Slave						
<input checked="" type="checkbox"/>		dipsw_pio	PIO(ParallelIO)						
		clk	Clock Input						
		reset	Reset Input						
		s1	Avalon Memory Mapped Slave						
<input checked="" type="checkbox"/>		button_pio	PIO(ParallelIO)						
		clk	Clock Input						
		reset	Reset Input						
		s1	Avalon Memory Mapped Slave						
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART						
		clk	Clock Input						
		reset	Reset Input						
		s1	Avalon Memory Mapped Slave						
<input checked="" type="checkbox"/>		fpga_only_master	JTAG to Avalon Master Bridge						
		clk	Clock Input						
		clk_reset	Reset Input						
		master	Avalon Memory Mapped Master						
		master_reset	Reset Output						
<input checked="" type="checkbox"/>		intr_capture_0	Interrupt Capture Module						
		clock	Clock Input						
		reset_sink	Reset Input						
		averton_stew_0	Avalon Memory Mapped Slave						
<input checked="" type="checkbox"/>		clk_p	Clock Source						
		clk_in	Clock Input						
		clk_in_reset	Reset Input						
		clk_reset	Reset Output						
		clk	clock						
		reset	reset						

Patching

Thursday, January 9, 2014 11:24 PM

Patched Live555 example to up buffer size in testRTSPClient.cpp

See .patch file for example

Live555 Info

Thursday, January 9, 2014 11:25 PM

Buffer had to be increased in demo program to handle MJPEG. Did this with a patch.

Can get timestamp for received frame!!!!

Also, -m option saves each frame as a file!!!!

Decoding Video to Uncompressed Frame

Wednesday, January 15, 2014 12:23 AM

Libjpeg can be used to decode jpeg images which live555 can create from MJPEG streams. H.264 requires Libav to decode

Libav decoding tutorial: http://www.inb.uni-luebeck.de/~boehme/using_libavcodec.html

Decoding JPEG

Friday, January 17, 2014 11:20 PM

There are 2 commonly used libraries for jpeg decoding, one is a fork of the other and both have oe recipes.

Libjpeg - the original library

Libjpeg-turbo - optimized library for SIMD including NEON (ARM)

Recipes can be found at

<http://layers.openembedded.org/layerindex/branch/master/recipes/?q=libjpeg>

Libjpeg - <http://layers.openembedded.org/layerindex/recipe/606/> (part of oe core)

Libjpeg-turbo - <http://layers.openembedded.org/layerindex/recipe/1139/> (part of meta-oe)

Libjpeg-turbo reports to have 2-4x performance increase over traditional library

Will probably go with libjpeg-turbo implementation due to performance and fact that ARM processor has NEON

Another bonus is that libjpeg-turbo is compliant with the libjpeg version 8 API.

Website for libjpeg-turbo is <http://www.libjpeg-turbo.org/>

Additional info about using libjpeg at <http://www.cim.mcgill.ca/~junaed/libjpeg.php>

Altera VIP Suite (Video Processing Suite)

Saturday, January 18, 2014 11:31 PM

Has FPGA IP to convert from RGB to Avalon-ST Video (a bus standard used by their video processing IP) and back.

While this would work, we are getting RGB from the JPEGs, the converters are so-called "clocked converters". They expect video to be coming in at a regular rate and to do real time, low latency processing on that video (ie. Scaling, mixing, deinterlacing). While we are doing some of the same things, we don't have a guaranteed steady stream of video, nor are we treating this like video. We are looking at individual frames and are extracting figures from them. It is a different sort of problem, we are not simply taking in video, processing it, and outputting the result.

~~While we can keep this library in mind, I think we will wind up writing our own IP to do much of this. At the very least, to use it, we need some IP to provide the RGB data and generate a dummy clock for the converter IP provided by altera.~~

SRATCH that. The VIP reference design pulls images from memory and overlays them on clocked video!!! It uses an IP component called a "Frame Reader". This is sort of what we are doing (except there is no clocked video input). It may be worth looking into this. However, it is still overkill for what we are doing but it may provide the key for getting video into and out of the FPGA.

However, getting the video back into ram is a little more difficult. There are frame buffers but they are meant to take video in and stream it back out. Since we would need to write our own video logic that is compliant to this standard if we use it, this will probably not be a huge problem to make a to memory output. Plus, our to memory output is pieces of the image, not the whole thing

After reviewing custom IP tutorial, we may want to use avalon-st (streaming). This is because our video processing is likely unidirectional and we don't want the trouble of dealing with a memory map between DSP components (yuck). Avalon-ST is how almost all DSP is done with Altera. Can write custom component which takes avalon st data and outputs it to memory via avalon-mm. Could be that on the output of object detection, it only raises the valid flag when there is an object to report. The Avalon-ST to Avalon-MM would take that message and store it in mem.

Should likely use Avalon-ST Video as the way to transmit the image data as it supplies both the standard way to transmit the pixel data as well as control packets and user defined packets

Linux Module Development

Saturday, April 26, 2014 7:04 PM

Rocketboards - in the forum

One person has made a sample module for writing a driver for the GHRD

This was written by

<http://rocketboards.org/foswiki/Projects/MyFirstModule>

ProFromDover

Generating Device Tree

Saturday, April 26, 2014 7:57 PM

The steps to generate the device tree are at

<http://rocketboards.org/foswiki/Documentation/GSRDDeviceTreeGenerator>

As one would expect by this point, driver development for the SoC kit is not vanilla Linux driver development. Unfortunately, because Altera already had the NIOS, there can be a lot of confusion on what the correct way to develop a driver is. In many cases, Altera IP has been developed so that it can work with either the NIOS or the Linux soc. It should be clear that not all IP has Linux drivers. It appears that one must develop a driver or use an existing one. The driver, IP link is accomplished in the _hw.tcl file by setting a "compatible with" flag which corresponds to the same value in a device driver.

http://rocketboards.org/foswiki/Documentation/DeviceTreeGenerator#Adding_Device_Tree_Generator_Support_to_an_IP_Block

Copy the resulting socfpga.dts file onto the Linux build system in the
~/yocto-ghrd/meta-altera/recipes-gsrd/linux/files and replace the current file

NOTE in 13.0sp1, there is no HPS timing file, so don't try to include it as you will not be able to find it

Rerun bitbake on the Linux image

Running these commands caused Linux to fail to load. The message from serial was

```
U-Boot SPL 2012.10 (Jan 09 2014 - 17:06:33)
SDRAM : Initializing MMR registers
SDRAM : Calibrationg PHY
SEQ.C: Preparing to start memory calibration
SEQ.C: CALIBRATION PASSED
DESIGNWARE SD/MMC: 0
```

```
U-Boot 2012.10 (Jan 09 2014 - 17:06:33)

CPU      : Altera SOC FPGA Platform
BOARD    : Altera SOC FPGA Cyclone 5 Board
DRAM     : 1 Gib
MMC      : DESIGNWARE SD/MMC: 0
*** Warning - bad CRC, using default environment

In:      serial
Out:     serial
Err:     serial
Net:     mii0
Warning: failed to set MAC address

Hit any key to stop autoboot:  0
reading uImage

2693952 bytes read
reading socfpga.dtb
```

```
9488 bytes read
## Booting kernel from Legacy Image at 00007fc0 ...
Image Name: Linux-3.7.0
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 2693888 Bytes = 2.6 MiB
Load Address: 00008000
Entry Point: 00008000
## Flattened Device Tree blob at 00000100
Booting using the fdt blob at 0x00000100
XIP Kernel Image ... OK
OK
Loading Device Tree to 0fff9000, end 0fffe50f ... OK
Starting kernel ...
```

Based on info from, <http://www.alteraforum.com/forum/showthread.php?t=41647>
need to remove the sysID from the device tree file

Driver Stuff

Sunday, April 27, 2014 3:41 AM

Drivers can be found in the linux_altera_3.7/drivers

Altera PIO driver can be found in drivers/gpio/altera-gpio
This is based on another driver.

The driver uses the platform_device.h library which can be found under includes/kernel
Documentation is at /Documentation/driver-model

Flashing FPGA

Saturday, May 3, 2014 3:33 PM

Follow instructions on <http://rocketboards.org/foswiki/Documentation/GSRDQuartusProgrammer>

1. Compile Hardware design (instructions at <http://rocketboards.org/foswiki/Documentation/GSRDCompileHardwareDesign> I figured out a lot of this myself as we made many modifications). It should just work.)
 - a. Generate Qsys system.
 - b. Run tcl scripts for memory assignments (if necessary, they should have already been run)
 - c. run compiler
2. Flash SD card but DO NOT insert it into dev board
3. Set SW2 DW 2 (Fact Load) to OFF - all switches to right
4. Start programmer from Quartus II
5. Follow instructions in ug_cv_sco_dev_kit

Writing to memory

Sunday, May 4, 2014 12:42 AM

Memory can be written to directly using mmap and /dev/mem

when opening /dev/mem, the O_SYNC option must be set to ensure write actually gets to memory
(equiv to calling sync after each write according to man page)

HPS to FPGA DDR3 Notes

Friday, May 9, 2014 5:25 PM

FPGA does not have access to entire FPGA DDR3 memory space. The h2f AXI master starts at 0xC000 0000 from the perspective of the ARM. Only 28 bits of the 32 bit address space of the memory can be accessed, yielding only 268.4 MB or 1/4 of the DDR3 memory space. This is consistent with the concept of bridge address spaces as detailed in http://www.altera.com/literature/ug/ug_sopc_builder.pdf. This should be OK for us since the amount of data being stored is relatively low. There are of course ways around this but that would be a step for the next prototype.

New Note:

It looks like the 0xFFFF FFFF is a limitation in Qsys. More investigation is required to figure out why this is the case but I will not focus on this now

QoS Stats RTSP MJPEG

Thursday, May 22, 2014 2:57 AM

```
christopher@seniorDesignUbuntuLTS:~/live555$ openRTSP -t -v -Q -F mjpg-video
rtsp://192.168.11.31:554/live2.sdp > /dev/null
Opening connection to 192.168.11.31, port 554...
...remote connection opened
Sending request: OPTIONS rtsp://192.168.11.31:554/live2.sdp RTSP/1.0
CSeq: 2
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
```

```
Received 152 new bytes of response data.
Received a complete OPTIONS response:
RTSP/1.0 200 OK
CSeq: 2
Date: Thu, May 22 2014 09:56:44 GMT
Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PARAMETER
```

```
Sending request: DESCRIBE rtsp://192.168.11.31:554/live2.sdp RTSP/1.0
CSeq: 3
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Accept: application/sdp
```

```
Received 574 new bytes of response data.
Received a complete DESCRIBE response:
RTSP/1.0 200 OK
CSeq: 3
Date: Thu, May 22 2014 09:56:44 GMT
Content-Base: rtsp://192.168.11.31/live2.sdp/
Content-Type: application/sdp
Content-Length: 410
```

```
v=0
o=- 1400752572597196 1 IN IP4 192.168.11.31
s=RTSP/RTP stream 2 from DCS-2332L-CAM1
i=live2.sdp with v2.0
t=0 0
a=type:broadcast
a=control:*
a=range:npt=0-
a=x-qt-text-nam:RTSP/RTP stream 2 from DCS-2332L-CAM1
a=x-qt-text-inf:live2.sdp
m=video 0 RTP/AVP 26
c=IN IP4 0.0.0.0
b=AS:1500
a=x-dimensions:1280,800
a=control:track1
m=audio 0 RTP/AVP 0
```

```
c=IN IP4 0.0.0.0  
b=AS:64  
a=control:track2
```

Opened URL "rtsp://192.168.11.31:554/live2.sdp", returning a SDP description:

```
v=0  
o=- 1400752572597196 1 IN IP4 192.168.11.31  
s=RTSP/RTP stream 2 from DCS-2332L-CAM1  
i=live2.sdp with v2.0  
t=0 0  
a=type:broadcast  
a=control:  
a=range:npt=0-  
a=x-qt-text-nam:RTSP/RTP stream 2 from DCS-2332L-CAM1  
a=x-qt-text-inf:live2.sdp  
m=video 0 RTP/AVP 26  
c=IN IP4 0.0.0.0  
b=AS:1500  
a=x-dimensions:1280,800  
a=control:track1  
m=audio 0 RTP/AVP 0  
c=IN IP4 0.0.0.0  
b=AS:64  
a=control:track2
```

Created receiver for "video/JPEG" subsession (client ports 34692-34693)

Ignoring "audio/PCMU" subsession, because we've asked to receive a single video session only

Sending request: SETUP rtsp://192.168.11.31/live2.sdp/track1 RTSP/1.0

CSeq: 4

User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)

Transport: RTP/AVP/TCP;unicast;interleaved=0-1

Received 179 new bytes of response data.

Received a complete SETUP response:

RTSP/1.0 200 OK

CSeq: 4

Date: Thu, May 22 2014 09:56:44 GMT

Transport: RTP/AVP/TCP;unicast;destination=192.168.11.20;source=192.168.11.31;interleaved=0-1

Session: 6B8B4567

Setup "video/JPEG" subsession (client ports 34692-34693)

Outputting data from the "video/JPEG" subsession to 'stdout'

Sending request: PLAY rtsp://192.168.11.31/live2.sdp/ RTSP/1.0

CSeq: 5

User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)

Session: 6B8B4567

Range: npt=0.000-

Received a complete PLAY response:

RTSP/1.0 200 OK

```
CSeq: 5
Date: Thu, May 22 2014 09:56:44 GMT
Range: npt=0.000-
Session: 6B8B4567
RTP-Info: url=rtsp://192.168.11.31/live2.sdp/track1;seq=57466;rtptime=0, url=rtsp://192.168.11.31/live2.sdp/track2;seq=0;rtptime=0
```

```
Started playing session
Receiving streamed data (signal with "kill -HUP 31603" or "kill -USR1 31603" to terminate)...
Got shutdown signal
begin_QOS_statistics
subsession video/JPEG
num_packets_received 15988
num_packets_lost 2
elapsed_measurement_time 33.000209
kBytes_received_total 22824.376000
measurement_sampling_interval_ms 1000
kbits_per_second_min 1081.594110
kbits_per_second_ave 5533.147017
kbits_per_second_max 10433.606571
packet_loss_percentage_min 0.000000
packet_loss_percentage_ave 0.012508
packet_loss_percentage_max 0.502513
inter_packet_gap_ms_min 0.009000
inter_packet_gap_ms_ave 2.027994
inter_packet_gap_ms_max 1292.886000
end_QOS_statistics
Sending request: TEARDOWN rtsp://192.168.11.31/live2.sdp/ RTSP/1.0
CSeq: 6
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Session: 6B8B4567
```

QoS Stats RTSP H.264

Thursday, May 22, 2014 4:10 AM

```
ristopher@seniorDesignUbuntuLTS:~/live555$ openRTSP -v -t -Q rtsp://192.168.11.31:554/live1.sdp >
/dev/null
Opening connection to 192.168.11.31, port 554...
...remote connection opened
Sending request: OPTIONS rtsp://192.168.11.31:554/live1.sdp RTSP/1.0
CSeq: 2
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
```

```
Received 152 new bytes of response data.
Received a complete OPTIONS response:
RTSP/1.0 200 OK
CSeq: 2
Date: Thu, May 22 2014 11:06:23 GMT
Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PARAMETER
```

```
Sending request: DESCRIBE rtsp://192.168.11.31:554/live1.sdp RTSP/1.0
CSeq: 3
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Accept: application/sdp
```

```
Received 816 new bytes of response data.
Received a complete DESCRIBE response:
RTSP/1.0 200 OK
CSeq: 3
Date: Thu, May 22 2014 11:06:23 GMT
Content-Base: rtsp://192.168.11.31/live1.sdp/
Content-Type: application/sdp
Content-Length: 652
```

```
v=0
o=- 1400752572595686 1 IN IP4 192.168.11.31
s=RTSP/RTP stream 1 from DCS-2332L-CAM1
i=live1.sdp with v2.0
t=0 0
a=type:broadcast
a=control:*
a=range:npt=0-
a=x-qt-text-nam:RTSP/RTP stream 1 from DCS-2332L-CAM1
a=x-qt-text-inf:live1.sdp
m=video 0 RTP/AVP 96
c=IN IP4 0.0.0.0
b=AS:1500
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1;profile-level-id=640028;sprop-parameter-sets=Z2QAKK2EBUViuKxUdCAqKxXFYqOhAVFYrisVHQgKisVxWKjoQFRWK4rFR0ICorFcVio6ECSFITk8nyfk/k
```

```

/J8nm5s00IEkKQnJ5Pk/J+T5PNzZprQCgDLSpAAAAwHgAAA4QYEABfXhAAAvrwr3vheEQjU,aO48sA==
a=control:track1
m=audio 0 RTP/AVP 0
c=IN IP4 0.0.0.0
b=AS:64
a=control:track2

Opened URL "rtsp://192.168.11.31:554/live1.sdp", returning a SDP description:
v=0
o=- 1400752572595686 1 IN IP4 192.168.11.31
s=RTSP/RTP stream 1 from DCS-2332L-CAM1
i=live1.sdp with v2.0
t=0 0
a=type:broadcast
a=control:*
a=range:npt=0-
a=x-qt-text-nam:RTSP/RTP stream 1 from DCS-2332L-CAM1
a=x-qt-text-inf:live1.sdp
m=video 0 RTP/AVP 96
c=IN IP4 0.0.0.0
b=AS:1500
a=rtpmap:96 H264/90000
a=fmt:96 packetization-mode=1;profile-level-id=640028;sprop-parameter-
sets=Z2QAKK2EBUViuKxUdCAqKxFYqOhAVFYrisVHQgKisVxWKjoQFRWK4rFR0ICorFcVio6ECSFITk8nyfk/k
/J8nm5s00IEkKQnJ5Pk/J+T5PNzZprQCgDLSpAAAAwHgAAA4QYEABfXhAAAvrwr3vheEQjU,aO48sA==
a=control:track1
m=audio 0 RTP/AVP 0
c=IN IP4 0.0.0.0
b=AS:64
a=control:track2

```

```

Created receiver for "video/H264" subsession (client ports 54318-54319)
Ignoring "audio/PCMU" subsession, because we've asked to receive a single video session only
Sending request: SETUP rtsp://192.168.11.31/live1.sdp/track1 RTSP/1.0
CSeq: 4
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Transport: RTP/AVP/TCP;unicast;interleaved=0-1

```

```

Received 179 new bytes of response data.
Received a complete SETUP response:
RTSP/1.0 200 OK
CSeq: 4
Date: Thu, May 22 2014 11:06:23 GMT
Transport: RTP/AVP/TCP;unicast;destination=192.168.11.20;source=192.168.11.31;interleaved=0-1
Session: OF18C5D3

```

```

Setup "video/H264" subsession (client ports 54318-54319)
Outputting data from the "video/H264" subsession to 'stdout'
Sending request: PLAY rtsp://192.168.11.31/live1.sdp/ RTSP/1.0
CSeq: 5
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)

```

```
Session: OF18C5D3
Range: npt=0.000-
```

```
Received a complete PLAY response:
RTSP/1.0 200 OK
CSeq: 5
Date: Thu, May 22 2014 11:06:23 GMT
Range: npt=0.000-
Session: OF18C5D3
RTP-Info: url=rtsp://192.168.11.31/live1.sdp/track1;seq=57896;rtptime=0,
url=rtsp://192.168.11.31/live1.sdp/track2;seq=0;rtptime=0
```

```
Started playing session
Receiving streamed data (signal with "kill -HUP 32050" or "kill -USR1 32050" to terminate)...
Got shutdown signal
begin_QOS_statistics
subsession video/H264
num_packets_received 16216
num_packets_lost 1
elapsed_measurement_time 54.670178
kBytes_received_total 22363.859000
measurement_sampling_interval_ms 1000
kbits_per_second_min 470.622294
kbits_per_second_ave 3272.549652
kbits_per_second_max 34736.226995
packet_loss_percentage_min 0.000000
packet_loss_percentage_ave 0.006166
packet_loss_percentage_max 0.473934
inter_packet_gap_ms_min 0.008000
inter_packet_gap_ms_ave 3.292083
inter_packet_gap_ms_max 1791.568000
end_QOS_statistics
Sending request: TEARDOWN rtsp://192.168.11.31/live1.sdp/ RTSP/1.0
CSeq: 6
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Session: OF18C5D3
```

```
^C
christopher@seniorDesignUbuntuLTS:~/live555$ openRTSP -v -t -Q rtsp://192.168.11.31:554/live1.sdp >
/dev/null
Opening connection to 192.168.11.31, port 554...
...remote connection opened
Sending request: OPTIONS rtsp://192.168.11.31:554/live1.sdp RTSP/1.0
CSeq: 2
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
```

```
Received 152 new bytes of response data.
Received a complete OPTIONS response:
RTSP/1.0 200 OK
```

```
CSeq: 2
Date: Thu, May 22 2014 11:08:12 GMT
Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PARAMETER
```

```
Sending request: DESCRIBE rtsp://192.168.11.31:554/live1.sdp RTSP/1.0
CSeq: 3
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Accept: application/sdp
```

Received 816 new bytes of response data.

```
Received a complete DESCRIBE response:
RTSP/1.0 200 OK
CSeq: 3
Date: Thu, May 22 2014 11:08:12 GMT
Content-Base: rtsp://192.168.11.31/live1.sdp/
Content-Type: application/sdp
Content-Length: 652
```

```
v=0
o=- 1400752572595686 1 IN IP4 192.168.11.31
s=RTSP/RTP stream 1 from DCS-2332L-CAM1
i=live1.sdp with v2.0
t=0 0
a=type:broadcast
a=control:*
a=range:npt=0-
a=x-qt-text-nam:RTSP/RTP stream 1 from DCS-2332L-CAM1
a=x-qt-text-inf:live1.sdp
m=video 0 RTP/AVP 96
c=IN IP4 0.0.0.0
b=AS:1500
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1;profile-level-id=640028;sprop-parameter-
sets=Z2QAKK2EBUViuKxUdCAqKxXFYqOhAVFYrisVHQgKisVxWKjoQFRWK4rFR0ICorFcVio6ECSFITk8nyfk/k
/J8nm5s00!EkKQnJ5Pk/J+T5PNzZprQCgDLSpAAAAwHgAAA4QYEAAABfxhAAAvrwr3vheEQjU,aO48sA==
a=control:track1
m=audio 0 RTP/AVP 0
c=IN IP4 0.0.0.0
b=AS:64
a=control:track2
```

Opened URL "rtsp://192.168.11.31:554/live1.sdp", returning a SDP description:

```
v=0
o=- 1400752572595686 1 IN IP4 192.168.11.31
s=RTSP/RTP stream 1 from DCS-2332L-CAM1
i=live1.sdp with v2.0
t=0 0
a=type:broadcast
a=control:*
a=range:npt=0-
a=x-qt-text-nam:RTSP/RTP stream 1 from DCS-2332L-CAM1
```

```
a=x-qt-text-inf:live1.sdp
m=video 0 RTP/AVP 96
c=IN IP4 0.0.0.0
b=AS:1500
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1;profile-level-id=640028;sprop-parameter-
sets=Z2QAKK2EBUViuKxUdCAqKxFYqOhAVFYrisVHQgKisVxWKjoQFRWK4rFR0ICorFcVio6ECSFITk8nyfk/k
/J8nm5s00!EkKQnj5Pk/J+T5PNzZprQCgDLSpAAAAwHgAAA4QYEABfXhAAAvrwr3vheEQjU,aO48sA==
a=control:track1
m=audio 0 RTP/AVP 0
c=IN IP4 0.0.0.0
b=AS:64
a=control:track2
```

```
Created receiver for "video/H264" subsession (client ports 54608-54609)
Ignoring "audio/PCMU" subsession, because we've asked to receive a single video session only
Sending request: SETUP rtsp://192.168.11.31/live1.sdp/track1 RTSP/1.0
CSeq: 4
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Transport: RTP/AVP/TCP;unicast;interleaved=0-1
```

```
Received 179 new bytes of response data.
Received a complete SETUP response:
RTSP/1.0 200 OK
CSeq: 4
Date: Thu, May 22 2014 11:08:12 GMT
Transport: RTP/AVP/TCP;unicast;destination=192.168.11.20;source=192.168.11.31;interleaved=0-1
Session: 5524F612
```

```
Setup "video/H264" subsession (client ports 54608-54609)
Outputting data from the "video/H264" subsession to 'stdout'
Sending request: PLAY rtsp://192.168.11.31/live1.sdp/ RTSP/1.0
CSeq: 5
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Session: 5524F612
Range: npt=0.000-
```

```
Received a complete PLAY response:
RTSP/1.0 200 OK
CSeq: 5
Date: Thu, May 22 2014 11:08:12 GMT
Range: npt=0.000-
Session: 5524F612
RTP-Info: url=rtsp://192.168.11.31/live1.sdp/track1;seq=24603;rtptime=
0,url=rtsp://192.168.11.31/live1.sdp/track2;seq=0;rtptime=0
```

```
Started playing session
Receiving streamed data (signal with "kill -HUP 32059" or "kill -USR1 32059" to terminate)...
Got shutdown signal
```

```
begin_QOS_statistics
subsession video/H264
num_packets_received 9355
num_packets_lost 6
elapsed_measurement_time 32.000355
kBytes_received_total 12889.925000
measurement_sampling_interval_ms 1000
kbits_per_second_min 0.000000
kbits_per_second_ave 3222.445501
kbits_per_second_max 12897.314977
packet_loss_percentage_min 0.000000
packet_loss_percentage_ave 0.064096
packet_loss_percentage_max 0.875274
inter_packet_gap_ms_min 0.009000
inter_packet_gap_ms_ave 3.362048
inter_packet_gap_ms_max 1689.810000
end_QOS_statistics
Sending request: TEARDOWN rtsp://192.168.11.31/live1.sdp/ RTSP/1.0
CSeq: 6
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Session: 5524F612
```