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.
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
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
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-iptest_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-grsd-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
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 originaly 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 change 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-
allowed 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, identity 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 Mbit/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 \( \frac{1024 \times 100 \text{ kBit/sec Ethernet}}{5533 \text{ kBit/sec/stream}} \approx 18.5 \) 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 streams 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 may 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.
<table>
<thead>
<tr>
<th>Description</th>
<th>Mfr.</th>
<th>Mfr. Part Number</th>
<th>Qty.</th>
<th>Cost / Unit</th>
<th>Cost</th>
<th>Amount Requested from SCU</th>
<th>Donation Expected</th>
<th>Provided by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quartus II Subscription Edition</td>
<td>Altera</td>
<td>SW-QUARTUS-SE-FIX</td>
<td>3</td>
<td>$2,995.00</td>
<td>$8,985.00</td>
<td>0</td>
<td>$8,985.00</td>
<td>Altera</td>
</tr>
<tr>
<td>Cyclone V SoC Development Kit and SoC Embedded Design Suite</td>
<td>Altera</td>
<td>DK-DEV-5CSXC6NES</td>
<td>2</td>
<td>$1,595.00</td>
<td>$3,190.00</td>
<td>0</td>
<td>$3,190.00</td>
<td>Altera</td>
</tr>
<tr>
<td>ONVIF Compliant Fixed IP Camera</td>
<td>D-Link</td>
<td>DCS-2310L</td>
<td>2</td>
<td>$297.99</td>
<td>$595.98</td>
<td>$595.98</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ONVIF Compliant 720P PTZ IP Camera</td>
<td>Panasonic</td>
<td>WV-ST165</td>
<td>1</td>
<td>$488.99</td>
<td>$488.99</td>
<td>$488.99</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Seagate Backup Plus 2 TB USB 3.0 Desktop External Hard Drive</td>
<td>Seagate</td>
<td>STCA2000100</td>
<td>1</td>
<td>$89.99</td>
<td>$89.99</td>
<td>$89.99</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Seagate Barracuda 1 TB HDD SATA Internal Bare Drive</td>
<td>Seagate</td>
<td>ST1000DM003</td>
<td>3</td>
<td>$67.19</td>
<td>$201.57</td>
<td>$201.57</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Misc. Cables / Connectors</td>
<td></td>
<td></td>
<td>1</td>
<td>$80.00</td>
<td>$80.00</td>
<td>$80.00</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Server 2012 Standard Edition</td>
<td>Microsoft</td>
<td></td>
<td>3</td>
<td>$882.00</td>
<td>$2,646.00</td>
<td>0</td>
<td>$2,646.00</td>
<td>Dreamspark Academic (Microsoft)</td>
</tr>
<tr>
<td>Microsoft SQL Server 2012 Standard</td>
<td>Microsoft</td>
<td>1 Core License (Min 4 Per Server)</td>
<td>4</td>
<td>$1,793.00</td>
<td>$7,172.00</td>
<td>0</td>
<td>$7,172.00</td>
<td>Dreamspark Academic (Microsoft)</td>
</tr>
<tr>
<td>Microsoft Visual Studio Premium (w/ MSDN)</td>
<td>Microsoft</td>
<td></td>
<td>4</td>
<td>$6,119.00</td>
<td>$24,476.00</td>
<td>0</td>
<td>$24,476.00</td>
<td>Dreamspark Academic (Microsoft)</td>
</tr>
<tr>
<td>Team Foundation Service (TFS)</td>
<td>Microsoft</td>
<td>1 License</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Microsoft (Free Service)</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$47,925.53</td>
<td>$1,456.53</td>
<td>$46,469</td>
</tr>
</tbody>
</table>

Key:

System Materials

Development Tools

Table 4.1: Development BOM - For Development Setup + Development Tools
<table>
<thead>
<tr>
<th>Description</th>
<th>Manufacturer</th>
<th>Manufacturer Part Number</th>
<th>Vendor</th>
<th>Vendor Part Number</th>
<th>Qty</th>
<th>Cost/Unit</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclone V SoC FPGA</td>
<td>ALTERA</td>
<td>5CSXFC6D6F31C7N</td>
<td>DIGIKEY</td>
<td>5CSXFC6D6F31C7N-ND</td>
<td>1</td>
<td>$283.61</td>
<td>$283.61</td>
</tr>
<tr>
<td>MAX V CPLD System</td>
<td>ALTERA</td>
<td>5M2210ZF256I5N</td>
<td>DIGIKEY</td>
<td>544-2973-ND</td>
<td>1</td>
<td>$25.90</td>
<td>$25.90</td>
</tr>
<tr>
<td>LVDS clock</td>
<td>SILICON LABS</td>
<td>570FAB00973DG</td>
<td>SILICON LABS</td>
<td>Custom Part</td>
<td>1</td>
<td>$40.00</td>
<td>$40.00</td>
</tr>
<tr>
<td>50 MHz crystal</td>
<td>SILICON LABS</td>
<td>510GBA50M0000BAG</td>
<td>SILICON LABS</td>
<td>Custom Part</td>
<td>1</td>
<td>$5.00</td>
<td>$5.00</td>
</tr>
<tr>
<td>MagJack 1000BaseT</td>
<td>BEL FUSE</td>
<td>L829-1J1T-43</td>
<td>DIGIKEY</td>
<td>380-1110-ND</td>
<td>1</td>
<td>$5.75</td>
<td>$5.75</td>
</tr>
<tr>
<td>Real Time Clock</td>
<td>MAXIM</td>
<td>DS1339C</td>
<td>DIGIKEY</td>
<td>DS1339C-33#-ND</td>
<td>1</td>
<td>$6.78</td>
<td>$6.78</td>
</tr>
<tr>
<td>32M 16 8, 1024-MB DDR3 SDRAM</td>
<td>Micron</td>
<td>MT41K256M16HA-125:E-ND</td>
<td>DIGIKEY</td>
<td>MT41K256M16HA-125:E-ND</td>
<td>1</td>
<td>$7.18</td>
<td>$7.18</td>
</tr>
<tr>
<td>256-Mb NOR flash</td>
<td>ALTERA</td>
<td>EPCQ256SH16N</td>
<td>DIGIKEY</td>
<td>EPCQ256SH16N-ND</td>
<td>1</td>
<td>$50.00</td>
<td>$50.00</td>
</tr>
<tr>
<td>32-Kb EEPROM</td>
<td>MICROCHIP</td>
<td>24LC32A</td>
<td>DIGIKEY</td>
<td>24LC32A-I/SN-ND</td>
<td>1</td>
<td>$0.41</td>
<td>$0.41</td>
</tr>
<tr>
<td>Micro SD Card slot</td>
<td>WURTH</td>
<td>693 071 010 811</td>
<td>DIGIKEY</td>
<td>732-3819-2-ND</td>
<td>1</td>
<td>$2.10</td>
<td>$2.10</td>
</tr>
<tr>
<td>Power Supply Controller</td>
<td>LINEAR</td>
<td>LTC2978</td>
<td>DIGIKEY</td>
<td>LTC2978AIUP#PBF-ND</td>
<td>1</td>
<td>$21.51</td>
<td>$21.51</td>
</tr>
<tr>
<td>Misc Components</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>$100.00</td>
<td>$100.00</td>
</tr>
<tr>
<td>PCB (6” x 6” Standard 2 Layer) Low Quantity</td>
<td>ADVANCE CIRCUITS</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>$18.33</td>
<td>$18.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$566.57</td>
</tr>
</tbody>
</table>

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

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

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 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](image)

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)
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](image-url)
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](image)

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.

<table>
<thead>
<tr>
<th>MJPEG</th>
<th>H.264</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each frame is a complete frame</td>
<td>On I-frames are complete. The rest are diffs from previous frames</td>
</tr>
<tr>
<td>Higher bandwidth required</td>
<td>Less bandwidth required</td>
</tr>
<tr>
<td></td>
<td>Lossy Compression</td>
</tr>
</tbody>
</table>

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 know 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 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.

![Marching Square Look Up Table](image)

Figure 5.6: Marching Square Look Up Table
Figure 5.7: Original Image
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 identity 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 farther 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.

![Greyscale Image](./bmp/stillSampleMultiple_0000000320.bmp) ![Contours](./bmp/stillSampleMultiple_0000000320.bmp) ![Detected Objects](./bmp/stillSampleMultiple_0000000320.bmp)

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)
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 processed 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.

![Image After Canny Edge Detection](image)

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.
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 and 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 because it 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, a 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.

![Sample of a Directed Weighted Graph](image)

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
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](image)

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 Microsoft's 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 Mircosoft’s 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.

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

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 of 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-tiered 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 stakeholders 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 kgCO$_2$eq/die with 43.2 kgCO$_2$eq/die of that from use. This leaves 3.8 kgCO$_2$eq/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 kgCO$_2$/kg and 2.71 kgCO$_{2e}$/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 possible 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 functionally 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 be 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 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 productions 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


79


80


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].

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

Table 16.1: FPGA (5CSXFC6D6F31C8NES) Overview [6]
<table>
<thead>
<tr>
<th>Resource</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Element (LE)</td>
<td>110K</td>
</tr>
<tr>
<td>ALM</td>
<td>41,509</td>
</tr>
<tr>
<td>Register</td>
<td>166,036</td>
</tr>
<tr>
<td>Memory (M10K)</td>
<td>5,570 Kb</td>
</tr>
<tr>
<td>Memory (MLAB)</td>
<td>621 Kb</td>
</tr>
<tr>
<td>Variable-precision DSP Block</td>
<td>112</td>
</tr>
<tr>
<td>18 x 18 Multiplier</td>
<td>224</td>
</tr>
<tr>
<td>FPGA PLL</td>
<td>6</td>
</tr>
<tr>
<td>HPS PLL</td>
<td>3</td>
</tr>
<tr>
<td>3 Gbps Transceiver</td>
<td>9</td>
</tr>
<tr>
<td>FPGA GPIO</td>
<td>228</td>
</tr>
<tr>
<td>HPS I/O</td>
<td>181</td>
</tr>
<tr>
<td>LVDS (Transmitter)</td>
<td>72</td>
</tr>
<tr>
<td>LVDS (Receiver)</td>
<td>72</td>
</tr>
<tr>
<td>PCIe Hard IP Block</td>
<td>2</td>
</tr>
<tr>
<td>FPGA Hard Memory Controller</td>
<td>1</td>
</tr>
<tr>
<td>HPS Hard Memory Controller</td>
<td>1</td>
</tr>
<tr>
<td>ARM Cortex-A9 MPCore Processor</td>
<td>Dual-core</td>
</tr>
</tbody>
</table>

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

### 16.2 Object Detection - Matlab Simulation Code

```matlab
%% Christopher Yarp
%% Contour Test
%% Senior Design

close all;
clear; clc;

%x = imread('eight.tif');
thresh = 80;
minPts = 10;
minLength = 2;
slackSquareBounding = 0.70;
%slackCircle = ;
slackCenter = 6;
slackRatio = 1;
ratioCenterInner = 3;
ratioCenterOuter = 4.25;
boundingBoxMargin = 5;

imgNameFormat = '.bmp/stillSampleMultiple_%010d.bmp';
startImgName = 2;
endImgName = 850;
%startImgName = 319;
%endImgName = 320;
imgNameStep = 1;
fps = 30/imgNameStep;

videoFileName = 'tracking.avi';
M(floor((endImgName-startImgName)/imgNameStep)) = struct('cdata',[],'colormap',[]);
vidFrameNum = 1;
```
videoWriter = VideoWriter(videoFileName);
open(videoWriter);

for frameNum = startImgName : imgNameStep : endImgName
    clear_all but ('thresh', 'minPts', 'minLength', 'slackSquareBounding', ...
        'slackCenter', 'slackRatio', 'ratioCenterInner', 'ratioCenterOuter', ...
        'boundingBoxMargin', 'imgNameFormat', 'startImgName', 'endImgName', ...
        'imgNameStep', 'fps', 'M', 'frameNum', 'vidFrameNum', 'videoFileName', ...
        'videoWriter');

    fileName = sprintf(imgNameFormat , frameNum);
    %img = importdata('stillSample_0000000100.bmp');
    %img = importdata('stillSample_0000000300.bmp');
    %img = importdata('stillSample_0000000600.bmp');
    img = importdata(fileName);
    display(['frame: ' fileName]);

    x = rgb2gray(img);
    fig = figure(1);
    %figure size from http://www.mathworks.com/matlabcentral/newsreader/view_thread/262957
    set(fig, 'Position', [0, 100, 1280, 800]);
    subplot(2,2,1);
    imshow(x);
    title('Greyscale Image');
    subplot(2,2,2);
    imshow(img);
    title(['Origional Img: ' fileName , 'Interpreter', 'none']);
    subplot(2,2,4)
    h1 = hist(double(x), 256);
    plot(h1)
    h1 = imhist(x,256);
    plot(h1)
    close(h1)
    subplot(2,3);
    %imcontour(x,[225, 225])
    c1 = imcontour(x,[thresh, thresh]);
    title('Contours');

    ix = 1;
    ic = 1;
    npts = 0;
    maxX = 0;
    minX = 0;
    maxY = 0;
    minY = 0;
    numCurvesRec = 1;
    startInd = 1;
    endInd = 1;

    [c1Rows, c1Cols] = size(c1);

    for i = 1:c1Cols
        if (i == ix)
            sizeX = maxX-minX;
            sizeY = maxY-minY;
            if (sizeX > sizeY)
                maxSize = sizeX;
                minSize = sizeY;
            else
                maxSize = sizeY;
                minSize = sizeX;
            end
        end

84
if (maxSide > 0 && minSide > 0)
    boundingBoxRatio = minSide / maxSide;
else
    boundingBoxRatio = 0;
end

% display ([’min X ’ num2str(minX) ’, max X ’ num2str(maxX) ’, min Y ’ num2str(minY) ’, max Y ’ num2str(maxY) ’]);
% display ([’size X ’ num2str(sizeX) ’, size Y ’ num2str(sizeY) ’]);
% display ([’boundingBoxRatio ’ num2str(boundingBoxRatio) ’]);

% filter out curves that have too few points or have nonsquare bounding boxes
if (i˜ = 1 & & npts >= minPts & & sizeX > minLength & & sizeY > minLength & & boundingBoxRatio >= slackSquareBounding)
    % record data of prev curve (including maxx, maxy)
    s = struct (’npts’, npts, ’minX’, minX, ’maxX’, maxX, ’minY’, minY, ’maxY’, maxY, ’startInd’, startInd, ’endInd’, endInd);
    curves(numCurvesRec) = s;
    % display ([’contour ’ num2str(ic) ’, points = ’ num2str(npts) ’, ([’
        num2str(minX) ’, ’ num2str(maxX) ’], [’ num2str(minY) ’, ’ num2str(maxY) ’]));
    numCurvesRec = numCurvesRec + 1;
end

minX = c1(1, i + 1);
maxX = c1(1, i + 1);

minY = c1(2, i + 1);
maxY = c1(2, i + 1);

npts = c1(2, i);
startInd = ix + 1;
ix = ix + npts + 1;
endInd = ix - 1;
ix = ic + 1;
else
    if (c1(1, i) < minX)
        minX = c1(1, i);
    end
    if (c1(1, i) > maxX)
        maxX = c1(1, i);
    end
    if (c1(2, i) < minY)
        minY = c1(2, i);
    end
    if (c1(2, i) > maxY)
        maxY = c1(2, i);
    end
end

% record last curve
sizeX = maxX - minX;
sizeY = maxY - minY;

if (sizeX > sizeY)
    maxSide = sizeX;
    minSide = sizeY;
else
    maxSide = sizeY;
    minSide = sizeX;
end

if (maxSide > 0 && minSide > 0)
    boundingBoxRatio = minSide / maxSide;
else
    boundingBoxRatio = 0;
end
if (isempty(c1) && npts >= minPts && sizeX > minLength && sizeY > minLength && boundingBoxRatio => slackSquareBounding)

% record data of prev curve (including maxx, maxy)

s = struct('npts', npts, 'minX', minX, 'maxX', maxX, 'minY', minY, 'maxY', maxY, 'startInd', startInd, 'endInd', endInd);
curves(numCurvesRec) = s;

display(['contour ', num2str(ic), ', points = ', num2str(npts), ', ([', num2str(minX), ', ', num2str(maxX), '], [', num2str(minY), ', ', num2str(maxY), '])'])

end

display(['num contours: ', num2str(ic)])
display(['num potential contours: ', num2str(numCurvesRec)])

% check center points (this is n^2 -> not good !!!!)

used = zeros(length(curves));

foundCount = 0;

for i = 1:length(curves)
    if (used(i) == 0)
        centerX = (curves(i).maxX + curves(i).minX)/2;
        centerY = (curves(i).maxY + curves(i).minY)/2;
        %display(['(' num2str(centerX) ', ' num2str(centerY) ')'])
        numConcentric = 1;
        concentric(1) = curves(i);
        centerCircleDim = ((curves(i).maxX - curves(i).minX) + (curves(i).maxY - curves(i).minY))/2;
        for j = i+1:length(curves)
            centerXnew = (curves(j).maxX + curves(j).minX)/2;
            centerYnew = (curves(j).maxY + curves(j).minY)/2;
            %check if curve already put in bin and if centers are same
            if (used(j) == 0 && abs(centerX - centerXnew) < slackCenter && abs(centerY - centerYnew) < slackCenter)
                numConcentric = numConcentric + 1;
                concentric(numConcentric) = curves(j);
                if centerCircleDim > ((curves(j).maxX - curves(j).minX) + (curves(j).maxY - curves(j).minY))/2;
                    centerCircleDim = ((curves(j).maxX - curves(j).minX) + (curves(j).maxY - curves(j).minY))/2;
                    center = curves(j);
                end
                used(j) = 1;
            end
        end
    end
    %if more than 3 possibilities, we may have the target, we now need to
    %check ratios of bounding boxes
    if (numConcentric >= 3)
        display(['possibly found one at (' num2str(centerX) ', ' num2str(centerY) ')']);
        %candidateGroups = candidateGroups +1;
        %candidateGroup(candidateGroups) = concentric;
        %now look at ratios -> may be overkill for our application
        foundInner = 0;
        foundOuter = 0;
        for k = 1:length(concentric)
            newCircleDim = ((concentric(k).maxX - concentric(k).minX) + (concentric(k).maxY - concentric(k).minY))/2;
            if abs(newCircleDim / centerCircleDim - ratioCenterInner) <= slackRatio
                foundInner = 1;
                inner = concentric(k);
            else
                foundOuter = 1;
                outer = concentric(k);
            end
        end
    end
end
elseif abs(newCircleDim / centerCircleDim - ratioCenterOuter) <= slackRatio
    foundOuter = 1;
    outer = concentric(k);
end

if (foundInner ~= 0 && foundOuter ~= 0)
    %found it
    foundCount = foundCount + 1;
    foundStr = struct('center', center, 'inner', inner, 'outer', outer);
    found(foundCount) = foundStr;
    display(['found one at (' num2str(centerX) ', ' num2str(centerY) ')']);
end

end

%now plot rings and where we found them
subplot(2,2,4)
imshow(img)
title('Detected Objects');
hold on;
for l = 1:length(found)
    centerBegin = found(l).center.startInd;
    centerEnd = found(l).center.endInd;
    plot(c1(1, centerBegin:centerEnd), c1(2, centerBegin:centerEnd), 'y', 'LineWidth', 2);

    innerBegin = found(l).inner.startInd;
    innerEnd = found(l).inner.endInd;
    plot(c1(1, innerBegin:innerEnd), c1(2, innerBegin:innerEnd), 'g', 'LineWidth', 2);

    outerBegin = found(l).outer.startInd;
    outerEnd = found(l).outer.endInd;
    plot(c1(1, outerBegin:outerEnd), c1(2, outerBegin:outerEnd), 'b', 'LineWidth', 2);

    %plot bounding box
    plot([(found(l).outer.minX - boundingBoxMargin), (found(l).outer.minX - boundingBoxMargin), (found(l).outer.maxX + boundingBoxMargin), (found(l).outer.maxX + boundingBoxMargin), (found(l).outer.minY - boundingBoxMargin), (found(l).outer.minY - boundingBoxMargin)], 'r', 'LineWidth', 4);
end
hold off;

M(vidFrameNum) = getframe(fig);
writeVideo(videoWriter, M(vidFrameNum));

end

% Video writer: http://www.mathworks.com/help/matlab/ref/videowriter-class.html
close(videoWriter);

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)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<system name="$$[FILENAME]">
  <component name="$$[FILENAME]">
    <displayName>$[FILENAME]</displayName>
    <version>1.0</version>
    <description>""</description>
    <tags>""</tags>
    <categories>"System" /></component>
  <parameter name="bonusData"><![CDATA[bonusData]]></parameter>
  <element name="$$[FILENAME]">}</element>
  <element IPTest_0>
    <datum _sortIndex>
      <value>"16";</value>
      <type>"int";</type>
    </datum>
  </element>
  <element ImgDiff_0>
    <datum _sortIndex>
      <value>"15";</value>
      <type>"int";</type>
    </datum>
  </element>
  <element jtag_uart . avalon . jtag_slave>
    <datum lockedAddress>
      <value>"1";</value>
      <type>"boolean";</type>
    </datum>
    <datum baseAddress>
      <value>"131072";</value>
      <type>"String";</type>
    </datum>
  </element>
  <element intr_capturer_0 . avalon_slave_0>
    <datum lockedAddress>
      <value>"1";</value>
      <type>"boolean";</type>
    </datum>
    <datum baseAddress>
      <value>"196608";</value>
      <type>"String";</type>
    </datum>
  </element>
  <element mem_if_ddr3_fpga . avl_0>
    <datum baseAddress>
      <value>"0";</value>
      <type>"String";</type>
    </datum>
  </element>
  <element button_pio>
    <element $[FILENAME]>
    <element IPTest_0>
      <element ImgDiff_0>
        <element jtag_uart . avalon . jtag_slave>
          <element intr_capturer_0 . avalon_slave_0>
            <element mem_if_ddr3_fpga . avl_0>
              <element button_pio>
```
datum _sortIndex
{
  value = "5";
  type = "int";
}
datum sopeditor_expanded
{
  value = "1";
  type = "boolean";
}
element sysid_qsys.control_slave
{
  datum lockedAddress
  {
    value = "1";
    type = "boolean";
  }
  datum baseAddress
  {
    value = "65536";
    type = "String";
  }
}
element IPTest_0.csr
{
  datum baseAddress
  {
    value = "2048";
    type = "String";
  }
}
element ImgDiff_0.csr
{
  datum baseAddress
  {
    value = "0";
    type = "String";
  }
}
element dipsw_pio
{
  datum _sortIndex
  {
    value = "4";
    type = "int";
  }
}
element ext_clk_50
{
  datum _sortIndex
  {
    value = "9";
    type = "int";
  }
}
element hps_0.f2h_axi_slave
{
  datum lockedAddress
  {
    value = "0";
    type = "boolean";
  }
  datum baseAddress
  {
    value = "0";
    type = "String";
  }
}
element fpga_only_master
{
    datum _sortIndex
    {
        value = "7";
        type = "int";
    }
}

element hps_0
{
    datum _sortIndex
    {
        value = "0";
        type = "int";
    }
}

element hps_only_master
{
    datum _sortIndex
    {
        value = "1";
        type = "int";
    }
}

element intr_capturer_0
{
    datum _sortIndex
    {
        value = "8";
        type = "int";
    }
}

element jtag_uart_irq
{
    datum _tags
    {
        value = "";
        type = "String";
    }
}

element jtag_uart
{
    datum _sortIndex
    {
        value = "6";
        type = "int";
    }
}

element led_pio
{
    datum _sortIndex
    {
        value = "3";
        type = "int";
    }
}

element mem_if_ddr3_fpga
{
    datum _sortIndex
    {
        value = "13";
        type = "int";
    }
}

element mm_clock_crossing_bridge_0
{
    datum _sortIndex
}
```plaintext
datum .sortIndex
    value = "14";
type = "int";
}

element pll_0

datum .sortIndex
    value = "11";
type = "int";
}

element refclk_100

datum .sortIndex
    value = "10";
type = "int";
}

element jtag_uart.reset

datum .tags
    value = "";
type = "String";
}

element mm_clock.crossing.bridge_1.s0

datum baseAddress
    value = "196608";
type = "String";
}

element mm_clock.crossing.bridge_0.s0

datum baseAddress
    value = "0";
type = "String";
}

element dipsw.pio.s1

datum .lockedAddress
    value = "1";
type = "boolean";
}

datum baseAddress
    value = "65664";
type = "String";
}

element led.pio.s1
```
datum _lockedAddress
{
    value = "1";
    type = "boolean";
}
datum baseAddress
{
    value = "65600";
    type = "String";
}
}

element button_pio.s1
{
    datum _lockedAddress
    {
        value = "1";
        type = "boolean";
    }
datum baseAddress
    {
        value = "65728";
        type = "String";
    }
}

element sysid_qsys
{
    datum _sortIndex
    {
        value = "2";
        type = "int";
    }
}
]]>
</parameter>

<parameter name="clockCrossingAdapter" value="FIFO" />
<parameter name="device" value="5CSXFC6D6F31C8ES" />
<parameter name="deviceFamily" value="Cyclone V" />
<parameter name="deviceSpeedGrade" value="8_H6" />
<parameter name="fabricMode" value="QSYS" />
<parameter name="generateLegacySim" value="false" />
<parameter name="generationId" value="0" />
<parameter name="globalResetBus" value="false" />
<parameter name="hdlLanguage" value="VERILOG" />
<parameter name="maxAdditionalLatency" value="4" />
<parameter name="projectName" value="orion_system.qpf" />
<parameter name="sopcBorderPoints" value="false" />
<parameter name="systemHash" value="1" />
<parameter name="timeStamp" value="1399677421041" />
<parameter name="useTestBenchNamingPattern" value="false" />
</instanceScript>
</instanceScript>

@interface name="memory" internal="hps_0.memory" type="conduit" dir="end" />
@interface name="hps_0.hps_io" internal="hps_0.hps_io" type="conduit" dir="end" />
@interface name="clk_50" internal="ext_clk_50.clk_in" type="clock" dir="end" />
@interface name="led_pio_external_connection" internal="led_pio_external_connection" type="conduit" dir="end" />
@interface name="dipsw_pio_external_connection" internal="dipsw_pio_external_connection" type="conduit" dir="end" />
@interface name="button_pio_external_connection" internal="button_pio_external_connection" type="conduit" dir="end" />
<interface>
  name="hps_0.h2f_reset"
  internal="hps_0.h2f_reset"
  type="reset"
  dir="start" />
</interface>

<interface>
  name="fpga_memory"
  internal="mem_if_ddr3.fpga_memory"
  type="conduit"
  dir="end" />
</interface>

<interface>
  name="fpga_memory_oct"
  internal="mem_if_ddr3.fpga_memory_oct"
  type="conduit"
  dir="end" />
</interface>

<interface name="refclk_100" internal="refclk_100.clk_in" type="clock" dir="end" />

<interface>
  name="reset_50"
  internal="ext_clk_50.clk_in_reset"
  type="reset"
  dir="end" />
</interface>

<interface>
  name="mem_if_ddr3.fpga_status"
  internal="mem_if_ddr3.fpga_status"
  type="conduit"
  dir="end" />
</interface>

<interface>
  name="mem_if_ddr3.fpga_pll_sharing"
  internal="mem_if_ddr3.fpga_pll_sharing"
  type="conduit"
  dir="end" />
</interface>

<module kind="altera_avalon_sysid_qsys"
version="13.0"
enabled="1"
name="sysid_qsys">
  <parameter name="id" value="-1395322102" />
  <parameter name="timestamp" value="0" />
  <parameter name="AUTO_CLK_CLOCK_RATE" value="50000000" />
  <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>

<module kind="altera_hps" version="13.0.1" enabled="1" name="hps_0">
  <parameter name="MEM_VENDOR" value="JEDEC" />
  <parameter name="MEM_FORMAT" value="DISCRETE" />
  <parameter name="RDIMM_CONFIG" value="0x0000000000000000" />
  <parameter name="LRDIMM_EXTENDED_CONFIG">0x0000000000000000</parameter>
  <parameter name="DISCRETE_FLY_BY" value="true" />
  <parameter name="DEVICE_DEPTH" value="1" />
  <parameter name="MEM_MIRROR_ADDRESSING" value="0" />
  <parameter name="MEM_CLK_FREQ_MAX" value="800.0" />
  <parameter name="MEM_ROW_ADDR_WIDTH" value="15" />
  <parameter name="MEM_COL_ADDR_WIDTH" value="10" />
  <parameter name="MEM_DQ_WIDTH" value="40" />
  <parameter name="MEM_DQ_PER_DQS" value="8" />
  <parameter name="MEM_BANK_ADDR_WIDTH" value="3" />
  <parameter name="MEM_IF_DM_PINS_EN" value="true" />
  <parameter name="MEM_IF_DQS_EN" value="true" />
  <parameter name="MEM_NUMBER_OF_DIMMS" value="1" />
  <parameter name="MEM_NUMBER_OF_RANKS_PER_DIMM" value="1" />
  <parameter name="MEM_NUMBER_OF_RANKS_PER_DEVICE" value="1" />
  <parameter name="MEM_RANK_Multiplication_FACTOR" value="1" />
  <parameter name="MEM_CLK_WIDTH" value="1" />
  <parameter name="MEM_CS_WIDTH" value="1" />
  <parameter name="MEM_CLK_EN_WIDTH" value="1" />
  <parameter name="ALTTEMEMPHY_COMPATIBLE_MODE" value="false" />
  <parameter name="NEXTGEN" value="true" />
  <parameter name="MEM_IF_BOARD_BASE_DELAY" value="10" />
  <parameter name="MEM_IF_SIM_VALID_WINDOW" value="0" />
</module>
<parameter name="MEM_GUARANTEED_WRITE_INIT" value="false" />
<parameter name="MEM_VERBOSE" value="true" />
<parameter name="PINGPONGPHY_EN" value="false" />
<parameter name="REFRESH_BURST_VALIDATION" value="false" />
<parameter name="MEM_CLK" value="OFF" />
<parameter name="MEM_BF" value="Sequential" />
<parameter name="MEM_ASX" value="Manual" />
<parameter name="MEM_PD" value="DLL off" />
<parameter name="MEM_DRY_STR" value="RZQ/6" />
<parameter name="MEM_DLL_EN" value="true" />
<parameter name="MEM_RTT_NOM" value="RZQ/6" />
<parameter name="MEM_RTT_WR" value="Dynamic ODT off" />
<parameter name="MEM_WTC" value="true" />
<parameter name="MEM_ATCL" value="Disabled" />
<parameter name="MEM_TCL" value="7" />
<parameter name="MEM_AUTO_LEVELING_MODE" value="true" />
<parameter name="MEM_USER_LEVELING_MODE" value="Leveling" />
<parameter name="MEM_INIT_EN" value="false" />
<parameter name="MEM_INIT_FILE" value="" />
<parameter name="DATADATA WIDTH" value="224" />
<parameter name="TIMING_TIS" value="170" />
<parameter name="TIMING_TIH" value="120" />
<parameter name="TIMING_TDS" value="10" />
<parameter name="TIMING_TDH" value="45" />
<parameter name="TIMING_TDQSO" value="100" />
<parameter name="TIMING_TQHS" value="300" />
<parameter name="TIMING_TQH" value="0.38" />
<parameter name="TIMING_TDQSCK" value="225" />
<parameter name="TIMING_TDQSCKDS" value="450" />
<parameter name="TIMING_TDQSCSDKLM" value="900" />
<parameter name="TIMING_TDQSCKDL" value="1200" />
<parameter name="TIMING_TDQSS" value="0.27" />
<parameter name="TIMING_TDQSH" value="0.35" />
<parameter name="TIMING_TQH" value="0.4" />
<parameter name="TIMING_TDH" value="0.18" />
<parameter name="TIMING_TDSS" value="0.18" />
<parameter name="MEM_PINUS" value="500" />
<parameter name="MEM_TMRD_CLK" value="4" />
<parameter name="MEM_TRAS_NS" value="35.0" />
<parameter name="MEM_TRCD_NS" value="13.75" />
<parameter name="MEM_TRP_NS" value="13.75" />
<parameter name="MEM_TREFL_US" value="7.8" />
<parameter name="MEM_TRFC_NS" value="260.0" />
<parameter name="CFG_TCDD_NS" value="3.5" />
<parameter name="MEM_TWR_NS" value="15.0" />
<parameter name="MEM_TWTR" value="4" />
<parameter name="MEM_TFAW_NS" value="30.0" />
<parameter name="MEM_TRR_NS" value="10.0" />
<parameter name="MEM_TRTP_NS" value="10.0" />
<parameter name="POWER_OF_TWO_BUS" value="false" />
<parameter name="SOPC_COMPAT_RESET" value="false" />
<parameter name="AVL_MAX_SIZE" value="4" />
<parameter name="BYTE_ENABLE" value="true" />
<parameter name="ENABLE_CTRL_AVALON_INTERFACE" value="true" />
<parameter name="CTL_DEEP_POWERDN_EN" value="false" />
<parameter name="CTL_SELF_REFRESH_EN" value="false" />
<parameter name="AUTO_POWERDN_EN" value="false" />
<parameter name="AUTO_PD CYCLES" value="0" />
<parameter name="CTL_USER_REFRESH_EN" value="false" />
<parameter name="CTL_AUTOPOCH_EN" value="false" />
<parameter name="CTL_ZQCAL_EN" value="false" />
<parameter name="ADDR_ORDER" value="0" />
<parameter name="CTL_LOOK_AHEAD_DEPTH" value="4" />
<parameter name="CONTROLER LATENCY" value="5" />
<parameter name="CFG_REORDER_DATA" value="true" />
<parameter name="STARVE LIMIT" value="10" />
<parameter name="CTL_CSR_ENABLED" value="false" />
<parameter name="CTL_CSR_CONNECTION" value="INTERNAL_JTAG" />
<parameter name="CTL_ECC_ENABLED" value="false" />
<parameter name="CTL_HBB_ENABLED" value="false" />
<parameter name="CTL_ECC_AUTO_CORRECTION_ENABLED" value="false" />
<parameter name="CTL_DYNAMIC_BANK_ALLOCATION" value="false" />
<parameter name="CTL_DYNAMIC_BANK_NUM" value="4" />
<parameter name="DEBUG_MODE" value="false" />
<parameter name="ENABLE_BURST_MERGE" value="false" />
<parameter name="ENABLE_BURST_INTERRUPT" value="true" />
<parameter name="ENABLE_BURST_TERMINATE" value="true" />
<parameter name="LOCAL_ID_WIDTH" value="8" />
<parameter name="WRBUFFER_ADDR_WIDTH" value="6" />
<parameter name="MAX_PENDING_WR_CMD" value="8" />
<parameter name="MAX_PENDING_RD_CMD" value="16" />
<parameter name="USE_MM_ADAPTER" value="true" />
<parameter name="USE_AXI_ADAPTER" value="false" />
<parameter name="HTX_COMPAT_MODE" value="false" />
<parameter name="CTL_CMD_QUEUE_DEPTH" value="8" />
<parameter name="CTL_CSR_READ_ONLY" value="1" />
<parameter name="CFG_DATA_REORDERING_TYPE" value="INTER_BANK" />
<parameter name="NUM_OF_PORTS" value="1" />
<parameter name="ENABLE_BONDING" value="false" />
<parameter name="ENABLE_USER_ECC" value="false" />
<parameter name="AVL_DATA_WIDTH_PORT" value="32,32,32,32,32,32" />
<parameter name="PRIORITY_PORT" value="1.1.1.1.1.1" />
<parameter name="WEIGHT_PORT" value="0.0,0.0,0.0" />
<parameter name="CPOR_TYPE_PORT" value="Bidirectional, Bidirectional, Bidirectional, Bidirectional, Bidirectional, Bidirectional" />
<parameter name="ENABLE_EMIT_BFM_MASTER" value="false" />
<parameter name="ENABLE_SEQUENCER_MARGINING_ON_BY_DEFAULT" value="false" />
<parameter name="REF_CLK_FREQ" value="25.0" />
<parameter name="REF_CLK_FREQ_PARAM_VALID" value="false" />
<parameter name="REF_CLK_FREQ_MIN_PARAM" value="0.0" />
<parameter name="REF_CLK_FREQ_MAX_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQPARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_DR_CLK_PHASE_PS_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_MULTgetParam" value="0.0" />
<parameter name="PLL_MEM_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_MEM_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_MEM_CLK_PHASE_PS_PARAM" value="0.0" />
<parameter name="PLL_MEM_CLK_MULTPARAM" value="0.0" />
<parameter name="PLL_AFI_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_CLK_PHASE_PS_PARAM" value="0.0" />
<parameter name="PLL_AFI_CLK_MULTPARAM" value="0.0" />
<parameter name="PLL_WRITE_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_WRITE_CLK_PHASE_PS_PARAM" value="0.0" />
<parameter name="PLL_WRITE_CLK_MULTPARAM" value="0.0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_PARAM" value="0.0" />
<parameter name="PLL_ADDR_CMD_CLK_MULT_PARAM" value="0.0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_PARAM" value="0"/>
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_SIM_STR_PARAM" value=""/>
<parameter name="PLL_AFI_HALF_CLK_MULT_PARAM" value="0"/>
<parameter name="PLL_AFI_HALF_CLK_DIV_PARAM" value="0"/>
<parameter name="PLL_NIOS_CLK_FREQ_PARAM" value="0.0"/>
<parameter name="PLL_NIOS_CLK_PHASE_PS_PARAM" value="0"/>
<parameter name="PLL_NIOS_CLK_PHASE_PS_SIM_STR_PARAM" value=""/>
<parameter name="PLL_NIOS_CLK_MULT_PARAM" value="0"/>
<parameter name="PLL_NIOS_CLK_DIV_PARAM" value="0"/>
<parameter name="PLL_CONFIG_CLK_FREQ_PARAM" value="0.0"/>
<parameter name="PLL_CONFIG_CLK_FREQ_SIM_STR_PARAM" value=""/>
<parameter name="PLL_CONFIG_CLK_PHASE_PS_PARAM" value="0"/>
<parameter name="PLL_CONFIG_CLK_PHASE_PS_SIM_STR_PARAM" value=""/>
<parameter name="PLL_CONFIG_CLK_MULT_PARAM" value="0"/>
<parameter name="PLL_CONFIG_CLK_DIV_PARAM" value="0"/>
<parameter name="PLL_P2C_READ_CLK_FREQ_PARAM" value="0.0"/>
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_PARAM" value="0"/>
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_SIM_STR_PARAM" value=""/>
<parameter name="PLL_P2C_READ_CLK_MULT_PARAM" value="0"/>
<parameter name="PLL_P2C_READ_CLK_DIV_PARAM" value="0"/>
<parameter name="PLL_C2P_WRITE_CLK_FREQ_PARAM" value="0.0"/>
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_PARAM" value="0"/>
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value=""/>
<parameter name="PLL_C2P_WRITE_CLK_MULT_PARAM" value="0"/>
<parameter name="PLL_C2P_WRITE_CLK_DIV_PARAM" value="0"/>
<parameter name="PLL_HR_CLK_FREQ_PARAM" value="0.0"/>
<parameter name="PLL_HR_CLK_FREQ_SIM_STR_PARAM" value=""/>
<parameter name="PLL_HR_CLK_PHASE_PS_PARAM" value="0"/>
<parameter name="PLL_HR_CLK_PHASE_PS_SIM_STR_PARAM" value=""/>
<parameter name="PLL_HR_CLK_MULT_PARAM" value="0"/>
<parameter name="PLL_HR_CLK_DIV_PARAM" value="0"/>
<parameter name="PLL_AFI_PHY_CLK_FREQ_PARAM" value="0.0"/>
<parameter name="PLL_AFI_PHY_CLK_FREQ_SIM_STR_PARAM" value=""/>
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_PARAM" value="0"/>
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_SIM_STR_PARAM" value=""/>
<parameter name="PLL_AFI_PHY_CLK_MULT_PARAM" value="0"/>
<parameter name="PLL_AFI_PHY_CLK_DIV_PARAM" value="0"/>
<parameter name="PLL_CLK_PARAM_VALID" value="false"/>
<parameter name="ENABLE_EXTRA_REPORTING" value="false"/>
<parameter name="NUM_EXTRA_REPORT_PATH" value="10"/>
<parameter name="ENABLE_ISS_PROBES" value="false"/>
<parameter name="CALIB_REG_WIDTH" value="8"/>
<parameter name="USE_SEQUENCER_BFM" value="false"/>
<parameter name="DEFAULT_FAST_SIM_MODEL" value="true"/>
<parameter name="PLL_SHARING_MODE" value="None"/>
<parameter name="NUM_PLL_SHARING_INTERFACES" value="1"/>
<parameter name="EXPORT_AFI_HALF_CLK" value="false"/>
<parameter name="ABSTRACT_REAL_COMPARE_TEST" value="false"/>
<parameter name="INCLUDE_BOARD_DELAY_MODEL" value="false"/>
<parameter name="INCLUDE_MULTIRANK_BOARD_DELAY_MODEL" value="false"/>
<parameter name="USE_FAKE_PHY" value="false"/>
<parameter name="FORCE_MAX_LATENCY_COUNT_WIDTH" value="0"/>
<parameter name="ENABLE_NON_DESTRUCTIVE_CALIB" value="false"/>
<parameter name="TRACKING_ERROR_TEST" value="false"/>
<parameter name="TRACKING_WATCH_TEST" value="false"/>
<parameter name="MARGIN_VARIATION_TEST" value="false"/>
<parameter name="EXTRA_SETTINGS" value=""/>
<parameter name="MEM_DEVICE" value="MISSING_MODEL"/>
<parameter name="FORCE_SYNTHESIS_LANGUAGE" value=""/>
<parameter name="SEQUENCER_TYPE" value="NIOS"/>
<parameter name="ADVERTISE_SEQUENCER_SW_BUILD_FILES" value="false"/>
<parameter name="FORCED_NUM_WRITE_FR_CYCLE_SHIFTS" value="0"/>
<parameter name="FORCED_NON_LDC_ADDR_CMD_MEM_CLK_INVERT" value="false"/>
<parameter name="PHY_ONLY" value="false"/>
<parameter name="SEQ_MODE" value="0"/>
<parameter name="ADD EFFICIENCY_MONITOR" value="false" />
<parameter name="ENABLE ABS RAM MEM_INIT" value="false" />
<parameter name="ABS_RAM_MEM_INIT_FILENAME" value="meminit" />
<parameter name="DLL_SHARING_MODE" value="None" />
<parameter name="NUM_DLL_SHARING_INTERFACES" value="1" />
<parameter name="OCT_SHARING_MODE" value="None" />
<parameter name="NUM_OCT_SHARING_INTERFACES" value="1" />
<parameter name="MPU_EVENTS_Enable" value="false" />
<parameter name="GP_Enable" value="false" />
<parameter name="DEBUGAPB_Enable" value="false" />
<parameter name="STM_Enable" value="false" />
<parameter name="CTI_Enable" value="false" />
<parameter name="TPU_FPGA_Enable" value="false" />
<parameter name="TEST_Enable" value="false" />
<parameter name="HLGPI_Enable" value="false" />
<parameter name="BSEL_EN" value="false" />
<parameter name="BSEL" value="1" />
<parameter name="CSEL_EN" value="false" />
<parameter name="CSEL" value="0" />
<parameter name="F2S_Width" value="2" />
<parameter name="S2F_Width" value="2" />
<parameter name="LWH2F_Enable" value="true" />
<parameter name="F2SDRAM_Type" value="" />
<parameter name="F2SDRAM_Width" value="" />
<parameter name="S2FCLK_COLDRST_Enable" value="false" />
<parameter name="S2FCLK_PENDINGRST_Enable" value="false" />
<parameter name="F2FCLKDBG_RST_Enable" value="false" />
<parameter name="F2FCLK_WARM_RST_Enable" value="false" />
<parameter name="F2FCLK_COLD_RST_Enable" value="false" />
<parameter name="NOF_Enable" value="No, No, No, No, No, No, No, No" />
<parameter name="S2FINTERRUPT_Enable" value="true" />
<parameter name="S2FINTERRUPT_CAN_Enable" value="false" />
<parameter name="S2FINTERRUPT_CLOCKPERIPHERAL_Enable" value="false" />
<parameter name="S2FINTERRUPT_CTI_Enable" value="false" />
<parameter name="S2FINTERRUPT_DMA_Enable" value="false" />
<parameter name="S2FINTERRUPT_EMAC_Enable" value="false" />
<parameter name="S2FINTERRUPT_FPGAMANAGER_Enable" value="false" />
<parameter name="S2FINTERRUPT_GPIO_Enable" value="false" />
<parameter name="S2FINTERRUPT_I2CEMAC_Enable" value="false" />
<parameter name="S2FINTERRUPT_I2CPeripheral_Enable" value="false" />
<parameter name="S2FINTERRUPT_J4TIMER_Enable" value="false" />
<parameter name="S2FINTERRUPT_NAND_Enable" value="false" />
<parameter name="S2FINTERRUPT_OsCTIMER_Enable" value="false" />
<parameter name="S2FINTERRUPT_QSPI_Enable" value="false" />
<parameter name="S2FINTERRUPT_SDMMC_Enable" value="false" />
<parameter name="S2FINTERRUPT_SPI_MASTER_Enable" value="false" />
<parameter name="S2FINTERRUPT_SPI_SLAVE_Enable" value="false" />
<parameter name="S2FINTERRUPT_UART_Enable" value="false" />
<parameter name="S2FINTERRUPT_USB_Enable" value="false" />
<parameter name="S2FINTERRUPT_WATCHDOG_Enable" value="false" />
<parameter name="EMAC0_Pinmuxing" value="Unused" />
<parameter name="EMAC0_Mode" value="N/A" />
<parameter name="EMAC1_Pinmuxing" value="HPS I/O Set 0" />
<parameter name="EMAC1_Mode" value="RGMII" />
<parameter name="NAND_Pinmuxing" value="Unused" />
<parameter name="NAND_Mode" value="N/A" />
<parameter name="QSPI_Pinmuxing" value="HPS I/O Set 0" />
<parameter name="QSPI_Mode" value="1 SS" />
<parameter name="SDIO_Pinmuxing" value="HPS I/O Set 0" />
<parameter name="SDIO_Mode" value="4-bit Data" />
<parameter name="USB0_Pinmuxing" value="Unused" />
<parameter name="USB0_Mode" value="N/A" />
<parameter name="USB1_Pinmuxing" value="HPS I/O Set 0" />
<parameter name="USB1_Mode" value="SDR" />
<parameter name="SPI0_Pinmuxing" value="HPS I/O Set 0" />
<parameter name="SPI0_Mode" value="Single Slave Select" />
<parameter name="SPI1_Pinmuxing" value="Unused" />
<parameter name="SPI_M1_Mode" value="N/A" />
<parameter name="SPI0_PinMuxing" value="Unused" />
<parameter name="SPI0_MODE" value="N/A" />
<parameter name="SPI1_PinMuxing" value="Unused" />
<parameter name="SPI1_MODE" value="N/A" />
<parameter name="UART0_PinMuxing" value="HPS 1/O Set 2" />
<parameter name="UART0_Mode" value="No Flow Control" />
<parameter name="UART1_PinMuxing" value="Unused" />
<parameter name="UART1_MODE" value="N/A" />
<parameter name="I2C0_PinMuxing" value="HPS 1/O Set 1" />
<parameter name="I2C0_MODE" value="I2C" />
<parameter name="I2C1_PinMuxing" value="Unused" />
<parameter name="I2C1_MODE" value="N/A" />
<parameter name="I2C2_PinMuxing" value="Unused" />
<parameter name="I2C2_MODE" value="N/A" />
<parameter name="I2C3_PinMuxing" value="Unused" />
<parameter name="I2C3_MODE" value="N/A" />
<parameter name="CAN0_PinMuxing" value="HPS 1/O Set 0" />
<parameter name="CAN0_MODE" value="CAN" />
<parameter name="CAN1_PinMuxing" value="Unused" />
<parameter name="CAN1_MODE" value="N/A" />
<parameter name="TRACE_PinMuxing" value="HPS 1/O Set 0" />
<parameter name="TRACE_MODE" value="HPS" />
<parameter name="GPIO_Enable" value="false" />
<parameter name="CAN0_PinMuxing" value="HPS 1/O Set 0" />
<parameter name="CAN0_MODE" value="CAN" />
<parameter name="CAN1_PinMuxing" value="Unused" />
<parameter name="CAN1_MODE" value="N/A" />
<parameter name="CAN2_PinMuxing" value="Unused" />
<parameter name="CAN2_MODE" value="N/A" />
<parameter name="CAN3_PinMuxing" value="Unused" />
<parameter name="CAN3_MODE" value="N/A" />
<parameter name="LOANIO_Enable" value="false" />
<parameter name="S2FCLK_USER00CLK_Enable" value="false" />
<parameter name="S2FCLK_USER00CLK_FREQ" value="50" />
<parameter name="S2FCLK_USER10CLK_Enable" value="false" />
<parameter name="S2FCLK_USER10CLK_FREQ" value="100" />
<parameter name="S2FCLK_USER20CLK_Enable" value="false" />
<parameter name="S2FCLK_USER20CLK_FREQ" value="100" />
<parameter name="F2SCLK_PERIPHCLK_Enable" value="false" />
<parameter name="F2SCLK_PERIPHCLK_FREQ" value="100" />
<parameter name="F2SCLK_SDRAMCLK_Enable" value="false" />
<parameter name="F2SCLK_SDRAMCLK_FREQ" value="100" />
<parameter name="F2H_AXI_CLK_ENABLE" value="false" />
<parameter name="F2H_AXI_CLK_FREQ" value="50000000" />
<parameter name="F2H_LW_AXI_CLK_ENABLE" value="false" />
<parameter name="F2H_LW_AXI_CLK_FREQ" value="50000000" />
<parameter name="F2H_SDRAM0_CLK_ENABLE" value="false" />
<parameter name="F2H_SDRAM0_CLK_FREQ" value="100" />
<parameter name="F2H_SDRAM1_CLK_ENABLE" value="false" />
<parameter name="F2H_SDRAM1_CLK_FREQ" value="100" />
<parameter name="F2H_SDRAM2_CLK_ENABLE" value="false" />
<parameter name="F2H_SDRAM2_CLK_FREQ" value="100" />
<parameter name="F2H_SDRAM3_CLK_ENABLE" value="false" />
<parameter name="F2H_SDRAM3_CLK_FREQ" value="100" />
<parameter name="F2H_SDRAM4_CLK_ENABLE" value="false" />
<parameter name="F2H_SDRAM4_CLK_FREQ" value="100" />
<parameter name="F2H_SDRAM5_CLK_ENABLE" value="false" />
<parameter name="F2H_SDRAM5_CLK_FREQ" value="100" />
<parameter name="F2H_CTI_CLK_ENABLE" value="false" />
<parameter name="F2H_CTI_CLK_FREQ" value="100" />
<parameter name="F2H_TPIU_CLK_ENABLE" value="false" />
<parameter name="F2H_TPIU_CLK_FREQ" value="100" />
<parameter name="F2H_DEBUG_APB_CLK_ENABLE" value="false" />
<parameter name="F2H_DEBUG_APB_CLK_FREQ" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_TX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_RX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_GTX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_GTX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_TX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_GTX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_QSPI_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SDIO_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SDIO_CCLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USBO_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USB1_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SPIM0_SCLK_OUT" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SPIM1_SCLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C1_SCL_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C0_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C1_SCL_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C2_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C3_SCL_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C3_CLK" value="100" />
<parameter name="device_name" value="5CSXFC6D6F31C8ES" />
<parameter name="quartus_ini.hps.ip.enable.all_peripheral.fpga_interfaces" value="false" />
<parameter name="quartus_ini.hps.ip_enable_emac0_peripheral_fpga_interface" value="false" />
<parameter name="quartus_ini.hps.ip_enable_test_interface" value="false" />
<parameter name="quartus_ini.hps_ip_fast_f2sdram_sim_model" value="false" />
<parameter name="quartus_ini.hps_ip_suppress_sdram_synth" value="false" />
<parameter name="quartus_ini.hps_ip_enable_loanio" value="false" />
<parameter name="quartus_ini.hps_ip_enable_low_speed_serial_fpga_interfaces" value="false" />
<parameter name="quartus_ini.hps_ip_enable_bsel_esel" value="false" />
</module>
<module kind="altera_jtag_avalon_master"
version="13.0"
enabled="1"
name="fpga_only_master">
<parameter name="USE_PLI" value="0" />
<parameter name="PLI_PORT" value="50000" />
<parameter name="COMPONENT_CLOCK" value="0" />
<parameter name="FIFO_DEPTHS" value="2" />
<parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
<parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
</module>
<module kind="intr_capturer"
version="100.99.98.97"
enabled="1"
name="intr_capturer_0">
<parameter name="NUM_INTR" value="32" />
<parameter name="AUTO_CLOCK_CLOCK_RATE" value="50000000" />
</module>
<module kind="clock_source" version="13.0" enabled="1" name="ext_clk_50">
<parameter name="clockFrequency" value="50000000" />
<parameter name="clockFrequencyKnown" value="true" />
<parameter name="inputClockFrequency" value="0" />
<parameter name="resetSynchronousEdges" value="NONE" />
</module>
<module kind="altera_jtag_avalon_master"
version="13.0"
enabled="1"
name="hps_only_master">
<parameter name="USE_PLI" value="0" />
<parameter name="PLI_PORT" value="50000" />
<parameter name="COMPONENT_CLOCK" value="0" />
<parameter name="FAST_VER" value="0" />
<parameter name="FIFO_DEPTHS" value="2" />
<parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
<parameter name="AUTO_DEVICE" value="5CSXF6D6F31C8ES" />
</module>

<module kind="altera_avalon_jtag_uart"
        version="13.0.1.99.2"
        enabled="1"
        name="jtag_uart">
<parameter name="allowMultipleConnections" value="true" />
<parameter name="hubInstanceID" value="0" />
<parameter name="readBufferDepth" value="64" />
<parameter name="readIRQThreshold" value="8" />
<parameter name="simInputCharacterStream" value="" />
<parameter name="simInteractiveOptions">INTERACTIVE_ASCII_OUTPUT</parameter>
<parameter name="useRegistersForReadBuffer" value="false" />
<parameter name="useRegistersForWriteBuffer" value="false" />
<parameter name="useRelativePathForSimFile" value="false" />
<parameter name="writeBufferDepth" value="64" />
<parameter name="write_IRQHandlerThreshold" value="8" />
<parameter name="avalonSpec" value="2.0" />
</module>

<module kind="altera_avalon_pio"
        version="13.0.1.99.2"
        enabled="1"
        name="button_pio">
<parameter name="bitClearEdgeCapReg" value="true" />
<parameter name="bitModifyingOutReg" value="false" />
<parameter name="captureEdge" value="true" />
<parameter name="direction" value="Input" />
<parameter name="edgeType" value="FALLING" />
<parameter name="generateIRQ" value="true" />
<parameter name="irqType" value="EDGE" />
<parameter name="resetValue" value="0" />
<parameter name="simDoTestBenchWiring" value="false" />
<parameter name="simDrivenValue" value="0" />
<parameter name="width" value="2" />
<parameter name="clockRate" value="50000000" />
</module>

<module kind="altera_avalon_pio"
        version="13.0.1.99.2"
        enabled="1"
        name="dipsw_pio">
<parameter name="bitClearEdgeCapReg" value="true" />
<parameter name="bitModifyingOutReg" value="false" />
<parameter name="captureEdge" value="true" />
<parameter name="direction" value="Input" />
<parameter name="edgeType" value="ANY" />
<parameter name="generateIRQ" value="true" />
<parameter name="irqType" value="EDGE" />
<parameter name="resetValue" value="0" />
<parameter name="simDoTestBenchWiring" value="false" />
<parameter name="simDrivenValue" value="0" />
<parameter name="width" value="4" />
<parameter name="clockRate" value="50000000" />
</module>

<module kind="altera_avalon_pio"
        version="13.0.1.99.2"
        enabled="1"
        name="led_pio">
<parameter name="bitClearEdgeCapReg" value="false" />
<parameter name="bitModifyingOutReg" value="true" />
<parameter name="captureEdge" value="false" />
<parameter name="direction" value="InOut" />
<parameter name="edgeType" value="RISING" />
<parameter name="generateIRQ" value="false" />
</module>

103
<parameter name="TIMING_TDQSCDCL" value="1200" />
<parameter name="TIMING_TDQSS" value="0.25" />
<parameter name="TIMING_TQSH" value="0.38" />
<parameter name="TIMING_TDSS" value="0.2" />
<parameter name="MEM_TINIT_US" value="500" />
<parameter name="MEM_IMRD_CLK" value="4" />
<parameter name="MEM_TRAS_NS" value="35.0" />
<parameter name="MEM_TRCD_NS" value="13.75" />
<parameter name="MEM_TRP_NS" value="13.75" />
<parameter name="MEM_TREFI_US" value="7.8" />
<parameter name="MEM_TRFC_NS" value="260.0" />
<parameter name="CFG_TCCD_NS" value="2.5" />
<parameter name="MEM_TWTR_NS" value="15.0" />
<parameter name="MEM_TWIR" value="4" />
<parameter name="MEM_TFAW_NS" value="40.0" />
<parameter name="MEM_TRRD_NS" value="7.5" />
<parameter name="MEM_TRTP_NS" value="7.5" />
<parameter name="RATE" value="Full" />
<parameter name="MEM_CLK_FREQ" value="400.0" />
<parameter name="USE_MEM_CLK_FREQ" value="false" />
<parameter name="FORCE_DQS_TRACKING" value="AUTO" />
<parameter name="FORCE_SHADOW_REGS" value="AUTO" />
<parameter name="MRS_MIRROR_PING_PONG_ATSO" value="false" />
<parameter name="SYS_INFO_DEVICE_FAMILY" value="Cyclone V" />
<parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM_VALID" value="false" />
<parameter name="SAVE_FRIENDLY_DEVICE_FAMILY_PARAM" value="" />
<parameter name="SPEED_GRADE" value="7" />
<parameter name="IS_ES_DEVICE" value="false" />
<parameter name="DISABLE_CHILD_MESSAGING" value="false" />
<parameter name="HARD_EMI" value="true" />
<parameter name="HHP_HPS" value="false" />
<parameter name="HHP_HPS_VERIFICATION" value="false" />
<parameter name="HHP_HPS_SIMULATION" value="false" />
<parameter name="HPS_PROTOCOL" value="DEFAULT" />
<parameter name="CUT_NEW_FAMILY_TIMING" value="true" />
<parameter name="POWER_OF_2QBUS" value="true" />
<parameter name="SOPC_COMPAT_RESET" value="false" />
<parameter name="AVL_MAX_SIZE" value="4" />
<parameter name="BYTEENABLE" value="true" />
<parameter name="ENABLE_CTRL_AVALON_INTERFACE" value="true" />
<parameter name="CTL_DEEP_POWERDN_EN" value="false" />
<parameter name="CTL_SELF_REFRESH_EN" value="false" />
<parameter name="AUTOPOWERDN_EN" value="false" />
<parameter name="AUTO_PD_CYCLES" value="0" />
<parameter name="CTLUSR_REFRESH_EN" value="false" />
<parameter name="CTL_AUTOPCH_EN" value="false" />
<parameter name="CTRL_LOCAL_EN" value="false" />
<parameter name="ADDRORDER" value="0" />
<parameter name="CTL_LOOK_AHEAD_DEPTH" value="4" />
<parameter name="CONTROLLER_LATENCY" value="5" />
<parameter name="CFG_REORDER_DATA" value="false" />
<parameter name="STARVE_LIMIT" value="10" />
<parameter name="CTL_CSR_ENABLED" value="false" />
<parameter name="CTL_CSR_CONNECTION" value="INTERNAL_JTAG" />
<parameter name="CTL_ECC_ENABLED" value="false" />
<parameter name="CTL_HBR_ENABLED" value="false" />
<parameter name="CTL_ECC_AUTO_CORRECTION_ENABLED" value="false" />
<parameter name="MULTICAST_EN" value="false" />
<parameter name="CTL_DYNAMIC_BANK_ALLOCATION" value="false" />
<parameter name="CTL_DYNAMIC_BANK_NUM" value="4" />
<parameter name="DEBUGMODE" value="false" />
<parameter name="ENABLE_BURST_MERGE" value="false" />
<parameter name="CTL_ENABLE_BURST_INTERRUPT" value="false" />
<parameter name="CTL_ENABLE_BURST_TERMINATE" value="false" />
<parameter name="LOCAL_ID_WIDTH" value="8" />
<parameter name="WRBUFFER_ADDR_WIDTH" value="6" />
<parameter name="MAX_PENDING_WR_CMD" value="8" />
<parameter name="MAX_PENDING_RD_CMD" value="16" />
<parameter name="USE_MM_ADAPTOR" value="true" />
<parameter name="USE_AXI_ADAPTOR" value="false" />
<parameter name="HCK_COMPATMODE" value="false" />
<parameter name="CTRL_CMD_QUEUEDEPTH" value="8" />
<parameter name="CTRL_CSR_READONLY" value="1" />
<parameter name="CFG_DATA_REORDERING_TYPE" value="INTER_BANK" />
<parameter name="NUM_OF_PORTS" value="1" />
<parameter name="ENABLE_BONDING" value="false" />
<parameter name="ENABLE_USER_ECC" value="false" />
<parameter name="AVL_DATA_WIDTH_PORT" value="256,32,32,32,32,32" />
<parameter name="PRIORITY_PORT" value="1.1.1.1.1.1" />
<parameter name="WEIGHT_PORT" value="0,0,0,0,0,0" />
<parameter name="CPORT_TYPE_PORT" value="Bidirectional, Bidirectional, Bidirectional," />
<parameter name="ENABLE_EMIT_BFM_MASTER" value="false" />
<parameter name="FORCE_SEQUENCE_TCL_DEBUG_MODE" value="false" />
<parameter name="ENABLE_SEQUENCE_MARGENING_ON_BY_DEFAULT" value="false" />
<parameter name="REF_CLK_FREQ" value="100.0" />
<parameter name="REF_CLK_FREQ_PARAM" value="0.0" />
<parameter name="REF_CLK_FREQ_MIN_PARAM" value="0.0" />
<parameter name="REF_CLK_FREQ_MAX_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_DR_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_DR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_DR_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_DR_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_MEM_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_MEM_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_MEM_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_WRITE_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_WRITE_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_ADDR_CMD_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_HALF_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_NIOS_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_NIOS_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_NIOS_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_CONFIG_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_CONFIG_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_CONFIG_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_P2C_READ_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_P2C_READ_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_C2P_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_C2P_WRITE_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_HIR_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_HIR_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_HIR_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_HIR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_HIR_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_HIR_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_PHY_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_PHY_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_CLK_PARAM_VALID" value="false" />
<parameter name="ENABLE_EXTRA_REPORTING" value="false" />
<parameter name="NUM_EXTRA_REPORT_PATH" value="10" />
<parameter name="ENABLE_ISS_PROBES" value="false" />
<parameter name="CALIB_REG_WIDTH" value="8" />
<parameter name="USE_SEQUENCER_BFM" value="false" />
<parameter name="DEFAULT_FAST_SIM_MODEL" value="true" />
<parameter name="PLL_SHARING_MODE" value="None" />
<parameter name="NUM_PLL_SHARING_INTERFACES" value="1" />
<parameter name="EXPORT_AFI_HALF_CLK" value="false" />
<parameter name="ABSTRACT_REAL_COMPARE_TEST" value="false" />
<parameter name="INCLUDE_BOARD_DELAY_MODEL" value="false" />
<parameter name="INCLUDE_MULTIRANK_BOARD_DELAY_MODEL" value="false" />
<parameter name="USE_FAKE_PHY" value="false" />
<parameter name="FORCE_MAX_LATENCY_COUNT_WIDTH" value="0" />
<parameter name="ENABLE_NON_DESTRUCTIVE_CALIB" value="false" />
<parameter name="TRACKING_ERROR_TEST" value="false" />
<parameter name="TRACKING_WATCH_TEST" value="false" />
<parameter name="MARGIN_VARIATION_TEST" value="false" />
<parameter name="EXTRA_SETTINGS" value="" />
<parameter name="MEM_DEVICE" value="MISSING_MODEL" />
<parameter name="FORCE_SYNTHESIS_LANGUAGE" value="" />
<parameter name="FORCED_NUM_WRITE_FR_CYCLE_SHIFTS" value="0" />
<parameter name="SEQUENCER_TYPE" value="NIOS" />
<parameter name="ADVERTIZE_SEQUENCER_SW_BUILD_FILES" value="false" />
<parameter name="FORCED_NON_LDC_ADDR_CMD_MEM_CLK_INVERT" value="false" />
<parameter name="PHY_ONLY" value="false" />
<parameter name="SEQ_MODE" value="0" />
<parameter name="ADVANCED_CLK_PHASES" value="false" />
<parameter name="COMMAND_PHASE" value="0.0" />
<parameter name="MEM_CLK_PHASE" value="0.0" />
<parameter name="P2C_READ_CLOCK_ADD_PHASE" value="0.0" />
<parameter name="C2P_WRITE_CLOCK_ADD_PHASE" value="0.0" />
<parameter name="ACV_PHY_CLK_ADD_FR_PHASE" value="0.0" />
<parameter name="MEM_VOLTAGE" value="1.5V_DDR3" />
<parameter name="PLL_LOCATION" value="Top, Bottom" />
<parameter name="SKIP_MEM_INIT" value="true" />
<parameter name="READ_DQ_DQS_CLOCK_SOURCE" value="INVERTED_DQS_BUS" />
<parameter name="DQ_INPUT_REG_USE_CLKN" value="false" />
<parameter name="DQS_DQSN_MODE" value="DIFFERENTIAL" />
<parameter name="AFI_DEBUG_INFO_WIDTH" value="32" />
<parameter name="CALIBRATION_MODE" value="Skip" />
<parameter name="NIOS_ROM_DATA_WIDTH" value="32" />
<parameter name="READ_FIFO_SIZE" value="8" />
<parameter name="PHY_CSR_ENABLED" value="false" />
<parameter name="PHY_CSR_CONNECTION" value="INTERNAL_JTAG" />
<parameter name="USER_DEBUG_LEVEL" value="1" />
<parameter name="TIMING_BOARD_DERATE_METHOD" value="AUTO" />
<parameter name="TIMING_BOARD_CK_CKIN_SLEW_RATE" value="2.0" />
<parameter name="TIMING_BOARD_AC_SLEW_RATE" value="1.0" />
<parameter name="TIMING_BOARD_DQS_DQSN_SLEW_RATE" value="2.0" />
<parameter name="TIMING_BOARD_DQ_SLEW_RATE" value="1.0" />
<parameter name="TIMING_BOARD_TIS" value="0.0" />
<parameter name="TIMING_BOARD_THI" value="0.0" />
<parameter name="TIMING_BOARD_TD" value="0.0" />
<parameter name="TIMING_BOARD_TDHI" value="0.0" />
<parameter name="TIMING_BOARD_JTAG" value="false" />
<parameter name="TIMING_BOARD_MAX_CK_DELAY" value="0.29132963" />
<parameter name="TIMING_BOARD_MAX_DQS_DELAY" value="0.286640405" />
<parameter name="TIMING_BOARD_CKDSK_CKDSK_DIMM_MIN" value="-0.005828107" />
<parameter name="TIMING_BOARD_CKDSK_CKDSK_DIMM_MAX" value="0.048251827" />
<parameter name="TIMING_BOARD_CKDSK BETWEEN DIMMS" value="0.05" />
<parameter name="TIMING_BOARD_CKDSK WITHIN_DQS" value="0.001782034" />
<parameter name="TIMING_BOARD_CKDSK BETWEEN DQS" value="0.029609881" />
<parameter name="TIMING_BOARD_DQ_TO_DQS_SKEW" value="0.0" />
<parameter name="TIMING_BOARD_DQ_TO_CK_SKEW" value="0.09664741" />
<parameter name="TIMING_BOARD_CK TO CKE SKEW" value="0.026243846" />
<parameter name="ENABLE_EXPORT_SEQ_DEBUG_BRIDGE" value="false" />
<parameter name="CORE_DEBUG_CONNECTION" value="EXPO" />
<parameter name="ADD_EXTERNAL_SEQ_DEBUG_NIOS" value="false" />
<parameter name="ED EXPORT SEQ_DEBUG" value="false" />
<parameter name="ADD EFFICIENCY MONITOR" value="false" />
<parameter name="ENABLE ABS RAM_MEM INIT" value="false" />
<parameter name="ABS_RAM_MEM_INIT_FILENAMES" value="meminit" />
<parameter name="NUM DLL SHARING INTERFACES" value="1" />
<parameter name="OCT SHARING MODE" value="None" />
<parameter name="NUM OCT SHARING INTERFACES" value="1" />
<parameter name="AUTO DEVICE" value="5CSXFC6D6F31C8ES" />
</module>

<module kind="clock_source" version="13.0" enabled="1" name="refclk_100" />
<parameter name="clockFrequency" value="100000000" />
<parameter name="clockFrequencyKnown" value="true" />
<parameter name="inputClockFrequency" value="0" />
<parameter name="resetSynchronousEdges" value="NONE" />
</module>

<module kind="altera_pll" version="13.0" enabled="1" name="pll_0" />
<parameter name="device_family" value="Cyclone V" />
<parameter name="gui_device_speed_grade" value="7" />
<parameter name="gui_pll_mode" value="Integer-N PLL" />
<parameter name="gui_reference_clock_frequency" value="400.0" />
<parameter name="gui_channel_spacing" value="0.0" />
<parameter name="gui_operation_mode" value="normal" />
<parameter name="gui_feedback_clock" value="Global Clock" />
<parameter name="gui_fractional_cout" value="32" />
<parameter name="gui_dsm_out_sel" value="1st_order" />
<parameter name="gui_use locked" value="false" />
<parameter name="gui_en_adv_params" value="false" />
<parameter name="gui_number_of_clocks" value="1" />
<parameter name="gui_multiply_params" value="true" />
<parameter name="gui_divide_factor_n" value="1" />
<parameter name="gui_output_clock_frequency0" value="100.0" />

108
<parameter name="gui_divide_factor_c0" value="1" />
<parameter name="gui_actual_output_clock_frequency0" value="0 MHz" />
<parameter name="gui_ps_units0" value="ps" />
<parameter name="gui_phase_shift0" value="0" />
<parameter name="gui_phase_shift_deg0" value="0" />
<parameter name="gui_actual_phase_shift0" value="0" />
<parameter name="gui_duty_cycle0" value="50" />
<parameter name="gui_output_clock_frequency1" value="100.0" />
<parameter name="gui_divide_factor_c1" value="1" />
<parameter name="gui_actual_output_clock_frequency1" value="0 MHz" />
<parameter name="gui_ps_units1" value="ps" />
<parameter name="gui_phase_shift1" value="0" />
<parameter name="gui_phase_shift_deg1" value="0" />
<parameter name="gui_actual_phase_shift1" value="0" />
<parameter name="gui_duty_cycle1" value="50" />
<parameter name="gui_output_clock_frequency2" value="100.0" />
<parameter name="gui_divide_factor_c2" value="1" />
<parameter name="gui_actual_output_clock_frequency2" value="0 MHz" />
<parameter name="gui_ps_units2" value="ps" />
<parameter name="gui_phase_shift2" value="0" />
<parameter name="gui_phase_shift_deg2" value="0" />
<parameter name="gui_actual_phase_shift2" value="0" />
<parameter name="gui_duty_cycle2" value="50" />
<parameter name="gui_output_clock_frequency3" value="100.0" />
<parameter name="gui_divide_factor_c3" value="1" />
<parameter name="gui_actual_output_clock_frequency3" value="0 MHz" />
<parameter name="gui_ps_units3" value="ps" />
<parameter name="gui_phase_shift3" value="0" />
<parameter name="gui_phase_shift_deg3" value="0" />
<parameter name="gui_actual_phase_shift3" value="0" />
<parameter name="gui_duty_cycle3" value="50" />
<parameter name="gui_output_clock_frequency4" value="100.0" />
<parameter name="gui_divide_factor_c4" value="1" />
<parameter name="gui_actual_output_clock_frequency4" value="0 MHz" />
<parameter name="gui_ps_units4" value="ps" />
<parameter name="gui_phase_shift4" value="0" />
<parameter name="gui_phase_shift_deg4" value="0" />
<parameter name="gui_actual_phase_shift4" value="0" />
<parameter name="gui_duty_cycle4" value="50" />
<parameter name="gui_output_clock_frequency5" value="100.0" />
<parameter name="gui_divide_factor_c5" value="1" />
<parameter name="gui_actual_output_clock_frequency5" value="0 MHz" />
<parameter name="gui_ps_units5" value="ps" />
<parameter name="gui_phase_shift5" value="0" />
<parameter name="gui_phase_shift_deg5" value="0" />
<parameter name="gui_actual_phase_shift5" value="0" />
<parameter name="gui_duty_cycle5" value="50" />
<parameter name="gui_output_clock_frequency6" value="100.0" />
<parameter name="gui_divide_factor_c6" value="1" />
<parameter name="gui_actual_output_clock_frequency6" value="0 MHz" />
<parameter name="gui_ps_units6" value="ps" />
<parameter name="gui_phase_shift6" value="0" />
<parameter name="gui_phase_shift_deg6" value="0" />
<parameter name="gui_actual_phase_shift6" value="0" />
<parameter name="gui_duty_cycle6" value="50" />
<parameter name="gui_output_clock_frequency7" value="100.0" />
<parameter name="gui_divide_factor_c7" value="1" />
<parameter name="gui_actual_output_clock_frequency7" value="0 MHz" />
<parameter name="gui_ps_units7" value="ps" />
<parameter name="gui_phase_shift7" value="0" />
<parameter name="gui_phase_shift_deg7" value="0" />
<parameter name="gui_actual_phase_shift7" value="0" />
<parameter name="gui_duty_cycle7" value="50" />
<parameter name="gui_output_clock_frequency8" value="100.0" />
<parameter name="gui_divide_factor_c8" value="1" />
<parameter name="gui_actual_output_clock_frequency8" value="0 MHz" />
<parameter name="gui_ps_units8" value="ps" />
<parameter name="gui_phase_shift8" value="0" />
<parameter name="gui_divide_factor\_c17" value="1" />
<parameter name="gui_actual\_output\_clock\_frequency\_c17" value="0 MHz" />
<parameter name="gui\_ps\_units\_c17" value="ps" />
<parameter name="gui\_phase\_shift\_c17" value="0" />
<parameter name="gui\_phase\_shift\_deg\_c17" value="0" />
<parameter name="gui\_actual\_phase\_shift\_c17" value="0" />
<parameter name="gui\_duty\_cycle\_c17" value="50" />
<parameter name="gui\_pll\_auto\_reset" value="Off" />
<parameter name="gui\_pll\_bandwidth\_preset" value="Auto" />
<parameter name="gui\_en\_reconf" value="false" />
<parameter name="gui\_en\_dps\_ports" value="false" />
<parameter name="gui\_en\_phout\_ports" value="false" />
<parameter name="gui\_mif\_generate" value="false" />
<parameter name="gui\_enable\_mif\_dps" value="false" />
<parameter name="gui\_dps\_cntn" value="C0" />
<parameter name="gui\_dps\_num" value="1" />
<parameter name="gui\_dps\_dir" value="Positive" />
<parameter name="gui\_refclk\_switch" value="false" />
<parameter name="gui\_refclk\_1\_frequency" value="100.0" />
<parameter name="gui\_switchover\_mode" value="Automatic Switchover" />
<parameter name="gui\_switchover\_delay" value="0" />
<parameter name="gui\_active\_clk" value="false" />
<parameter name="gui\_clk\_bad" value="false" />
<parameter name="gui\_enable\_cascade\_out" value="false" />
<parameter name="gui\_enable\_cascade\_in" value="false" />
<parameter name="gui\_pll\_cascading\_mode" value="Create an adppll in signal to connect with an upstream PLL" />
<parameter name="AUTO\_REFCLK\_CLOCK\_RATE" value="4000000000" />
</module>
<module kind="altera\_avalon\_mm\_clock\_crossing\_bridge"
  version="13.0"
  enabled="1"
  name="mm\_clock\_crossing\_bridge\_0">
  <parameter name="DATA\_WIDTH" value="32" />
  <parameter name="SYMBOL\_WIDTH" value="8" />
  <parameter name="ADDRESS\_WIDTH" value="28" />
  <parameter name="ADDRESS\_UNITS" value="WORDS" />
  <parameter name="MAX\_BURST\_SIZE" value="1" />
  <parameter name="COMMAND\_FIFO\_DEPTH" value="4" />
  <parameter name="RESPONSE\_FIFO\_DEPTH" value="4" />
  <parameter name="MASTER\_SYNC\_DEPTH" value="2" />
  <parameter name="SLAVE\_SYNC\_DEPTH" value="2" />
  <parameter name="AUTO\_M0\_CLK\_CLOCK\_RATE" value="1000000000" />
  <parameter name="AUTO\_SO\_CLK\_CLOCK\_RATE" value="500000000" />
  <parameter name="AUTO\_DEVICE\_FAMILY" value="Cyclone V" />
</module>
<module kind="ImgDiff" version="1.0" enabled="0" name="ImgDiff\_0">
  <parameter name="DATA\_WIDTH" value="32" />
  <parameter name="ADDRESS\_WIDTH" value="32" />
  <parameter name="MASTER\_WRITE\_BURST\_CAPABLE" value="0" />
  <parameter name="MASTER\_WRITE\_MAXIMUM\_BURST\_COUNT" value="2" />
  <parameter name="MASTER\_WRITE\_BURST\_COUNT\_WIDTH" value="2" />
  <parameter name="MASTER\_WRITE\_FIFO\_DEPTH" value="32" />
  <parameter name="MASTER\_WRITE\_FIFO\_DEPTH\_LOG2" value="5" />
  <parameter name="MASTER\_WRITE\_MEMORY\_BASED\_FIFO" value="1" />
  <parameter name="MASTER\_READ\_BURST\_CAPABLE" value="0" />
  <parameter name="MASTER\_READ\_MAXIMUM\_BURST\_COUNT" value="2" />
  <parameter name="MASTER\_READ\_BURST\_COUNT\_WIDTH" value="2" />
  <parameter name="MASTER\_READ\_FIFO\_DEPTH" value="32" />
  <parameter name="MASTER\_READ\_FIFO\_DEPTH\_LOG2" value="5" />
  <parameter name="MASTER\_READ\_MEMORY\_BASED\_FIFO" value="1" />
  <parameter name="IRQ\_EN" value="0" />
  <parameter name="AUTO\_CLK\_IN\_CLOCK\_RATE" value="1000000000" />
  <parameter name="AUTO\_DEVICE\_FAMILY" value="Cyclone V" />
</module>
<module kind="altera\_avalon\_mm\_clock\_crossing\_bridge"
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="1" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="0" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="2" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="0" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="1" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="2" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="0" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="1" />
</connection>
<connection kind="interrupt" version="13.0"
start="/intr_capturer_0.service" end="/intr_capturer_0.service">
<parameter name="irqNumber" value="2" />
</connection>
<connection kind=" avalon" version="13.0"
start="/hps_0.service" end="/jtag_uart.service">
<parameter name="arbitrationPriority" value="1" />
</connection>
<connection kind=" avalon" version="13.0"
start="/hps_0.service" end="/jtag_uart.service">
<parameter name="arbitrationPriority" value="1" />
</connection>
<connection kind=" clock" version="13.0"
start="/fpga_only.service" end="/fpga_only.service" />
<connection kind=" clock" version="13.0"
start="/ext_clk_50.service" end="/ext_clk_50.service" />
<connection kind=" clock" version="13.0"
start="/ext_clk_50.service" end="/ext_clk_50.service" />
<connection kind=" clock" version="13.0"
start="/ext_clk_50.service" end="/ext_clk_50.service" 

<connection kind="clock" version="13.0" start="ext_clk_50.clk" end="button_pio.clk" />
<connection kind="clock" version="13.0" start="ext_clk_50.clk" end="dipsw_pio.clk" />
<connection kind="clock" version="13.0" start="ext_clk_50.clk" end="led_pio.clk" />
<connection kind="clock" version="13.0" start="ext_clk_50.clk" end="sysid_qsys.control_slave" />
<connection kind="clock" version="13.0" start="ext_clk_50.clk" end="hps.0.h2f_lw_axi_clock" />
<connection kind="clock" version="13.0" start="ext_clk_50.clk" end="hps.0.f2h_axi_clock" />
<connection kind="avalon" version="13.0" start="fpga_only_master.master" end="jtag_uart.avalon_jtag_slave">
  <parameter name="arbitrationPriority" value="1" />
  <parameter name="baseAddress" value="0x00020000" />
  <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon" version="13.0" start="fpga_only_master.master" end="button_pio.sl1">
  <parameter name="arbitrationPriority" value="1" />
  <parameter name="baseAddress" value="0x000100c0" />
  <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon" version="13.0" start="fpga_only_master.master" end="dipsw_pio.sl1">
  <parameter name="arbitrationPriority" value="1" />
  <parameter name="baseAddress" value="0x00010080" />
  <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon" version="13.0" start="fpga_only_master.master" end="led_pio.sl1">
  <parameter name="arbitrationPriority" value="1" />
  <parameter name="baseAddress" value="0x00010040" />
  <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon" version="13.0" start="fpga_only_master.master" end="sysid_qsys.control_slave" >
<parameter name="arbitrationPriority" value="1" />
<parameter name="baseAddress" value="0x00010000" />
<parameter name="defaultConnection" value="false" />
</connection>
<connection>
  kind="avalon"
  version="13.0"
  start="fpga_only_master.master"
  end="intra_capturer_0.avalon_slave_0">
  <parameter name="arbitrationPriority" value="1" />
  <parameter name="baseAddress" value="0x00030000" />
  <parameter name="defaultConnection" value="false" />
</connection>
</connection>
<connection>
  kind="clock"
  version="13.0"
  start="ext_clk_50.clk"
  end="hps_only_master.clk"
</connection>
<connection>
  kind="avalon"
  version="13.0"
  start="hps_0.f2h_axi_slave"
  end="intr_capturer_0.avalon_slave_0">
  <parameter name="arbitrationPriority" value="1" />
  <parameter name="baseAddress" value="0x0000" />
  <parameter name="defaultConnection" value="false" />
</connection>
</connection>
<connection>
  kind="clock"
  version="13.0"
  start="ext_clk_50.clk"
  end="hps_only_master.clk"
</connection>
<connection>
  kind="avalon"
  version="13.0"
  start="hps_0.f2h_axi_slave"
  end="intr_capturer_0.avalon_slave_0">
  <parameter name="arbitrationPriority" value="1" />
  <parameter name="baseAddress" value="0x0000" />
  <parameter name="defaultConnection" value="false" />
</connection>
</connection>
<connection>
  kind="reset"
  version="13.0"
  start="ext_clk_50.clk_reset"
  end="intr_capturer_0.reset_sink"
</connection>
<connection>
  kind="reset"
  version="13.0"
  start="ext_clk_50.clk_reset"
  end="fpga_only_master.clk_reset"
</connection>
<connection>
  kind="reset"
  version="13.0"
  start="ext_clk_50.clk_reset"
  end="jtag_uart.reset"
</connection>
<connection>
  kind="reset"
  version="13.0"
  start="ext_clk_50.clk_reset"
  end="button_pio.reset"
</connection>
<connection>
  kind="reset"
  version="13.0"
  start="ext_clk_50.clk_reset"
  end="dipsw_pio.reset"
</connection>
<connection>
  kind="reset"
  version="13.0"
  start="ext_clk_50.clk_reset"
  end="led_pio.reset"
</connection>
<connection>
  kind="reset"
  version="13.0"
  start="ext_clk_50.clk_reset"
  end="sysid_qsys.reset"
</connection>
<connection>
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
end="hps_only_master.clk_reset" />
<connection
kind="clock"
version="13.0"
start="refclk.100.clk"
dern="mem_if_ddr3_fpga_pll_refclk" />
<connection
kind="clock"
version="13.0"
start="mem_if_ddr3_fpga_pll_refclk"
dern="p11_0.refclk" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_cmd_reset_n_0" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_rfifo_reset_n_0" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_wfifo_reset_n_0" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_rfifo_reset_n_1" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_wfifo_reset_n_1" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_rfifo_reset_n_2" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_wfifo_reset_n_2" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_rfifo_reset_n_3" />
<connection
kind="reset"
version="13.0"
start="ext_clk_50.clk_reset"
dern="mem_if_ddr3_fpga_pll_wfifo_reset_n_3" />
<connection
kind="clock"
version="13.0"
start="p11_0.outclk0"
<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.cmd_clk_0" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.rfifo_clk_0" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.r fifo_clk_1" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.rfifo_clk_2" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.rfifo_clk_3" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.wfifo_clk_0" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.wfifo_clk_1" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.wfifo_clk_2" />

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
end="mem_if_ddr3_fpga.mp.wfifo_clk_3" />

<connection kind="reset" version="13.0"
start="ext_clk_50.clk_reset"
end="refclk_100.clk_in_reset" />

<connection kind="avalon" version="13.0"
start="hps_0.h2f_axi_master"
end="mm_clock_crossing_bridge_0.s0">
</parameter name="arbitrationPriority" value="1" />
<parameter name="baseAddress" value="0x0000" />
<parameter name="defaultConnection" value="false" />
</connection>

<connection kind="avalon" version="13.0"
start="mm_clock_crossing_bridge_0.m0"
end="mem_if_ddr3_fpga.avl_0">
</parameter name="arbitrationPriority" value="1" />
<parameter name="baseAddress" value="0x0000" />
<parameter name="defaultConnection" value="false" />
</connection>

<connection kind="clock" version="13.0"
start="pll_0.outclk0"
<connection kind="clock" version="13.0">
    start="ext_clk_50_clk"
    end="mm_clock_crossing_bridge_0.s0_clk" />
<connection kind="reset" version="13.0">
    start="ext_clk_50_clk_reset"
    end="mm_clock_crossing_bridge_0.s0_reset" />
<connection kind="reset" version="13.0">
    start="refclk_100_clk_reset"
    end="mem_if_ddr3_fpga_soft_reset" />
<connection kind="reset" version="13.0">
    start="refclk_100_clk_reset"
    end="mm_clock_crossing_bridge_0.m0_reset" />
<connection kind="reset" version="13.0">
    start="ext_clk_50_clk_reset"
    end="ImgDiff_0.clk_in_reset" />
<connection kind="clock" version="13.0">
    start="pll_0.outclk0"
    end="ImgDiff_0.clk_in" />
<connection kind="avalon" version="13.0">
    start="ImgDiff_0.master_read"
    end="mem_if_ddr3_fpga_avl_0">
    <parameter name="arbitrationPriority" value="1" />
    <parameter name="baseAddress" value="0x0000" />
    <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon" version="13.0">
    start="ImgDiff_0.master_write"
    end="mem_if_ddr3_fpga_avl_0">
    <parameter name="arbitrationPriority" value="1" />
    <parameter name="baseAddress" value="0x0000" />
    <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon" version="13.0">
    start="hps_0.h2f_lw_axi_master"
    end="mm_clock_crossing_bridge_1.s0">
    <parameter name="arbitrationPriority" value="1" />
    <parameter name="baseAddress" value="0x00030000" />
    <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon" version="13.0">
    start="mm_clock_crossing_bridge_1.m0"
    end="ImgDiff_0.csr"/>
<parameter name="arbitrationPriority" value="1" />
<parameter name="baseAddress" value="0x0000" />
<parameter name="defaultConnection" value="false" />
</connection>
<connection kind="reset"
    version="13.0"
    start="ext_clk_50.clk_reset"
    end="mm_clock_crossing_bridge_1.s0_reset" />
<connection kind="clock"
    version="13.0"
    start="p11_0.outclk0"
    end="mm_clock_crossing_bridge_1.s0_clk" />
<connection kind="clock"
    version="13.0"
    start="ext_clk_50.clk"
    end="mm_clock_crossing_bridge_1.m0_clk" />
<connection kind="reset"
    version="13.0"
    start="ext_clk_50.clk_reset"
    end="mm_clock_crossing_bridge_1.m0_reset" />
<connection kind="avalon"
    version="13.0"
    start="mm_clock_crossing_bridge_1.m0"
    end="IPTest_0.csr">
<parameter name="arbitrationPriority" value="1" />
<parameter name="baseAddress" value="0x0800" />
<parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon"
    version="13.0"
    start="IPTest_0.master.write"
    end="mem_if_ddr3_fpga_avl_0">
<parameter name="arbitrationPriority" value="1" />
<parameter name="baseAddress" value="0x0000" />
<parameter name="defaultConnection" value="false" />
</connection>
<connection kind="avalon"
    version="13.0"
    start="IPTest_0.master.read"
    end="mem_if_ddr3_fpga_avl_0">
<parameter name="arbitrationPriority" value="1" />
<parameter name="baseAddress" value="0x0000" />
<parameter name="defaultConnection" value="false" />
</connection>
<connection kind="clock"
    version="13.0"
    start="p11_0.outclk0"
    end="IPTest_0.clk_in" />
<connection kind="reset"
    version="13.0"
    start="ext_clk_50.clk_reset"
    end="IPTest_0.clk_in_reset" />
<interconnectRequirement for="$system" name="qsys_mm.clockCrossingAdapter" value="FIFO" />
<interconnectRequirement for="$system" name="qsys_mm.maxAdditionalLatency" value="4" />
<interconnectRequirement for="hps_only_master.master" name="qsys_mm.security" value="" }

</system>
16.3.4 Top Level Verilog (orion_top.v)

```verilog
// Title:         orion_top.v
// Rev:           Rev 1
// Description:   Senior Design Top Level File (based on golden_top.v)

// Original Comments

// Title:         golden_top.v
// Rev:           Rev 5
// Created:       November 30, 2012
// Description:   Cyclone V SX SoC pinout and IO Standard example design

// Revision History:
// Rev 0: 1st cut

// Copyright 2012 Altera Corporation. All rights reserved. Altera products
// are protected under numerous U.S. and foreign patents, maskwork rights,
// copyrights and other intellectual property laws.
// This reference design file, and your use thereof, is subject to and
// governed by the terms and conditions of the applicable Altera Reference
// Design License Agreement. By using this reference design file, you
// indicate your acceptance of such terms and conditions between you and
// Altera Corporation. In the event that you do not agree with such terms and
// conditions, you may not use the reference design file. Please promptly
// destroy any copies you have made.
// This reference design file being provided on an “as-is” basis and as an
// accommodation and therefore all warranties, representations or guarantees
// of any kind (whether express, implied or statutory) including, without
// limitation, warranties of merchantability, non-infringement, or fitness for
// a particular purpose, are specifically disclaimed. By making this
// reference design file available, Altera expressly does not recommend,
// suggest or require that this reference design file be used in combination
// with any other product not provided by Altera.

`define HPS
`define HPS_GPIO
`define DDR3_FPGA
`define DDR3_HPS
`define HPS_USB
`define I2C_FPGA
`define MAX_FPGA

`define SDI_FPGA
`define ENET_FPGA
`define PCIe_FPGA

`define HSMA_XCVR
`define HSMA_LVDS
`define HSMA_PARALLEL
`define HSMA_CMOS
```
module orion_top ( )
// /*
// User-IO------------------------------------------------------//
output [3:0] user_led_fpga,
//GPLL-CLK------------------------------------------------------//
input clk_100m_fpga, //2.5V //100 MHz (2nd copy to max)
input clk_50m_fpga, //2.5V //50MHz (2nd copy to max)
input clk_top1, //2.5V //156.25 MHz adjustable
input clk_bot1, //1.5V //100 MHz adjustable
output ddr3_fpga_csn, //SSTL15 //Chip Select
input cpu_resetn, //2.5V //NIOS CPU Reset Pushbutton
//
// Hard Processor System (HPS) pins --------------------------//
'ifdef HPS

//HPS-CLK--INPUT-----------------------------------------------//
//input clk_osc1, //3.3V //Not for instantiation
//input clk_osc2, //3.3V //Not for instantiation

//HPS-Reset-----------------------------------------------------//
//input mictor_rstn, //3.3V //Cold Reset
//input hps_resetn, //3.3V //Warm Reset

//HPS-UART------------------------------------------------------//
input uart_rx, //3.3V LV //UART Receive
output uart_tx, //3.3V //UART Transmit

//HPS-I2C--------------------------------------------------------//
inout i2c_scl_hps, //3.3V //HPS I2C Clock output
inout i2c_sda_hps, //3.3V //HPS I2C Data Input/Output

//HPS-SPI-Bus----------------------------------------------------//
output spi_csn, //3.3V //Slave Sel 0 – LTC Analog
input spi_miso, //3.3V //Master Input
output spi_mosi, //3.3V //Master Output
output spi_sck, //3.3V //Clock Output

//HPS-QSPI–Flash--------------------------------------------------//
output qspi_clk, //3.3V //Clock
inout [3:0] qspi_io, //3.3V //Data
output qspi_ss0, //3.3V //Select

//HPS-SD–Card–Flash---------------------------------------------//
output sd_clk, //3.3V //
inout sd_cmd, //3.3V //
/output sd_pwren, //3.3V //
inout [3:0] sd_dat, //3.3V //???????????????????? should only be (3:0)
????????
/inout [7:0] sd_dat, //3.3V //???????????????????? should only be (3:0)
????????

//HPS-Ethernet-----------------------------------------------10/100/1000-------------------//
output enet_hps_gtx_clk, //3.3V //Gb Ethernet Clock
/output enet_hps_intn, //3.3V //Placed on HPS GPIO
output enet_hps_mdc, //3.3V //MDIO Clock (TR=0)
inout enet_hps_mdio, //3.3V //MDIO Data (TR=0)
inout enet_hps_rx_clk, //3.3V //Receive Data
input  enet_hps_rx_dv, // 3.3V // Receive Data Valid / Cont
input [3:0] enet_hps rxd, // 3.3V // Receive Data
output enet_hps tx_en, // 3.3V // Transmit Data Enable / Cont
output [3:0] enet_hps txd, // 3.3V // Transmit Data

// HPS-CAN-BUS ––/2 pins //
input can_0 rx, // 3.3V // HPS only
output can_0 tx, // 3.3V // HPS only

// HPS-Trace ––/10 pins //
//
// input mictor_rstn, // 3.3V // ????????????? not in Qsys – needed?
output [7:0] trace clk, // 3.3V //
output [7:0] trace_data, // 3.3V //

'endif

'ifdef HPS_GPIO

'ifdef HPS_USB

'ifdef DDR3_HPS
output [14:0] ddr3_hps_a, // SSTL15 // Address
output [2:0] ddr3_hps ba, // SSTL15 // Bank Address
output ddr3_hps casn, // SSTL15 // Column Address Strobe
output ddr3_hps cke, // SSTL15 // Clock Enable
output ddr3_hps clk n, // SSTL15 // Diff Clock – Neg
output ddr3_hps clk p, // SSTL15 // Diff Clock – Pos
output ddr3_hps csn, // SSTL15 // Chip Select
output [4:0] ddr3_hps dm, // SSTL15 // Data Write Mask
inout [39:0] ddr3_hps dq, // SSTL15 // Data Bus
inout [4:0] ddr3_hps dqsn, // SSTL15 // Diff Data Strobe – Neg
inout [4:0] ddr3_hps dqsp, // SSTL15 // Diff Data Strobe – Pos
output ddr3_hps ot, // SSTL15 // On-Die Termination Enable
output ddr3_hps rasn, // SSTL15 // Row Address Strobe
output ddr3_hps resetn, // SSTL15 // Reset
output ddr3_hps wen, // SSTL15 // Write Enable
input ddr3_hps rzy, // OCT rzy
// input ddr3_rup, // OCT_rup
// input ddr3_rdn, // OCT_rdn
'else
// output ddr3_hps csn, // SSTL15 // Chip Select
'endif

'endif

// FPGA Pins //

122
```c
// FPGA--CLK --------------------------------------------//X pins
input clk_100m_fpga, // 2.5V // 100 MHz (2nd copy to max)
input clk_50m_fpga, // 2.5V // 50 MHz (2nd copy to max)
input clk_25m_fpga,
input clk_top1, // 2.5V // 156.25 MHz adjustable
input clk_bot1, // 1.5V // 100 MHz adjustable
// input clk_enet_fpga_p, // LVDS // 125 MHz fixed

// FPGA--User--IO -------------------------------------//l1 pins -----------------------------------------------
input cpu_resetn, // 2.5V // NIOS CPU Reset Pushbutton
input [3:0] user_dipsw_fpga, //
output [3:0] user_led_fpga, //
input [1:0] user_pb_fpga, //

// FPGA--DDR3--400Mx32 --------------------------------//7l pins -----------------------------------------------
ifdef DDR3_FPGA
output [14:0] ddr3_fpga_a, // SSTL15 // Address
output [2:0] ddr3_fpga_ba, // SSTL15 // Bank Address
output ddr3_fpga_casn, // SSTL15 // Column Address Strobe
output ddr3_fpga_clk, // SSTL15 // Clock Enable
output ddr3_fpga_clk_n, // SSTL15 // Diff Clock - Neg
output ddr3_fpga_clk_p, // SSTL15 // Diff Clock - Pos
output ddr3_fpga_csn, // SSTL15 // Chip Select
output [3:0] ddr3_fpga_dm, // SSTL15 // Data Write Mask
output [31:0] ddr3_fpga_dq, // SSTL15 // Data Bus
output [3:0] ddr3_fpga_dqs_n, // SSTL15 // Diff Data Strobe - Neg
output [3:0] ddr3_fpga_dqs_p, // SSTL15 // Diff Data Strobe - Pos
output ddr3_fpga_odt, // SSTL15 // On-Die Termination Enable
input ddr3_fpga_rasn, // SSTL15 // Row Address Strobe
input ddr3_fpga_resetn, // SSTL15 // Reset
input ddr3_fpga_wen, // SSTL15 // Write Enable
input ddr3_fpga_rzq, // OCT_rzqin
else
output ddr3_fpga_csn, // SSTL15 // Chip Select
endif

// FPGA--Ethernet1 ---10/100 --------------------------//14 pins -----------------------------------------------
ifdef ENET_FPGA
input enet1_rx_clk, // 2.5V // Receive Data Clock
input enet1_rx_d, // 2.5V // Receive Data
input enet1_rx_dv, // 2.5V // Receive Data Valid
input enet1_rx_error, // 2.5V // Receive Data Error
output enet1_tx_clk_fb, // 2.5V // Transmit Clock Feedback
output [3:0] enet1_tx_clk, // 2.5V // Transmit Data
output enet1_tx_d, // 2.5V // Transmit Data
output enet1_tx_en, // 2.5V // Transmit Data Enable
input enet1_rx_error, // 2.5V // Transmit Data Error
output enet_dual_resetn, // 2.5V // EtherCat PHY Reset
input enet2_rx_clk, // 2.5V // Receive Data Clock
input [3:0] enet2_rx_d, // 2.5V // Receive Data
input enet2_rx_dv, // 2.5V // Receive Data Valid
input enet2_rx_error, // 2.5V // Receive Data Error
output enet2_tx_clk_fb, // 2.5V // Transmit Clock Feedback
output [3:0] enet2_tx_d, // 2.5V // Transmit Data
output enet2_tx_dv, // 2.5V // Transmit Data
output enet2Tx_en, // 2.5V // Transmit Data Enable
input enet2_rx_error, // 2.5V // Transmit Data Error
output enet_fpga_mdc, //
input enet_fpga_mdio
endif

// SDI--XCVR--Video -----------------------------------//X pins -----------------------------------------------
ifdef use_SD1_XCVR_refclk_p
input clk_148_p, // LVDS // 148.5 MHz Prog. SDI VCXO
endif
ifdef SDI_XCVR
input gxb_rx_l14_p, // PCML // SDI Receiver (or SMA)
```
//PCML
output  gxb_tx_i4.p,    //SDI Transmitter (or SMA)

#ifdef SDI_FPGA
//148.5M programmable VCXO in refclk section above
output  sdi_clk148_dn,  //2.5V  //VCXO pump up
output  sdi_clk148_up,  //2.5V  //VCXO pump down
input   sdi_fault,     //2.5V  //SDI Cable Driver Fault
output  sdi_rsti,      //2.5V  //SDI Cable Driver Reset
output  sdi_rx_bypass, //2.5V  //SDI Equalizer Bypass
output  sdi_rx_en,     //2.5V  //SDI Equalizer Enable
output  sdi_tx_en,     //2.5V  //SDI Cable Driver Enable
output  sdi_tx_sd,     //2.5V  //SDI Cable Driver High-Def
#endif

ifdef I2C_FPGA
inout i2c_sda_fpga,
inout i2c_scl_fpga,
#endif

ifdef MAX_FPGA
output max_fpga_miso,
input  max_fpga_mosi,
inout  max_fpga_sck,
inout  max_fpga_ssel,
#endif

ifdef use_HSMA_XCVR_refclk_p
input  refclk_q12_p,    //LVDS  //HSMA Transceiver Refclk -reqs OCT
#endif

ifdef HSMA_XCVR
input [3:0] hsma_rx_p, //PCML?  //HSMA Receive Data -reqs OCT
output [3:0] hsma_tx_p, //PCML?  //HSMA Transmit Data
#endif

ifdef HSMA_CMOS
//Enable below for CMOS HSMC
//Enable below for LVDS HSMC
input  hsma_clk_in0,    //2.5V  //Primary single-ended CLkin
input  hsma_clk_in1,    //LVCMOS signal
input  hsma_clk_in2,    //LVCMOS signal
input  hsma_clk_in3,    //LVCMOS signal
output hsma_clk_out0,   //2.5V  //Primary single-ended CLKOUT
output hsma_clk_out1,   //LVCMOS signal
output hsma_clk_out2,   //LVCMOS signal
output hsma_clk_out3,   //LVCMOS signal
input  [3:0] hsma_d,    //2.5V  //Dedicated CMOS IO
input [16:0] hsma_rx_d, //LVDS  //LVDS Source-Sync Input
output [16:0] hsma_tx_d, //LVDS  //LVDS Source-Sync Output
input  hsma_prsntn,    //2.5V  //HSMC Presence Detect Input
output hsma_scl,       //2.5V  //SMBus Clock
input  hsma_sda,       //2.5V  //SMBus Data
#endif

ifdef use_PCIE_XCVR_refclk_p
input  pcie_refclk_p,   //LVDS  //PCIe Refclk -reqs OCT
#endif

ifdef PCIE_XCVR
input [3:0] pcie_rx_p,  //PCML  //PCIe Receive Data-req’s OCT
output [3:0] pcie_tx_p,  //PCML  //PCIe Transmit Data
#endif

//---Found multiple nets with the same name
ifdef PCIE_FPGA
input pcle_perstn_in , // Pin W21 instance 0
output pcle_perstn_out , // Pin AG6 instance 1
input pcle_prsnt2_x1 ,
inout pcle_prsnt2_x4 ,
inout pcle_smbclk ,
inout pcle_smbdat ,
inout pcle_waken
'endif
);

// Rev A board Ethernet reset can interrupt device configuration, to fix
// enet_dual_reset tristate as input or drive as output to 1
assign enet_dual_resetn = 1'b1;

// FPGA user dipswitches drive user LEDs
assign user_led_fpga = ~user_dipsw_fpga;

// Disable FPGA DDR3
assign ddr3_fpga_csn = 1'b1;

// **** From golden_top by Altera Corp. ****
wire [1:0] fpga_debounced_buttons;
wire [3:0] fpga_led_internal;
wire hps_fpga_reset_n;

// connection of internal logics
assign fpga_led_internal = user_led_fpga;

// Debounce logic to clean out glitches within 1ms
debounce debounce_inst ( 
  .clk (clk_50m_fpga),
  .reset_n (hps_fpga_reset_n),
  .data_in (user_pb_fpga[0]),
  .data_out (fpga_debounced_buttons)
);
defparam debounce_inst.WIDTH = 2;
defparam debounce_inst.POLARITY = "LOW";
defparam debounce_inst.TIMEOUT = 50000; // at 50Mhz this is a debounce
time of 1ms
defparam debounce_inst.TIMEOUT_WIDTH = 16; // ceiling(log2(TIMEOUT))

// Due to PLLs the clock pins need to be buffered when connected internal clock network
wire clk_50m_net;
wire clk_100m_net;

// clk_ctrl clk_ctrl_50m ( 
//  .inclk (clk_50m_fpga),
//  .outclk (clk_50m_net )
//);
// clk_ctrl clk_ctrl_100m ( 
//  .inclk (clk_100m_fpga ),
//  .outclk (clk_100m_net )
//);

// **** IMPORTANT NOTE ****
// Need to use clk_bot1 (100 MHz (adjustable) to drive PLL due to fitter constraint
// DO NOT use clk_100m_fpga as it is in the wrong IO block for the FPGA
orion_system u0 ( 
  .memory_mem_a (ddr3_hps_a),
  .memory_mem_b (ddr3_hps_ba),
  .memory_mem_b_a (ddr3_hps_ba),
  .memory_mem_b_a (ddr3_hps_ba)
.hps_0_hps_io_sdio_inst_CMD (sd_cmd),

// .hps_0_hps_io_sdio_inst_PWREN (sd_pwren),

.hps_0_hps_io_sdio_inst_D0 (sd_dat[0]),

.hps_0_hps_io_sdio_inst_D1 (sd_dat[1]),

.hps_0_hps_io_sdio_inst_CLK (sd_clk),

.hps_0_hps_io_sdio_inst_D2 (sd_dat[2]),

.hps_0_hps_io_sdio_inst_D3 (sd_dat[3]),

.hps_0_hps_io_usb1_inst_D0 (usb_data[0]),

.hps_0_hps_io_usb1_inst_D1 (usb_data[1]),

.hps_0_hps_io_usb1_inst_D2 (usb_data[2]),

.hps_0_hps_io_usb1_inst_D3 (usb_data[3]),

.hps_0_hps_io_usb1_inst_D4 (usb_data[4]),

.hps_0_hps_io_usb1_inst_D5 (usb_data[5]),

.hps_0_hps_io_usb1_inst_D6 (usb_data[6]),

.hps_0_hps_io_usb1_inst_D7 (usb_data[7]),

.hps_0_hps_io_usb1_inst_CLK (usb_clk),

.hps_0_hps_io_usb1_inst_STP (usb_stp),

.hps_0_hps_io_usb1_inst_DIR (usb_dir),

.hps_0_hps_io_usb1_inst_NXT (usb_nxt),

.hps_0_hps_io_spi0_inst_CLK (spi_sck),

.hps_0_hps_io_spi0_inst_MOSI (spi_mosi),

.hps_0_hps_io_spi0_inst_MISO (spi_miso),

.hps_0_hps_io_spi0_inst_S0 (spi_csn),

.hps_0_hps_io_uart0_inst_RX (uart_rx),

.hps_0_hps_io_uart0_inst_TX (uart_tx),

.hps_0_hps_io_i2c0_inst_SDA (i2c_sda,hps),

.hps_0_hps_io_i2c0_inst_SCL (i2c_scl,hps),

.hps_0_hps_io_can0_inst_RX (can0_rx),

.hps_0_hps_io_can0_inst_TX (can0_tx),

.hps_0_hps_io_trace_inst_CLK (trace_clk,mic),

.hps_0_hps_io_trace_inst_D0 (trace_data[0]),

.hps_0_hps_io_trace_inst_D1 (trace_data[1]),

.hps_0_hps_io_trace_inst_D2 (trace_data[2]),

.hps_0_hps_io_trace_inst_D3 (trace_data[3]),
.hps_0.hps_io_hps_io_trace_inst_D4 (trace_data[4]), //
.hps_0.hps_io_hps_io_trace_inst_D5 (trace_data[5]), //
.hps_0.hps_io_hps_io_trace_inst_D6 (trace_data[6]), //
.hps_0.hps_io_hps_io_trace_inst_D7 (trace_data[7]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO04 (user_dipsw_hps[0]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO05 (user_dipsw_hps[1]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO06 (user_dipsw_hps[2]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO07 (user_dipsw_hps[3]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO08 (user_pb_hps[0]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO09 (user_pb_hps[1]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO10 (user_led_hps[3]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO11 (user_led_hps[2]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO12 (user_led_hps[1]), //
// .hps_0.hps_io_hps_io_gpio_inst_GPIO13 (user_led_hps[0]), //
.clk_50.clk (clk_50m_fpga), //
.led_pio_external_connection_in_port (user_led_fpga), //
.led_pio_external_connection_out_port (user_led_fpga), //
.dipsw_pio_external_connection_export (user_dipsw_fpga), //
.button_pio_external_connection_export (fpga_debounced_buttons), //
.hps_0.h2f_reset_reset_n (hps_fpga_reset_n), //
.fpga_memory_mem_a (ddr3_fpga_a), //
.fpga_memory_mem_b (ddr3_fpga_ba), //
/*
.fpga_memory_mem_ck (ddr3_fpga_clk_p),
fpga_memory_mem_ck_n (ddr3_fpga_clk_n),
fpga_memory_mem_ck_e (ddr3_fpga_cke),
fpga_memory_mem_cs_n (ddr3_fpga_csn),
fpga_memory_mem_dm (ddr3_fpga_dm),
fpga_memory_mem_ras_n (ddr3_fpga_rasn),
fpga_memory_mem_ras (ddr3_fpga_ras),
fpga_memory_mem_ras (ddr3_fpga_ras),
fpga_memory_mem_we_n (ddr3_fpga_wen),
fpga_memory_mem_reset_n (ddr3_fpga_resetn),
fpga_memory_mem_dq (ddr3_fpga_dq),
fpga_memory_mem_dqs (ddr3_fpga_dqs_p),
fpga_memory_mem_dqs (ddr3_fpga_dqs_n),
*/
}
498 .fpga_memory_mem_odt (ddr3_fpga_odt), //
499 .fpga_memory_oct_rzqin (ddr3_fpga_rzq), //
500 .refclk_100_clk (clk_bot1), //
501 .reset_50_reset_n (hps_fpga_reset_n), //
502 .mem_if_ddr3_fpga_status_local_init_done ( ), //
503 .mem_if_ddr3_fpga_status_local_cal_success ( ), //
504 .mem_if_ddr3_fpga_status_local_cal_fail ( ), //
505 .mem_if_ddr3_fpga_pll_sharing_pll_mem_clk ( ), //
506 .mem_if_ddr3_fpga_pll_sharing_pll_write_clk ( ), //
507 .mem_if_ddr3_fpga_pll_sharing_pll_write_clk_pre_phy_clk ( ), //
508 .mem_if_ddr3_fpga_pll_sharing_pll_addr_cmd_clk ( ), //
509 .mem_if_ddr3_fpga_pll_sharing_pll_cmd_clk ( ), //
510 .mem_if_ddr3_fpga_pll_sharing_pll_locked ( ), //
511 .mem_if_ddr3_fpga_pll_sharing_pll_avl_clk ( ), //
512 .mem_if_ddr3_fpga_pll_sharing_pll_config_clk ( ), //
513 .mem_if_ddr3_fpga_pll_sharing_pll_phy_clk ( ), //
514 .mem_if_ddr3_fpga_pll_sharing_pll Avl_phy_clk ( ) //
515 );
516
517 endmodule

Listing 16.3: Top Level Verilog (orion_top.v)

16.3.5 Timing Constraint File (timing.base.sdc)

```plaintext
#Timing constraint commands from DDR3 Demo from Board Test System by Altera Corporation
#Some Modifications by Christopher Yarp
derive_pll_clocks
derive_clock_uncertainty

#set_false_path -from * -to {sld_signaltap:auto_signaltap_0|+}
#set_false_path -from {sld_signaltap:auto_signaltap_0|+} -to *
#set sys_clock_hmc {u0|pll0|altera_pll_i|general[0].gpll.PLL_OUTPUT_COUNTER|divclk}
#set_false_path -from {get_ports hps_fpga_reset_n} -to *
# Timing file copied from Holden Hardware Reference Design (GHRD)
#modified for ports in use
# 50MHz board input clock
create_clock -period 20 {get_ports clk_50m_fpga}
create_clock -period 10 {get_ports clk_100m_fpga}
# for enhancing USB BlasterII to be reliable, 25MHz
create_clock -name {altera_reserved_tck} -period 40 {altera_reserved_tck}
```

129
set_input_delay -clock altera_reserved_tck -clock_fall 3 [get_ports altera_reserved_tdi]
set_input_delay -clock altera_reserved_tck -clock_fall 3 [get_ports altera_reserved_tms]
set_output_delay -clock altera_reserved_tck -clock_fall 3 [get_ports alteraReserved_tdo]

# FPGA IO port constraints
set_false_path -from [get_ports {fpga_button_pio[0]}] -to *
set_false_path -from [get_ports {fpga_button_pio[1]}] -to *
set_false_path -from [get_ports {fpga_dipsw_pio[0]}] -to *
set_false_path -from [get_ports {fpga_dipsw_pio[1]}] -to *
set_false_path -from [get_ports {fpga_dipsw_pio[2]}] -to *
set_false_path -from [get_ports {fpga_dipsw_pio[3]}] -to *
set_false_path -from + -to [get_ports {fpga_led_pio[0]}]
set_false_path -from + -to [get_ports {fpga_led_pio[1]}]
set_false_path -from + -to [get_ports {fpga_led_pio[2]}]
set_false_path -from + -to [get_ports {fpga_led_pio[3]}]

# HPS peripherals port false path setting to workaround the unconstrained path (setting false_path for hps.0 ports will not affect the routing as it is hard silicon)

# **** outputs ****
set_false_path -from + -to [get_ports {uart_tx}]
set_false_path -from + -to [get_ports {spi_csn}]
set_false_path -from + -to [get_ports {spi_mosi}]
set_false_path -from + -to [get_ports {spi_sck}]
set_false_path -from + -to [get_ports {qspi_clk}]
set_false_path -from + -to [get_ports {qspi_ss0}]
set_false_path -from + -to [get_ports {sd_clk}]
set_false_path -from + -to [get_ports {sd_pwren}]
set_false_path -from + -to [get_ports {enet_hps_gtx_clk}]
set_false_path -from + -to [get_ports {enet_hps_mdc}]
set_false_path -from + -to [get_ports {enet_hps_tx_en}]
set_false_path -from + -to [get_ports {enet_hps_txd[0]}]
set_false_path -from + -to [get_ports {enet_hps_txd[1]}]
set_false_path -from + -to [get_ports {enet_hps_txd[2]}]
set_false_path -from + -to [get_ports {enet_hps_txd[3]}]
set_false_path -from + -to [get_ports {can_0_tx}]
set_false_path -from + -to [get_ports {trace_clk_mic}]
set_false_path -from + -to [get_ports {trace_data[0]}]
set_false_path -from + -to [get_ports {trace_data[1]}]
set_false_path -from + -to [get_ports {trace_data[2]}]
set_false_path -from + -to [get_ports {trace_data[3]}]
set_false_path -from + -to [get_ports {trace_data[4]}]
set_false_path -from + -to [get_ports {trace_data[5]}]
set_false_path -from + -to [get_ports {trace_data[6]}]
set_false_path -from + -to [get_ports {trace_data[7]}]

# **** inputs ****
set_false_path -from [get_ports {uart_rx}] -to *
set_false_path -from [get_ports {spi_miso}] -to *
set_false_path -from [get_ports {enet_hps_rx_clk}] -to *
set_false_path -from [get_ports {enet_hps_rx_dv}] -to *
set_false_path -from [get_ports {enet_hps_rxd[0]}] -to *
set_false_path -from [get_ports {enet_hps_rxd[1]}] -to *
set_false_path -from [get_ports {enet_hps_rxd[2]}] -to *
set_false_path -from [get_ports {enet_hps_rxd[3]}] -to *
set_false_path -from [get_ports {can_0_rx}] -to *

# **** inouts ****
set_false_path -from + -to [get_ports {i2c_scl_hps}]
set_false_path -from + -to [get_ports {i2c_scl_hps}] -to *
set_false_path -from + -to [get_ports {i2c_sda_hps}]
set_false_path -from [get_ports {i2e_sda_hps}] -to *
set_false_path -from * -to [get_ports {qspi_io[0]}]
set_false_path -from [get_ports {qspi_io[0]}] -to *
set_false_path -from * -to [get_ports {qspi_io[1]}]
set_false_path -from [get_ports {qspi_io[1]}] -to *
set_false_path -from * -to [get_ports {qspi_io[2]}]
set_false_path -from [get_ports {qspi_io[2]}] -to *
set_false_path -from * -to [get_ports {qspi_io[3]}]
set_false_path -from [get_ports {qspi_io[3]}] -to *
set_false_path -from * -to [get_ports {sd_cmd}]
set_false_path -from [get_ports {sd_cmd}] -to *
set_false_path -from * -to [get_ports {sd_dat[0]}]
set_false_path -from [get_ports {sd_dat[0]}] -to *
set_false_path -from * -to [get_ports {sd_dat[1]}]
set_false_path -from [get_ports {sd_dat[1]}] -to *
set_false_path -from * -to [get_ports {sd_dat[2]}]
set_false_path -from [get_ports {sd_dat[2]}] -to *
set_false_path -from * -to [get_ports {sd_dat[3]}]
set_false_path -from [get_ports {sd_dat[3]}] -to *
set_false_path -from * -to [get_ports {enet_hps_mdio}]
set_false_path -from [get_ports {enet_hps_mdio}] -to *

# Set Multicycle Path
#***************************************************************************
# Set Maximum Delay
#***************************************************************************
# Set Minimum Delay
#***************************************************************************
# Set Input Transition
#***************************************************************************
16.3.6 Quartus II Project File (orion_system.qpf)

```
# Copyright (C) 1991-2012 Altera Corporation
# Your use of Altera Corporation's design tools, logic functions
# and other software and tools, and its AMPP partner logic
# functions, and any output files from any of the foregoing
# (including device programming or simulation files), and any
# associated documentation or information are expressly subject
# to the terms and conditions of the Altera Program License
# Subscription Agreement, Altera MegaCore Function License
# Agreement, or other applicable license agreement, including,
# without limitation, that your use is for the sole purpose of
# programming logic devices manufactured by Altera and sold by
# Altera or its authorized distributors. Please refer to the
# applicable agreement for further details.

# Quartus II 64-Bit
# Version 13.0 Internal Build 69 11/14/2012 SJ Full Version
# Date created = 18:45:45 November 15, 2012

# QUARTUS_VERSION = "13.0"
# DATE = "18:45:45 November 15, 2012"

# Revisions

PROJECT_REVISION = "orion_system"
```

16.3.7 Quartus II Settings File (orion_system.qsf)

```
# Copyright (C) 1991-2012 Altera Corporation
# Your use of Altera Corporation's design tools, logic functions
# and other software and tools, and its AMPP partner logic
# functions, and any output files from any of the foregoing
# (including device programming or simulation files), and any
# associated documentation or information are expressly subject
# to the terms and conditions of the Altera Program License
# Subscription Agreement, Altera MegaCore Function License
# Agreement, or other applicable license agreement, including,
# without limitation, that your use is for the sole purpose of
# programming logic devices manufactured by Altera and sold by
# Altera or its authorized distributors. Please refer to the
# applicable agreement for further details.

# Quartus II 64-Bit
# Version 13.0 Internal Build 69 11/14/2012 SJ Full Version
# Date created = 18:45:45 November 15, 2012
```

Listing 16.5: Quartus II Project File (orion_system.qpf)
# Notes:
# 1) The default values for assignments are stored in the file:
#    bts_xcvr_assignment_defaults.qdf
#    If this file does not exist, see file:
#    assignment_defaults.qdf

# 2) Altera recommends that you do not modify this file. This
#    file is updated automatically by the Quartus II software
#    and any changes you make may be lost or overwritten.

---

set_global_assignment -name FAMILY "Cyclone V"
set_global_assignment -name DEVICE 5CSXFC6D6F31CSES
set_global_assignment -name ORIGINAL_QUARTUS_VERSION 12.1
set_global_assignment -name PROJECT_CREATION_TIME_DATE "22:55:46 OCTOBER 03, 2012"
set_global_assignment -name LAST_QUARTUS_VERSION 13.0
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files_sof_rpt
set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 256
set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim- Altera (Verilog)"
set_global_assignment -name EDA_TIME_SCALE "1 ps" --section_id eda_simulation
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT "VERILOG HDL" --section_id eda_simulation
set_global_assignment -name TOP_LEVEL_ENTITY orion_top
set_global_assignment -name POWER_PRESET_COOLING_SOLUTION "23 MM HEAT SINK WITH 200 LFPM AIRFLOW"
set_global_assignment -name POWER_BOARD_THERMAL_MODEL "NONE (CONSERVATIVE)"
set_global_assignment -name UNIFY_SEQUENCE_DQS_CONFIG_ENABLE_ON
set_global_assignment -name OPTIMIZE_MULTI_CORNER_TIMING ON
set_global_assignment -name ECO_REGENERATE_REPORT ON
set_global_assignment -name STRATIX_DEVICE_IO_STANDARD "2.5 V"
set_global_assignment -name STRATIXV_CONFIGURATION_SCHEME "PASSIVE PARALLEL X16"
set_global_assignment -name USE_CONFIGURATIONDEVICE OFF
set_global_assignment -name GENERATE_RBF_FILE ON
set_global_assignment -name GENERATE_HEX_FILE ON
set_global_assignment -name CRC_ERROR_OPEN_DRAIN ON
set_global_assignment -name ON_CHIP_BISTSTREAM_DECOMPRESSION OFF
set_global_assignment -name RESERVE_DATA15_THROUGH_DATA8_AFTER_CONFIGURATION "AS INPUT TRI-STATE"
set_global_assignment -name RESERVE_DATA7_THROUGH_DATA5_AFTER_CONFIGURATION "AS INPUT TRI-STATE"
set_global_assignment -name OUTPUT_IO_TIMING_NEAR_END_VMEAS "HALF VCCIO" -rise
set_global_assignment -name OUTPUT_IO_TIMING_NEAR_END_VMEAS "HALF VCCIO" -fall
set_global_assignment -name OUTPUT_IO_TIMING_FAR_END_VMEAS "HALF SIGNAL SWING" -rise
set_global_assignment -name OUTPUT_IO_TIMING_FAR_END_VMEAS "HALF SIGNAL SWING" -fall
set_location_assignment PIN_D25 -to clk_oscl
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to clk_oscl
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_rzq -tag
    _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[0] -tag
    _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[1] -tag
    _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[2] -tag
    _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[3] -tag
    _hps_s dram_p0
    _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[5] -tag
    _hps_s dram_p0
    _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[7] -tag
    _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[8] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[9] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[10] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[11] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[12] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[13] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[14] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[15] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[16] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[17] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[18] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[19] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[20] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[21] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[22] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[23] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[24] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[25] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[26] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[27] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[28] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[29] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[30] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[31] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[32] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[33] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[34] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[35] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[36] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[37] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[38] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[39] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_p[0] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_p[1] -tag _hps_s dram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_p[2] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_p[3] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_p[4] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_n[0] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_n[1] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_n[2] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_n[3] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_dqs_n[4] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_clk_p -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_hps_clk_n -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[0] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[10] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[11] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[12] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[13] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[14] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[1] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[2] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[3] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[4] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[5] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[6] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[7] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[8] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[9] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[0] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[1] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[2] -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_casn -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_cke -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_csbn -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_odt -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_rasn -tag _hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_wen -tag _hps_sdram_p0

135
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_resethp

set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[0] -tag

set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[1] -tag

set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[2] -tag

set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[3] -tag


set_location_assignment PIN_F26 -to ddr3_hps_a[0]

set_location_assignment PIN_G30 -to ddr3_hps_a[1]

set_location_assignment PIN_F28 -to ddr3_hps_a[2]

set_location_assignment PIN_F30 -to ddr3_hps_a[3]

set_location_assignment PIN_J25 -to ddr3_hps_a[4]

set_location_assignment PIN_J27 -to ddr3_hps_a[5]

set_location_assignment PIN_F29 -to ddr3_hps_a[6]

set_location_assignment PIN_E28 -to ddr3_hps_a[7]

set_location_assignment PIN_H27 -to ddr3_hps_a[8]

set_location_assignment PIN_G26 -to ddr3_hps_a[9]

set_location_assignment PIN_D29 -to ddr3_hps_a[10]

set_location_assignment PIN_C30 -to ddr3_hps_a[11]

set_location_assignment PIN_B30 -to ddr3_hps_a[12]

set_location_assignment PIN_C29 -to ddr3_hps_a[13]

set_location_assignment PIN_H25 -to ddr3_hps_a[14]

set_location_assignment PIN_E29 -to ddr3_hps_ba[0]

set_location_assignment PIN_J24 -to ddr3_hps_ba[1]

set_location_assignment PIN_J23 -to ddr3_hps_ba[2]

set_location_assignment PIN_K28 -to ddr3_hps_dm[0]

set_location_assignment PIN_M28 -to ddr3_hps_dm[1]

set_location_assignment PIN_R28 -to ddr3_hps_dm[2]

set_location_assignment PIN_W30 -to ddr3_hps_dm[3]

set_location_assignment PIN_W27 -to ddr3_hps_dm[4]

set_location_assignment PIN_K23 -to ddr3_hps_dm[5]

set_location_assignment PIN_K22 -to ddr3_hps_dm[6]

set_location_assignment PIN_H30 -to ddr3_hps_dm[7]

set_location_assignment PIN_G28 -to ddr3_hps_dm[8]

set_location_assignment PIN_L25 -to ddr3_hps_dm[9]

set_location_assignment PIN_L24 -to ddr3_hps_dm[10]

set_location_assignment PIN_J30 -to ddr3_hps_dm[11]

set_location_assignment PIN_J29 -to ddr3_hps_dm[12]

set_location_assignment PIN_K26 -to ddr3_hps_dm[13]

set_location_assignment PIN_L26 -to ddr3_hps_dm[14]

set_location_assignment PIN_K29 -to ddr3_hps_dm[15]

set_location_assignment PIN_K27 -to ddr3_hps_dm[16]

set_location_assignment PIN_M26 -to ddr3_hps_dm[17]

set_location_assignment PIN_M27 -to ddr3_hps_dm[18]

set_location_assignment PIN_L28 -to ddr3_hps_dm[19]

set_location_assignment PIN_M30 -to ddr3_hps_dm[20]

set_location_assignment PIN_U26 -to ddr3_hps_dm[21]

set_location_assignment PIN_T26 -to ddr3_hps_dm[22]

set_location_assignment PIN_N29 -to ddr3_hps_dm[23]

set_location_assignment PIN_N28 -to ddr3_hps_dm[24]

set_location_assignment PIN_P26 -to ddr3_hps_dm[25]

set_location_assignment PIN_P27 -to ddr3_hps_dm[26]

set_location_assignment PIN_N27 -to ddr3_hps_dm[27]

set_location_assignment PIN_R29 -to ddr3_hps_dm[28]

set_location_assignment PIN_P24 -to ddr3_hps_dm[29]

set_location_assignment PIN_P25 -to ddr3_hps_dm[30]

set_location_assignment PIN_T29 -to ddr3_hps_dm[31]

set_location_assignment PIN_T28 -to ddr3_hps_dm[32]
set_location_assignment PIN_R24 to ddr3_hps_dq[33]
set_location_assignment PIN_U27 to ddr3_hps_dq[34]
set_location_assignment PIN_V28 to ddr3_hps_dq[35]
set_location_assignment PIN_T25 to ddr3_hps_dq[36]
set_location_assignment PIN_U25 to ddr3_hps_dq[37]
set_location_assignment PIN_V27 to ddr3_hps_dq[38]
set_location_assignment PIN_Y29 to ddr3_hps_dq[39]
set_location_assignment PIN_M19 to ddr3_hps_dqs_n[0]
set_location_assignment PIN_N24 to ddr3_hps_dqs_n[1]
set_location_assignment PIN_R18 to ddr3_hps_dqs_n[2]
set_location_assignment PIN_R21 to ddr3_hps_dqs_n[3]
set_location_assignment PIN_T23 to ddr3_hps_dqs_n[4]
set_location_assignment PIN_N18 to ddr3_hps_dqs_p[0]
set_location_assignment PIN_N25 to ddr3_hps_dqs_p[1]
set_location_assignment PIN_R19 to ddr3_hps_dqs_p[2]
set_location_assignment PIN_R22 to ddr3_hps_dqs_p[3]
set_location_assignment PIN_T24 to ddr3_hps_dqs_p[4]
set_location_assignment PIN_E27 to ddr3_hps_casn
set_location_assignment PIN_L29 to ddr3_hpscke
set_location_assignment PIN_L23 to ddr3_hps_clk_n
set_location_assignment PIN_M23 to ddr3_hps_clk_p
set_location_assignment PIN_H24 to ddr3_hps_cs
set_location_assignment PIN_H28 to ddr3_hps_odt
set_location_assignment PIN_D30 to ddr3_hpsras
set_location_assignment PIN_P30 to ddr3_hpsrstn
set_location_assignment PIN_C28 to ddr3_hpswe
set_location_assignment PIN_D27 to ddr3_hpszq
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to clk_osc2
set_location_assignment PIN_F25 to clk_osc2
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to mictor_rstn
set_location_assignment PIN_C27 to mictor_rstn
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to hps_rstn
set_location_assignment PIN_F23 to hps_rstn
set_instance_assignment -name IO_STANDARD "1.5 V" to user_dipsw_hps
set_location_assignment PIN_N30 to user_dipsw_hps[0]
set_location_assignment PIN_P29 to user_dipsw_hps[1]
set_location_assignment PIN_P22 to user_dipsw_hps[2]
set_location_assignment PIN_V20 to user_dipsw_hps[3]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to user_led_hps
set_location_assignment PIN_E17 to user_led_hps[0]
set_location_assignment PIN_E18 to user_led_hps[1]
set_location_assignment PIN_G17 to user_led_hps[2]
set_location_assignment PIN_C18 to user_led_hps[3]
set_instance_assignment -name IO_STANDARD "1.5 V" to user_pb_hps
set_location_assignment PIN_T30 to user_pb_hps[0]
set_location_assignment PIN_U28 to user_pb_hps[1]
set_location_assignment PIN_T21 to user_pb_hps[2]
set_location_assignment PIN_U20 to user_pb_hps[3]
set_location_assignment PIN_E24 to uart_rx
set_location_assignment PIN_D24 to uart_tx
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to uart_tx
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to i2c_scl
set_location_assignment PIN_D22 to i2c_scl_hps
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to i2c_sda_hps
set_location_assignment PIN_C23 to i2c_sda_hps
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to i2c_sda_hps
set_location_assignment PIN_H20 to spi_cs
set_location_assignment PIN_B23 to spi_miso
set_location_assignment PIN_C22 to spi_mosi
set_location_assignment PIN_A23 to spi_sclk
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to spi_cs
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to spi_miso
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to spi_mosi
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" to spi_sclk
set_location_assignment PIN_D19 to qspi_clk
set_location_assignment PIN_C20 to qspi_io[0]
set_location_assignment PIN_H18 to qspi_io[1]
set_location_assignment PIN_A19 to qspi_io[2]
| set_location_assignment | PIN_E19 to qspi_io[3] |
| set_location_assignment | PIN_A18 to qspi_sso |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to qspi_clk |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to qspi_io |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to qspi_sso |
| set_location_assignment | PIN_B16 to sd_dat[3] |
| set_location_assignment | PIN_A16 to sd_clk |
| set_location_assignment | PIN_F18 to sd_cmd |
| set_location_assignment | PIN_G18 to sd_data[0] |
| set_location_assignment | PIN_C17 to sd_data[1] |
| set_location_assignment | PIN_D17 to sd_data[2] |
| set_location_assignment | PIN_B17 to sd_pwr |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to sd_clk |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to sd_cmd |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to sd_pwr |
| set_location_assignment | PIN_N16 to usb_clk |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to usb_clk |
| set_location_assignment | PIN_E16 to usb_data[0] |
| set_location_assignment | PIN_G16 to usb_data[1] |
| set_location_assignment | PIN_D16 to usb_data[2] |
| set_location_assignment | PIN_D14 to usb_data[3] |
| set_location_assignment | PIN_A15 to usb_data[4] |
| set_location_assignment | PIN_C14 to usb_data[5] |
| set_location_assignment | PIN_D15 to usb_data[6] |
| set_location_assignment | PIN_M17 to usb_data[7] |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to usb_data |
| set_location_assignment | PIN_A14 to usb_nxt |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to usb_nxt |
| set_location_assignment | PIN_E14 to usb_dir |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to usb_stp |
| set_location_assignment | PIN_C15 to usb_stp |
| set_location_assignment | PIN_H19 to enet_hps_gtx_clk |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to enet_hps_gtx_clk |
| set_location_assignment | PIN_B21 to enet_hps_mdc |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to enet_hps_mdc |
| set_location_assignment | PIN_E21 to enet_hps_mdio |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to enet_hps_mdio |
| set_location_assignment | PIN_G20 to enet_hps_rx_clk |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to enet_hps_rx_clk |
| set_location_assignment | PIN_K17 to enet_hps_rx_dv |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to enet_hps_rx_dv |
| set_location_assignment | PIN_A21 to enet_hps_rxd[0] |
| set_location_assignment | PIN_B20 to enet_hps_rxd[1] |
| set_location_assignment | PIN_B18 to enet_hps_rxd[2] |
| set_location_assignment | PIN_D21 to enet_hps_rxd[3] |
| set_location_assignment | PIN_A20 to enet_hps_tx_en |
| set_location_assignment | PIN_F20 to enet_hps_txd[0] |
| set_location_assignment | PIN_J19 to enet_hps_txd[1] |
| set_location_assignment | PIN_F21 to enet_hps_txd[2] |
| set_location_assignment | PIN_F19 to enet_hps_txd[3] |
| set_instance_assignment | name IO_STANDARD "2.5 V" to enet_hps_rxd |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to enet_hps_tx_en |
| set_instance_assignment | name IO_STANDARD "2.5 V" to enet_hps_txd |
| set_location_assignment | PIN_C19 to enet_hps_intn |
| set_location_assignment | PIN_B22 to can_0_rx |
| set_location_assignment | PIN_G22 to can_0_tx |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to can_0_rx |
| set_instance_assignment | name IO_STANDARD "3.3-V LVCMOS" to can_0_tx |
| set_location_assignment | PIN_B26 to trace_clk мик |
| set_location_assignment | PIN_B25 to trace_data[0] |
| set_location_assignment | PIN_C25 to trace_data[1] |
| set_location_assignment | PIN_A25 to trace_data[2] |
| set_location_assignment | PIN_H23 to trace_data[3] |
| set_location_assignment | PIN_A24 to trace_data[4] |
| set_location_assignment | PIN_G21 to trace_data[5] |
| set_location_assignment | PIN_C24 to trace_data[6] |
| set_location_assignment | PIN_E23 to trace_data[7] |
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to trace_clk_mic
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to trace_data
set_instance_assignment -name IO_STANDARD "2.5 V" -to usb_dir
set_location_assignment PIN_AB27 -to clk_100m_fpga
set_location_assignment -name IO_STANDARD "2.5 V" -to clk_100m_fpga
set_location_assignment PIN_AC18 -to clk_50m_fpga
set_location_assignment -name IO_STANDARD "1.5 V" -to clk_50m_fpga
set_location_assignment PIN_AF14 -to clk_bot1
set_location_assignment -name IO_STANDARD "1.5 V" -to clk_bot1
set_location_assignment PIN_Y26 -to clk_enet_fpga_p
set_location_assignment -name IO_STANDARD LVDS -to clk_enet_fpga_p
set_location_assignment PIN_AA26 -to clk_top1
set_location_assignment -name IO_STANDARD "2.5 V" -to clk_top1
set_location_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_a
set_location_assignment PIN_AJ14 -to ddr3_fpga_a[0]
set_location_assignment PIN_AK14 -to ddr3_fpga_a[1]
set_location_assignment PIN_AH12 -to ddr3_fpga_a[2]
set_location_assignment PIN_AJ12 -to ddr3_fpga_a[3]
set_location_assignment PIN_AG15 -to ddr3_fpga_a[4]
set_location_assignment PIN_AH15 -to ddr3_fpga_a[5]
set_location_assignment PIN_AK12 -to ddr3_fpga_a[6]
set_location_assignment PIN_AK13 -to ddr3_fpga_a[7]
set_location_assignment PIN_AH13 -to ddr3_fpga_a[8]
set_location_assignment PIN_AH14 -to ddr3_fpga_a[9]
set_location_assignment PIN_AJ9 -to ddr3_fpga_a[10]
set_location_assignment PIN_AK9 -to ddr3_fpga_a[11]
set_location_assignment PIN_AK7 -to ddr3_fpga_a[12]
set_location_assignment PIN_AK8 -to ddr3_fpga_a[13]
set_location_assignment PIN_AG12 -to ddr3_fpga_a[14]
set_location_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_ba
set_location_assignment PIN_AH10 -to ddr3_fpga_ba[0]
set_location_assignment PIN_AJ11 -to ddr3_fpga_ba[1]
set_location_assignment PIN_AK11 -to ddr3_fpga_ba[2]
set_location_assignment PIN_AH7 -to ddr3_fpga_casn
set_location_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_casn -tag
_orion_system_mem_if_ddr3_fpga_p0
set_location_assignment PIN_AJ21 -to ddr3_fpga_cke
set_location_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_cke -tag
_orion_system_mem_if_ddr3_fpga_p0
set_location_assignment PIN_AA15 -to ddr3_fpga_clk_n
set_location_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_clk_n -tag
_orion_system_mem_if_ddr3_fpga_p0
set_location_assignment PIN_AA14 -to ddr3_fpga_clk_p
set_location_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_clk_p -tag
_orion_system_mem_if_ddr3_fpga_p0
set_location_assignment PIN_AB15 -to ddr3_fpga_csn
set_location_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_csn -tag
_orion_system_mem_if_ddr3_fpga_p0
set_location_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dm
set_location_assignment PIN_AH17 -to ddr3_fpga_dm[0]
set_location_assignment PIN_AG23 -to ddr3_fpga_dm[1]
set_location_assignment PIN_AK23 -to ddr3_fpga_dm[2]
set_location_assignment PIN_AJ27 -to ddr3_fpga_dm[3]
set_location_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq
set_location_assignment PIN_AF18 -to ddr3_fpga_dq[0]
set_location_assignment PIN_AE17 -to ddr3_fpga_dq[1]
set_location_assignment PIN_AG16 -to ddr3_fpga_dq[2]
set_location_assignment PIN_AF16 -to ddr3_fpga_dq[3]
set_location_assignment PIN_AH20 -to ddr3_fpga_dq[4]
set_location_assignment PIN_AG21 -to ddr3_fpga_dq[5]
set_location_assignment PIN_AJ16 -to ddr3_fpga_dq[6]
set_location_assignment PIN_AH18 -to ddr3_fpga_dq[7]
set_location_assignment PIN_AK18 -to ddr3_fpga_dq[8]
set_location_assignment PIN_AJ17 -to ddr3_fpga_dq[9]
set_location_assignment PIN_AG18 -to ddr3_fpga_dq[10]
set_location_assignment PIN_AK19 -to ddr3_fpga_dq[11]
set_location_assignment PIN_AG20 -to ddr3_fpga_dq[12]
set_location_assignment PIN_AF19 -to ddr3_fpga_dq[13]
set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_scl
set_location_assignment PIN_J2 -to hsma_rx_p[3]
set_location_assignment PIN_L2 -to hsma_rx_p[2]
set_location_assignment PIN_N2 -to hsma_rx_p[1]
set_location_assignment PIN_R2 -to hsma_rx_p[0]
set_location_assignment PIN_J1 -to hsma_rx_n[3]
set_location_assignment PIN_L1 -to hsma_rx_n[2]
set_location_assignment PIN_N1 -to hsma_rx_n[1]
set_location_assignment PIN_R1 -to hsma_rx_n[0]
set_location_assignment PIN_F15 -to hsma_rx_d_p[16]
set_location_assignment PIN_C13 -to hsma_rx_d_p[15]
set_location_assignment PIN_F13 -to hsma_rx_d_p[14]
set_location_assignment PIN_E12 -to hsma_rx_d_p[13]
set_location_assignment PIN_D11 -to hsma_rx_d_p[12]
set_location_assignment PIN_B6 -to hsma_rx_d_p[10]
set_location_assignment PIN_F11 -to hsma_rx_d_p[9]
set_location_assignment PIN_F9 -to hsma_rx_d_p[8]
set_location_assignment PIN_G10 -to hsma_rx_d_p[7]
set_location_assignment PIN_H8 -to hsma_rx_d_p[6]
set_location_assignment PIN_J7 -to hsma_rx_d_p[5]
set_location_assignment PIN_G12 -to hsma_rx_d_p[4]
set_location_assignment PIN_K7 -to hsma_rx_d_p[3]
set_location_assignment PIN_J10 -to hsma_rx_d_p[2]
set_location_assignment PIN_K12 -to hsma_rx_d_p[1]
set_location_assignment PIN_H14 -to hsma_rx_d_p[0]
set_instance_assignment -name IO_STANDARD LVDS -to hsma_rx_d_p
set_location_assignment PIN_F14 -to hsma_rx_d_n[16]
set_location_assignment PIN_B12 -to hsma_rx_d_n[15]
set_location_assignment PIN_E13 -to hsma_rx_d_n[14]
set_location_assignment PIN_D12 -to hsma_rx_d_n[13]
set_location_assignment PIN_D10 -to hsma_rx_d_n[12]
set_location_assignment PIN_D9 -to hsma_rx_d_n[11]
set_location_assignment PIN_B5 -to hsma_rx_d_n[10]
set_location_assignment PIN_E11 -to hsma_rx_d_n[9]
set_location_assignment PIN_F8 -to hsma_rx_d_n[8]
set_location_assignment PIN_F10 -to hsma_rx_d_n[7]
set_location_assignment PIN_G8 -to hsma_rx_d_n[6]
set_location_assignment PIN_H7 -to hsma_rx_d_n[5]
set_location_assignment PIN_G11 -to hsma_rx_d_n[4]
set_location_assignment PIN_K8 -to hsma_rx_d_n[3]
set_location_assignment PIN_J9 -to hsma_rx_d_n[2]
set_location_assignment PIN_J12 -to hsma_rx_d_n[1]
set_location_assignment PIN_G13 -to hsma_rx_d_n[0]
set_location_assignment PIN_AD12 -to hsma_prsnnt
set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_prsnnt
set_location_assignment PIN_AG1 -to hsma_d[3]
set_location_assignment PIN_AG7 -to hsma_d[2]
set_location_assignment PIN_AF8 -to hsma_d[1]
set_location_assignment PIN(AF9 -to hsma_d[0]
set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_d
set_location_assignment PIN_A10 -to hsma_clk_out0
set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_clk_out0
set_location_assignment PIN_E7 -to hsma_clk_out_p2
set_instance_assignment -name IO_STANDARD LVDS -to hsma_clk_out_p2
set_location_assignment PIN_E6 -to hsma_clk_out_n2
set_location_assignment PIN_K14 -to hsma_clk_in0
set_instance_assignment -name IO_STANDARD "2.5 V" -to hsma_clk_in0
set_location_assignment PIN_H15 -to hsma_clk_in_p2
set_instance_assignment -name IO_STANDARD LVDS -to hsma_clk_in_p2
set_location_assignment PING15 -to hsma_clk_in_n2
set_location_assignment PIN_AB22 -to enet1_tx_en
set_location_assignment PIN_AB26 -to enet1_tx_d[3]
set_location_assignment PIN_AA25 -to enet1_tx_d[2]
set_location_assignment PIN_Y21 -to enet1_tx_d[1]
set_location_assignment PIN_W20 -to enet1_tx_d[0]
set_location_assignment PIN_W25 -to enet1_clk_fb
set_location_assignment PIN_AE28 -to enet1_rx_error
set_location_assignment PIN_AB4  to  pcie_tx_p [1]
set_location_assignment PIN_Y4  to  pcie_rx_p [2]
set_location_assignment PIN_V4  to  pcie_rx_p [3]
set_location_assignment PIN_AF13  to  usb_b2_clk
set_location_assignment PIN_AK28  to  usb_b2_data [0]
set_location_assignment PIN_AD20  to  usb_b2_data [1]
set_location_assignment PIN_AD21  to  usb_b2_data [2]
set_location_assignment PIN_Y19  to  usb_b2_data [3]
set_location_assignment PIN_AA20  to  usb_b2_data [4]
set_location_assignment PIN_AH27  to  usb_b2_data [5]
set_location_assignment PIN_AF25  to  usb_b2_data [6]
set_location_assignment PIN_AC22  to  usb_b2_data [7]
set_instance_assignment  -name IO_STANDARD "1.5 V"  to  usb_b2_clk
set_instance_assignment  -name IO_STANDARD "1.5 V"  to  usb_b2_data
set_location_assignment  -name IO_STANDARD  to  cpu_reseth
set_location_assignment  -name IO_STANDARD  to  enet1_tx_error
set_location_assignment  -name IO_STANDARD  to  enet2_tx_error
set_location_assignment  -name IO_STANDARD  to  clk_148_p
set_global_assignment  -name ENABLE.disable.TAP  OFF
set_global_assignment  -name USE SIGNAL TAP_FILE  output_files_sof.rpt/stp1.stp
set_global_assignment  -name MAX_CORE_JUNCTION_TEMP  85
set_location_assignment  -name RX_IN  to  i2c_scl_fpga
set_location_assignment  -name RX_IN  to  i2c_sda_fpga
set_location_assignment  -name RX_IN  to  i2c_scl_fpga
set_location_assignment  -name RX_IN  to  i2c_sda_fpga
set_location_assignment  -name RX_IN  to  usb_empty
set_location_assignment  -name RX_IN  to  usb_full
set_location_assignment  -name RX_IN  to  usb_oen
set_location_assignment  -name RX_IN  to  usb_rdn
set_location_assignment  -name RX_IN  to  usb_resethn
set_location_assignment  -name RX_IN  to  usb_scl
set_location_assignment  -name RX_IN  to  usb_sda
set_location_assignment  -name RX_IN  to  usb_wrn
set_location_assignment  -name RX_IN  to  enet.fpga_mdc
set_location_assignment  -name RX_IN  to  enet.fpga_mdio
set_location_assignment  -name RX_IN  to  clk_148_n
set_location_assignment  -name RX_IN  to  clk_enet_fpga_n
set_location_assignment  -name RX_IN  to  gxb_rx_i4_n
set_location_assignment  -name RX_IN  to  gxb_rx_i4_p
set_location_assignment  -name RX_IN  to  gxb_tx_i4_n
set_location_assignment  -name RX_IN  to  gxb_tx_i4_p
set_location_assignment  -name RX_IN  to  refclk_q12_n
set_location_assignment  -name RX_IN  to  refclk_q12_p
set_location_assignment  -name RX_IN  to  refclk_q12_p
set_location_assignment  -name RX_IN  to  pcie_smbclk
set_location_assignment  -name RX_IN  to  pcie_smbdat
set_location_assignment  -name RX_IN  to  pcie_waken
set_location_assignment  -name RX_IN  to  max_fpga_miso
set_location_assignment  -name RX_IN  to  max_fpga_mosi
set_location_assignment  -name RX_IN  to  max_fpga_sck
set_location_assignment  -name RX_IN  to  max_fpga_ssel
set_location_assignment  -name RX_IN  to  pcie_prsnt2_x1
set_location_assignment  -name RX_IN  to  pcie_prsnt2_x4
set_instance_assignment  -name IO_STANDARD LVDS  to  refclk_q12.p
set_global_assignment  -name PARTITION_NETLIST_TYPE  SOURCE  -section_id Top
set_global_assignment  -name PARTITION_FITTER_PRESERVATION_LEVEL  PLACEMENT_AND_ROUTING
set_global_assignment  -name PARTITION_COLOR  16764057  -section_id Top
set_instance_assignment  -name INPUT_TERMINATION "PARALLEL 50 QVM WITH CALIBRATION"  to  ddr3_hps_dq[0]
  -tag  .hps_sdram.p0
<table>
<thead>
<tr>
<th>Instance Assignment</th>
<th>Name</th>
<th>Terminator</th>
<th>Input/Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>ddr3_hps_dq[0]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[1]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[2]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[3]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[4]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[5]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[6]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[7]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[8]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[9]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[10]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[11]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[12]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[13]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[14]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[15]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[16]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION SERIES 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
<tr>
<td>ddr3_hps_dq[17]</td>
<td>tag hps_sdram_p0</td>
<td>TERMINATION PARALLEL 50 Ohm WITH CALIBRATION</td>
<td>-</td>
</tr>
</tbody>
</table>
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[17] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[18] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[18] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[19] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[19] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[20] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[20] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[21] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[21] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[22] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[22] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[23] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[23] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[24] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[24] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[25] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[25] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[26] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[26] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[27] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[27] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[28] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[28] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[29] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[29] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[30] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[30] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[31] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[31] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[32] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[32] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[33] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to
  ddr3_hps_dq[34] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[34] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[35] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[35] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[36] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[36] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[37] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[37] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[38] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[38] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[39] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dq[39] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[0] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[0] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[1] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[1] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[2] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[2] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[3] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[3] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[4] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_p[4] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[0] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[0] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[1] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[1] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[2] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[2] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[3] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[3] -tag _hps_sram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[4] -tag _hps_sram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_hps_dqs_n[4] -tag _hps_sram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_hps_a[0]
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[9] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[10] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[12] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[13] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[14] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[15] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[16] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[17] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[18] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[19] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[20] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[21] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[22] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[23] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[24] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[25] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[26] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[27] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[28] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[29] -tag
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_fpga_dq[30] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_n[0] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_n[1] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_n[2] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_n[3] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_p[0] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_p[1] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_p[2] -tag
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to ddr3_fpga_dqs_p[3] -tag
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to pcie_refclk_n
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_rxd[0]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_rxd[1]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_rxd[2]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_rxd[3]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_txd[3]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_txd[0]
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[25] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[26] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[26] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[27] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[27] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[28] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[28] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[29] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[29] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[30] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[30] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[31] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq[31] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[0] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[0] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[1] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[1] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[2] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[2] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[3] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_p[3] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[0] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[0] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[1] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[1] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[2] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[2] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[3] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITH CALIBRATION" -to ddr3_fpga_dq_n[3] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name NAME D5_DELAY 2 -to ddr3_fpga_clk_p -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITHOUT CALIBRATION" -to ddr3_fpga_clk_p -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 ΩM WITHOUT CALIBRATION" -to ddr3_fpga_clk_n -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name NAME D5_DELAY 2 -to ddr3_fpga_clk_n -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name NAME CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to ddr3_fpga_a[0] -tag __orion_system_mem_if_ddr3_fpga_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[9] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[10] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[12] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[13] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[14] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[15] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[16] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_fpga_dq[17] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[22] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[23] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[24] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[25] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[26] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[27] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[28] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[29] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[30] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[31] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[32] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[33] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[34] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[35] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[36] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[37] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dq[38] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dm[0] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dm[1] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dm[2] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dm[3] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dm[4] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_p[0] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_p[1] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_p[2] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_p[3] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_p[4] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_n[0] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_n[1] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_n[2] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_n[3] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_dqs_n[4] -tag
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[0] -tag
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst| p0|unemphy|uoio_pads|dq_ddio[2]|read_capture_clk_buffer -tag _hps_sdram_p0
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst| p0|unemphy|uread_datapath|reset_n_fifo_write_side[2] -tag _hps_sdram_p0
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst| p0|unemphy|uread_datapath|reset_n_fifo_wraddress[2] -tag _hps_sdram_p0
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst| p0|unemphy|uread_datapath|reset_n_fifo_write_side[3] -tag _hps_sdram_p0
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst| p0|unemphy|uread_datapath|reset_n_fifo_wraddress[3] -tag _hps_sdram_p0
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst| p0|unemphy|uread_datapath|reset_n_fifo_write_side[4] -tag _hps_sdram_p0
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u0|hps_0|hps_io|border|hps_sdram_inst| p0|unemphy|uread_datapath|reset_n_fifo_wraddress[4] -tag _hps_sdram_p0
set_instance_assignment -name PLL_COMPENSATION_MODE DIRECT -to u0|hps_0|hps_io|border|hps_sdram_inst|p110|bout -tag _hps_sdram_p0
set_location_assignment PIN_W16 -to ddr3_fpga_dqs_n[0]
set_location_assignment PIN_W17 -to ddr3_fpga_dqs_n[1]
set_location_assignment PIN_AA18 -to ddr3_fpga_dqs_n[2]
set_location_assignment PIN_AD19 -to ddr3_fpga_dqs_n[3]
set_location_assignment PIN_V16 -to ddr3_fpga_dqs_p[0]
set_location_assignment PIN_V17 -to ddr3_fpga_dqs_p[1]
set_location_assignment PIN_Y17 -to ddr3_fpga_dqs_p[2]
set_location_assignment PIN_AC20 -to ddr3_fpga_dqs_p[3]
set_location_assignment PIN_H3 -to hsma_tx_p[3]
set_location_assignment PIN_K3 -to hsma_tx_p[2]
set_location_assignment PIN_M3 -to hsma_tx_p[1]
set_location_assignment PIN_P3 -to hsma_tx_p[0]
set_location_assignment PIN_H4 -to hsma_tx_n[3]
set_location_assignment PIN_K4 -to hsma_tx_n[2]
set_location_assignment PIN_M4 -to hsma_tx_n[1]
set_location_assignment PIN_P4 -to hsma_tx_n[0]
set_global_assignment -name PHYSICAL_SYNTHESIS_COMBO_LOGIC ON
set_global_assignment -name PHYSICAL_SYNTHESIS_REGISTER_RETIMING ON
set_global_assignment -name PHYSICAL_SYNTHESISASYNCHRONOUS_SIGNAL_PIPELINING ON
set_global_assignment -name PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION ON
set_global_assignment -name PHYSICAL_SYNTHESIS_COMBO_LOGIC_FOR_AREA ON
set_global_assignment -name PHYSICAL_SYNTHESIS_EFFORT EXTRA
set_global_assignment -name OPTIMIZE_HOLD_TIMING "ALL PATHS"
set_global_assignment -name OPTIMIZE_POWER_DURING_FITTING OFF
set_global_assignment -name FITTER_EFFORT "STANDARD FIT"
set_global_assignment -name ROUTER_TIMING_OPTIMIZATION_LEVEL MAXIMUM
set_global_assignment -name ROUTER_CLOCKING_TOPOLOGY_ANALYSIS ON
set_global_assignment -name AUTO_PACKED_REGISTERS STRATIXII NORMAL
set_global_assignment -name ROUTER_CI_CELL_INSERTION_AND_LOGIC_DUPLICATION ON
set_instance_assignment -name PLL.AUTO_RESET_OFF -to "*q_sys_pll_0*|
altera_pll:altera_pll_0*"
set_instance_assignment -name PLL_BANDWIDTH_PRESET AUTO -to "*q_sys_pll_0*|
altera_pll:altera_pll_0*"
set_instance_assignment -name PLL_COMPENSATION_MODE NORMAL -to "*q_sys_pll_0*|
altera_pll:altera_pll_0*"
set_global_assignment -name SEED 100
set_global_assignment -name SDC_FILE timing_base.sdc
set_global_assignment -name VERILOG_FILE orion_top.v
set_global_assignment -name QIP_FILE orion_system/synthesis/orion_system.qip
set_global_assignment -name VERILOG_FILE ip/debounce/debounce.v
set_global_assignment -name QIP_FILE clk_ctrl.qip
set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top

16.3.8 Demo IP Top Verilog File (test_ip_top.v)
Christopher Yarp

This file was created based on the Altera custom_master template. This file supports multiple ports and connects to the DSP logic directly without requiring an export from qsys and attachment in Quartus.

module test_ip_top ( clk, reset,
    // BEGIN WRITE MASTER
    master_write_address, master_write_read, master_write_write, master_write_byteenable, master_write_readdata, master_write_readdatavalid, master_write_writedata, master_write_burstcount, master_write_waitrequest,
    // END WRITE MASTER

    // BEGIN READ MASTER
    master_read_address, master_read_read, master_read_write, master_read_byteenable, master_read_readdata, master_read_readdatavalid, master_read_writedata, master_read_burstcount, master_read_waitrequest,
    // END READ MASTER

    // BEGIN SLAVE
    signals to connect to an Avalon-MM slave interface
    slave_address, slave_read, slave_write, slave_readdata, slave_writedata, slave_byteenable,
    // interrupt signals
    slave_irq
    // END SLAVE
);

    parameter DATA_WIDTH = 32;
    parameter ADDRESS_WIDTH = 32;

    // BEGIN WRITE MASTER PARMS
    parameter MASTER_WRITE_MEMORY_BASED_FIFO = 1; // 0 for LE/ALUT FIFOs, 1 for memory FIFOs (highly recommend 1)
    parameter MASTER_WRITE_FIFO_DEPTH = 32;
    parameter MASTER_WRITE_FIFODEPTH_LOG2 = 5;
    parameter MASTER_WRITE_BURST_CAPABLE = 0; // 1 to enable burst, 0 to disable it
    parameter MASTER_WRITE_MAXIMUM_BURST_COUNT = 2;
    parameter MASTER_WRITE_BURST_COUNT_WIDTH = 2;
    // END WRITE MASTER PARMS
parameter MASTER_READ_MEMORY_BASED_FIFO = 1; // 0 for LE/ALUT FIFOs, 1 for memory FIFOs (highly recommend 1)
parameter MASTER_READ_FIFO_DEPTH = 32;
parameter MASTER_READ_FIFO_DEPTH_LOG2 = 5;
parameter MASTER_READ_BURST_CAPABLE = 0; // 1 to enable burst, 0 to disable it
parameter MASTER_READ_MAXIMUM_BURST_COUNT = 2;
parameter MASTER_READ_BURST_COUNT_WIDTH = 2;
===============END READ MASTER PARMS===============

//=================================
//parameter ENABLE_SYNC_SIGNALS = 0; // only used by the component .tcl file, 1 to expose user_chipselect/write/read, 0 to stub them
parameter IRQ_EN = 0; // 0 = Enable interrupt, 1 = Disable interrupt
//=================================

input clk;
inout reset;

//=================================
//BEGIN WRITE MASTER WIRES=
// control inputs and outputs
wire master_write_control_fixed_location;
// wire [ADDRESS_WIDTH−1:0] master_write_control_read_base; // for read master
// wire [ADDRESS_WIDTH−1:0] master_write_control_read_length; // for read master
wire [ADDRESS_WIDTH−1:0] master_write_control_write_base; // for write master
wire [ADDRESS_WIDTH−1:0] master_write_control_write_length; // for write master
wire master_write_control_go;
//* out/* wire master_write_control_done;
//* out/* wire master_write_control_early_done; // for read master

// user logic inputs and outputs
// wire master_write_user_read_buffer; // for read master
wire master_write_user_write_buffer; // for write master
wire [DATA_WIDTH−1:0] master_write_user_buffer_input_data; // for write master
//* out/* wire [DATA_WIDTH−1:0] master_write_user_buffer_output_data; // for read master
//* out/* wire master_write_user_data_available; // for read master
//* out/* wire master_write_user_buffer_full; // for write master

// master inputs and outputs
output wire [ADDRESS_WIDTH−1:0] master_write_address;
// output wire master_write_read; // for read master
output wire master_write_write; // for write master
output wire [(DATA_WIDTH/8)−1:0] master_write_bynenaable;
// input [DATA_WIDTH−1:0] master_write_readdata; // for read master
// input master_write_readdatavalid; // for read master
output wire [DATA_WIDTH−1:0] master_write_writedata; // for write master
output wire [MASTER_WRITE_BURST_COUNT_WIDTH−1:0] master_write_burstcount; // for bursting read and write masters
input master_write_waitrequest;
//=================================
//END WRITE MASTER WIRES=

//=================================
//BEGIN READ MASTER WIRES=
// control inputs and outputs
wire master_read_control_fixed_location;
wire [ADDRESS_WIDTH−1:0] master_read_control_read_base; // for read master
wire [ADDRESS_WIDTH−1:0] master_read_control_read_length; // for read master
wire [ADDRESS_WIDTH−1:0] master_read_control_write_base; // for write master
wire [ADDRESS_WIDTH−1:0] master_read_control_write_length; // for write master
wire master_read_control_go;
//* out/* wire master_read_control_done;
//* out/* wire master_read_control_early_done; // for read master

// user logic inputs and outputs
wire master_read_user_read_buffer; // for read master
wire master_read_user_write_buffer; // for write master
wire [DATA_WIDTH−1:0] master_read_user_buffer_input_data; // for write master
//* out/* wire [DATA_WIDTH−1:0] master_read_user_buffer_output_data; // for read master
//* out/* wire [DATA_WIDTH−1:0] master_read_user_buffer_output_data; // for read master
//* out/* wire [DATA_WIDTH−1:0] master_read_user_buffer_output_data; // for read master
// master inputs and outputs
output wire [ADDRESS_WIDTH-1:0] master_read_address;
output wire master_read_read;  // for read master
output wire master_read_write;  // for write master
output wire [DATA WIDTH/8-1:0] master_read_bytenable;
input [DATA WIDTH-1:0] master_read_readdata;  // for read master
input master_read_readdatavalid;  // for read master
output wire [DATA WIDTH-1:0] master_read_writedata;  // for write master
output wire [MASTER_READ_BURST_COUNT_WIDTH-1:0] master_read_burstcount;  // for bursting
read and write masters
input master_read_waitrequest;

//==================END READ MASTER WIRES=================

//==================BEGIN SLAVE WIRES====================
input [8:0] slave_address;
input slave_read;
input slave_write;
output wire [DATA WIDTH-1:0] slave_readdata;
input [DATA WIDTH-1:0] slave_writedata;
input [DATA WIDTH/8-1:0] slave_bytenable;
output wire slave_irq;

// user interface
/* output */ wire [DATA WIDTH-1:0] user_dataout_0;
/* output */ wire [DATA WIDTH-1:0] user_dataout_1;
/* output */ wire [DATA WIDTH-1:0] user_dataout_2;
/* output */ wire [DATA WIDTH-1:0] user_dataout_3;
/* output */ wire [DATA WIDTH-1:0] user_dataout_4;
/* output */ wire [DATA WIDTH-1:0] user_dataout_5;
wire [DATA WIDTH-1:0] user_datain_0;
wire [DATA WIDTH-1:0] user_datain_1;
wire [DATA WIDTH-1:0] user_datain_2;
wire [DATA WIDTH-1:0] user_datain_3;
wire [DATA WIDTH-1:0] user_datain_4;
wire [DATA WIDTH-1:0] user_datain_5;
/* output */ wire [DATA WIDTH/8-1:0] user_bytenable;
/* output */ wire user_write;
/* output */ wire user_read;

//===============BEGIN WRITE MASTER MODULE==============
custom_master_write_master(
  .clk(clk),
  .reset(reset),
  // control inputs and outputs
  .control_fixed_location(master_write_control_fixed_location),
  .control_read_base(),
  .control_read_length(),
  .control_write_base(master_write_control_write_base),
  .control_write_length(master_write_control_write_length),
  .control_go(master_write_control_go),
  .control_done(master_write_control_done),
  .control_early_done(),
  // user logic inputs and outputs
  .user_read_buffer(),
  .user_write_buffer(master_write_user_write_buffer),
  .user_buffer_input_data(master_write_user_buffer_input_data),
  .user_buffer_output_data(),
  .user_data_available(),
  .user_buffer_full(master_write_user_buffer_full),
  // master inputs and outputs
.master_address(master_write_address),
.master_read(),
.master_write(master_write_write),
.master_byteenable(master_write_byteenable),
.master_readdata(),
.master_readvalid(),
.master_writedata(master_write_writedata),
.master_burstcount(master_write_burstcount),
.master_waitrequest(master_write_waitrequest);

defparam write_master.MASTER_DIRECTION = 1; // 0 for read master, 1 for
write master
defparam write_master.DATA_WIDTH = DATA_WIDTH;
defparam write_master.MEMORY_BASED_FIFO = MASTER_WRITE_MEMORY_BASED_FIFO; // 0
for LE/ALUT FIFOs, 1 for memory FIFOs (highly recommend 1)
defparam write_master.FIFO_DEPTH = MASTER_WRITE_FIFO_DEPTH;
defparam write_master.FIFO_DEPTH_LOG2 = MASTER_WRITE_FIFO_DEPTH_LOG2;
defparam write_master.ADDRESS_WIDTH = ADDRESS_WIDTH;
defparam write_master.BURST_CAPABLE = MASTER_WRITE_BURST_CAPABLE; // 1 to
enable burst, 0 to disable it
defparam write_master.MAXIMUM_BURST_COUNT = MASTER_WRITE_MAXIMUM_BURST_COUNT;
defparam write_master.BURST_COUNT_WIDTH = MASTER_WRITE_BURST_COUNT_WIDTH;

//==================END WRITE MASTER MODULE==================

//==================BEGIN READ MASTER MODULE=============
custom_master read_master(
.clk(clk),
.reset(reset),

// control inputs and outputs
.control_fixed_location(master_read_control_fixed_location),
.control_read_base(master_read_control_read_base),
.control_read_length(master_read_control_read_length),
.control_write_base(),
.control_write_length(),
.control_go(master_read_control_go),
.control_done(master_read_control_done),
.control_early_done(master_read_control_early_done),

// user logic inputs and outputs
.user_read_buffer(master_read_user_read_buffer),
.user_write_buffer(),
.user_buffer_input_data(),
.user_buffer_output_data(master_read_user_buffer_output_data),
.user_data_available(master_read_user_data_available),
.user_buffer_full(),

// master inputs and outputs
.master_address(master_read_address),
.master_read(master_read_read),
.master_write(),
.master_byteenable(master_read_byteenable),
.master_readdata(master_read_readdata),
.master_readvalid(master_read_readvalid),
.master_writedata(),
.master_burstcount(master_read_burstcount),
.master_waitrequest(master_read_waitrequest)
);

defparam read_master.MASTER_DIRECTION = 0; // 0 for read master, 1 for
write master
defparam read_master.DATA_WIDTH = DATA_WIDTH;
defparam read_master.MEMORY_BASED_FIFO = MASTER_READ_MEMORY_BASED_FIFO; // 0
for LE/ALUT FIFOs, 1 for memory FIFOs (highly recommend 1)
defparam read_master.FIFO_DEPTH = MASTER_READ_FIFO_DEPTH;
defparam read_master.FIFO_DEPTH_LOG2 = MASTER_READ_FIFO_DEPTH_LOG2;
defparam read_master.ADDRESS_WIDTH = ADDRESS_WIDTH;
defparam read_master.BURST_CAPABLE = MASTER_READ_BURST_CAPABLE; // 1 to enable burst, 0 to disable it
defparam read_master.MAXIMUM_BURST_COUNT = MASTER_READ_MAXIMUM_BURST_COUNT;
defparam read_master.BURST_COUNT_WIDTH = MASTER_READ_BURST_COUNT_WIDTH;
//==========================================BEGIN READ MASTER MODULE==========================================
//enable burst, 0 to disable it

defparam read_master.MAXIMUM_BURST_COUNT = MASTER_READ_MAXIMUM_BURST_COUNT;
defparam read_master.BURST_COUNT_WIDTH = MASTER_READ_BURST_COUNT_WIDTH;
//==========================================END READ MASTER MODULE==========================================

//signals to connect to an Avalon clock source interface
slave_template csr(
  .clk(clk),
  .reset(reset),

  //signals to connect to an Avalon-MM slave interface
  .slave_address(slave_address),
  .slave_read(slave_read),
  .slave_write(slave_write),
  .slave_readdata(slave_readdata),
  .slave_writedata(slave_writedata),
  .slave_byteenable(slave_byteenable),

  //interrupt signals
  .slave_irq(slave_irq),

  //signals to connect to custom user logic (up to 16 input and output pairs)
  .user_dataout_0(user_dataout_0), //address
  .user_dataout_1(user_dataout_1), //mode
  .user_dataout_2(user_dataout_2), //data0
  .user_dataout_3(user_dataout_3), //data1
  .user_dataout_4(user_dataout_4), //data2
  .user_dataout_5(user_dataout_5), //data3
  .user_dataout_6(),
  .user_dataout_7(),
  .user_dataout_8(),
  .user_dataout_9(),
  .user_dataout_10(),
  .user_dataout_11(),
  .user_dataout_12(),
  .user_dataout_13(),
  .user_dataout_14(),
  .user_dataout_15(),
  .user_datain_0(user_datain_0), //readReady
  .user_datain_1(user_datain_1), //writeReady
  .user_datain_2(user_datain_2), //data0
  .user_datain_3(user_datain_3), //data1
  .user_datain_4(user_datain_4), //data2
  .user_datain_5(user_datain_5), //data3
  .user_datain_6(),
  .user_datain_7(),
  .user_datain_8(),
  .user_datain_9(),
  .user_datain_10(),
  .user_datain_11(),
  .user_datain_12(),
  .user_datain_13(),
  .user_datain_14(),
  .user_datain_15(),

  //optional signals so that your external logic knows what location is being accessed
  .user_chipselect(user_chipselect),
  .user_byteenable(user_byteenable),
  .user_write(user_write),
  .user_read(user_read)
);
defparam csr.DATA_WIDTH = DATA_WIDTH; //word size of each input and output register
defparam csr.ENABLE_SYNC_SIGNALS = ENABLE_SYNC_SIGNALS; //only used by the component .tcl file. 1 to expose user_chipselect/write/read, 0 to stub them
### Base Module Logic I/O

// inputs
.addr(user_datain_0),
.writeData0(user_dataout_2),
.writeData1(user_dataout_3),
.writeData2(user_dataout_4),
.writeData3(user_dataout_5),
.mode(user_dataout_1), // 0 for read, 1 for write
.modeAccessed(user_chipselect[1] & user_read),

// outputs
.readData0(user_datain_2),
.readData1(user_datain_3),
.readData2(user_datain_4),
.readData3(user_datain_5),
.writeReady(user_datain_1[0]),
.readReady(user_datain_0[0]),

### Memory Logic

// write logic
.master_write_control_fixed_location(master_write_control_fixed_location),
.master_write_control_write_base(master_write_control_write_base),
Listing 16.7: Demo IP Top Verilog File (test_ip_top.v)

16.3.9 Demo IP Top Verilog File (test_ip.v)

```verilog
module test_ip(
    clk,
    reset,

    // *** Base Module Logic I/O ***
    // inputs
    address,
    writeData0,
    writeData1,
    writeData2,
    writeData3,
    mode, // 0 for read, 1 for write
    modeAccessed,

    // outputs
    readData0,
    readData1,
    readData2,
    readData3,
    // writeReady, readReady,

    // *** Memory Logic ***
    // write logic
    master_write_control_write_length,
    master_write_control_write_base,
    master_write_control_write_length,
    master_write_user_write_buffer_input_data,
    master_write_user_write_buffer_full,
    master_write_user_control_go,
    master_write_user_control_done,

    // read logic
    master_read_control_fixed_location,
    master_read_control_read_base,
    master_read_control_read_length,
    master_read_user_read_buffer,
    master_read_user_read_buffer_output_data,
    master_read_user_data_available,
    master_read_user_control_go,
    master_read_user_control_done)
);

//==================
// END TEST IP MODULE===================

endmodule
```
master_read_control_read_length,
master_read_user_read_buffer,
master_read_user_buffer_output_data,
master_read_user_data_available,
master_read_user_control_go,
master_read_user_control_done
);

input clk;
input reset;
input address;
input writeData0;
input writeData1;
input writeData2;
input writeData3;
input mode;
input modeAccessed;

// outputs
output reg [31:0] readData0;
output reg [31:0] readData1;
output reg [31:0] readData2;
output reg [31:0] readData3;
output reg writeReady;
output reg readReady;

// avalon connections
output reg master_write_control_fixed_location;
output reg [31:0] master_write_control_write_base;
output reg [31:0] master_write_control_write_length;
output reg master_write_user_write_buffer;
output reg [31:0] master_write_user_buffer_input_data;
input master_write_user_buffer_full;
output reg master_write_user_control_go;
input master_write_user_control_done;

output reg master_read_control_fixed_location;
output reg [31:0] master_read_control_read_base;
output reg [31:0] master_read_control_read_length;
output reg master_read_user_read_buffer;
input [31:0] master_read_user_buffer_output_data;
input master_read_user_data_available;
output reg master_read_user_control_go;
input master_read_user_control_done;

reg [1:0] writeState;
// 00 ready
// 01 writingToBuffer
// 11 writeCommand
// 10 waitingForFinish
reg [1:0] writeByteNum;

reg [1:0] readState;
// 00 ready/sendCommand
// 01 readingFromBuffer
// 11 waitingForFinish
// 10 UNUSED
reg [1:0] readByteNum;

reg [31:0] writeAddr;
reg [31:0] writeData0r;
reg [31:0] writeData1r;
reg [31:0] writeData2r;
reg [31:0] writeData3r;

initial begin
readData0 <= 0;
readData1 <= 0;
readData2 <= 0;
readData3 <= 0;
writeReady <= 1;
readReady <= 1;
writeState <= 0;
writeByteNum <= 0;
readState <= 0;
readByteNum <= 0;
writeAddr <= 0;
writeData0r <= 0;
writeData1r <= 0;
writeData2r <= 0;
writeData3r <= 0;
master_write_user_buffer_input_data <= 0;
master_write_control_fixed_location <= 0;
master_write_control_write_base <= 0;
master_write_control_write_length <= 0;
master_write_user_write_buffer <= 0;
master_write_user_control_go <= 0;
master_read_control_fixed_location <= 0;
master_read_control_read_base <= 0;
master_read_control_read_length <= 0;
master_read_user_read_buffer <= 0;
master_read_user_control_go <= 0;
end

always @(posedge clk or posedge reset)
begin
if (reset)
begin
writeReady <= 1;
writeState <= 0;
writeByteNum <= 0;
writeAddr <= 0;
writeData0r <= 0;
writeData1r <= 0;
writeData2r <= 0;
writeData3r <= 0;
master_write_user_buffer_input_data <= 0;
master_write_control_fixed_location <= 0;
master_write_control_write_base <= 0;
master_write_control_write_length <= 0;
master_write_user_write_buffer <= 0;
master_write_user_control_go <= 0;
end
else begin
    case (writeState)
        2'b00: // ready
end
begin
  if (modeAccessed & & mode == 1)
  begin
    // begin the process of writing
    writeReady <= 0; // no longer ready to accept new write command
    writeAddr <= address;
    writeData0r <= writeData0;
    writeData1r <= writeData1;
    writeData2r <= writeData2;
    writeData3r <= writeData3;
    writeState <= 2'b01;
    writeByteNum <= 2'b00;
  end
end

2'b01:
begin
  // now writing bytes into fifo buffer
  case (writeByteNum)
  2'b00:
    begin
      master_write_user_buffer_input_data <= writeData0r;
      writeByteNum <= 2'b01;
    end
  2'b01:
    begin
      master_write_user_buffer_input_data <= writeData1r;
      writeByteNum <= 2'b11;
    end
  2'b11:
    begin
      master_write_user_buffer_input_data <= writeData2r;
      writeByteNum <= 2'b10;
    end
  2'b10:
    begin
      master_write_user_buffer_input_data <= writeData3r;
      writeByteNum <= 2'b00; // return to 0
      writeState <= 2'b11; // move to next phase
    end
  endcase
end

2'b11:
begin
  // send command
  master_write_control_write_base <= writeAddr;
  master_write_control_write_length <= 32'd16;
  master_write_user_control_go <= 1;
  writeState <= 2'b10;
end

2'b10:
begin
  // send command
  master_write_user_control_go <= 0;
  if (master_write_user_control_done == 1)
  begin
    writeState <= 2'b00; // done
    writeReady <= 1;
  end
end
endcase
end
end
/read function
always @(posedge clk)
begin
if (reset)
begin
readData0 <= 0;
readData1 <= 0;
readData2 <= 0;
readData3 <= 0;
readReady <= 1;
readState <= 0;
readByteNum <= 0;
master_read_control_fixed_location <= 0;
master_read_control_read_base <= 0;
master_read_control_read_length <= 0;
master_read_user_read_buffer <= 0;
master_read_user_control_go <= 0;
end
else
begin
case (readState)
2'b00: // ready
begin
master_read_user_read_buffer <= 0;
if (modeAccessed && mode == 0)
begin
// send read command
readReady <= 0; // no longer ready to accept new write command
master_read_control_read_base <= address;
master_read_control_read_length <= 32'd16;
master_read_user_control_go <= 1;
readState <= 2'b01;
readByteNum <= 2'b00;
end
end
end
2'b01:
begin
master_read_user_control_go <= 0;
// now writing bytes into fifo buffer
if (master_read_user_data_available == 1)
begin
case (readByteNum)
2'b00:
begin
readData0 <= master_read_user_buffer_output_data;
master_read_user_read_buffer <= 1;
readByteNum <= 2'b01;
end
2'b01:
begin
readData1 <= master_read_user_buffer_output_data;
master_read_user_read_buffer <= 1;
readByteNum <= 2'b11;
end
2'b11:
begin
readData2 <= master_read_user_buffer_output_data;
master_read_user_read_buffer <= 1;
end
end
end
end
end
end
Listing 16.8: Demo IP Verilog File (test_ip.v)

16.3.10 Demo IP_hw.tcl File (IPTest_hw.tcl)

# TCL File Generated by Component Editor 8.0
# Sat May 31 20:41:17 PDT 2008
# DO NOT MODIFY

# Based on the TCL Files for the Avalon Master Template and the Avalon Slave Template by Altera Corp.

# ==============================================================
module burst_read_master
#
set_module_property DESCRIPTION "ARM Mapped IP Test"
set_module_property NAME IPTest
set_module_property VERSION 1.0
set_module_property GROUP "Test IP"
set_module_property AUTHOR cyarp
set_module_property ICON_PATH SCU_Logo_Seperate_Small.gif
set_module_property DISPLAY_NAME "Test IP"
set_module_property TOP_LEVEL_HDL_FILE test_ip_top.v
set_module_property TOPLEVEL_HDL_MODULE test_ip_top
#set_module_property INSTANTIATE_IN SYSTEM_MODULE true
set_module_property EDITABLE false
set_module_property SIMULATIONMODEL_IN_VERILOG false
set_module_property SIMULATIONMODEL_IN_VHDL false
set_module_property SIMULATIONMODEL_HAS_TULIPS false
set_module_property SIMULATIONMODEL_IS_OBFUSCATED false

module_assignment embeddedsw.dts.compatible "simple-bus"

module_property ELABORATION_CALLBACK elaborate_me
module_property VALIDATION_CALLBACK validate_me

# | files
add_file test_ip_top.v {SYNTHESIS SIMULATION}
add_file test_ip.v {SYNTHESIS SIMULATION}
add_file custom_master.v {SYNTHESIS SIMULATION}
add_file burst_write_master.v {SYNTHESIS SIMULATION}
add_file burst_read_master.v {SYNTHESIS SIMULATION}
add_file master_write_master.v {SYNTHESIS SIMULATION}
add_file latency_aware_read_master.v {SYNTHESIS SIMULATION}
add_file slave_template.v {SYNTHESIS SIMULATION}
add_file interrupt_logic.v {SYNTHESIS SIMULATION}
add_file slave_template_macros.h {SYNTHESIS SIMULATION}

# ===========BEGIN GENERAL PARMS====================
# Avalon Master Settings
add_parameter DATA_WIDTH Integer 32 "Width of the data path"
set_parameter_property DATA_WIDTH VISIBLE false
set_parameter_property DATA_WIDTH DISPLAY_NAME "Data Width"
set_parameter_property DATA_WIDTH GROUP "Avalon-MM Master Properties"
set_parameter_property DATA_WIDTH AFFECTS_PORT_WIDTHS true
set_parameter_property DATA_WIDTH ALLOWED_RANGES [8 16 32 64 128 256 512 1024]

add_parameter ADDRESS_WIDTH Integer 32 "Address Width"
set_parameter_property ADDRESS_WIDTH VISIBLE false
set_parameter_property ADDRESS_WIDTH DISPLAY_NAME "Address Width"
set_parameter_property ADDRESS_WIDTH GROUP "Avalon-MM Master Properties"
set_parameter_property ADDRESS_WIDTH AFFECTS_PORT_WIDTHS true

# ===========END GENERAL PARMS====================
# ===========BEGIN WRITE MASTER PARMS=============
# Burst Settings
add_parameter MASTER_WRITE_BURST_CAPABLE Integer 0 "Enable bursting"
set_parameter_property MASTER_WRITE_BURST_CAPABLE VISIBLE true
set_parameter_property MASTER_WRITE_BURST_CAPABLE DISPLAY_NAME "Burst Capable"
set_parameter_property MASTER_WRITE_BURST_CAPABLE GROUP "Write Master – Burst Properties"
set_parameter_property MASTER_WRITE_BURST_CAPABLE AFFECTS_PORT_WIDTHS true
set_parameter_property MASTER_WRITE_BURST_CAPABLE ALLOWED_RANGES ["0: Disabled" "1: Enabled"]

add_parameter MASTER_WRITE_MAXIMUM_BURST_COUNT Integer 2 "Maximum Burst Count"
set_parameter_property MASTER_WRITE_MAXIMUM_BURST_COUNT VISIBLE true
set_parameter_property MASTER_WRITE_MAXIMUM_BURST_COUNT DISPLAY_NAME "Maximum Burst Count"
set_parameter_property MASTER_WRITE_MAXIMUM_BURST_COUNT GROUP "Write Master – Burst Properties"
set_parameter_property MASTER_WRITE_MAXIMUM_BURST_COUNT AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_WRITE_MAXIMUM_BURST_COUNT ALLOWED_RANGES [1 2 4 8 16 32 64 128]
# Other Settings

```
add_parameter MASTER_WRITE_BURST_COUNT_WIDTH Integer "2" "Enable bursting"
set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH VISIBLE false
set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH DISPLAY_NAME "Burst Count Width"
set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH GROUP "Write Master – Burst Properties"
set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH AFFECTS_PORT_WIDTHS true
set_parameter_property MASTER_WRITE_BURST_COUNT_WIDTH ALLOWED_RANGES {1:8}
```

# Burst Settings

```
add_parameter MASTER_READ_BURST_CAPABLE Integer 0 "Enable bursting"
set_parameter_property MASTER_READ_BURST_CAPABLE VISIBLE true
set_parameter_property MASTER_READ_BURST_CAPABLE DISPLAY_NAME "Burst Capable"
set_parameter_property MASTER_READ_BURST_CAPABLE GROUP "Read Master – Burst Properties"
set_parameter_property MASTER_READ_BURST_CAPABLE AFFECTS_PORT_WIDTHS true
set_parameter_property MASTER_READ_BURST_CAPABLE ALLOWED_RANGES {"0: Disabled" "1: Enabled"}
add_parameter MASTER_READ_MAXIMUM_BURST_COUNT Integer "2" "Maximum Burst Count"
set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT VISIBLE true
set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT DISPLAY_NAME "Maximum Burst Count"
set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT GROUP "Read Master – Burst Properties"
set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT ALLOWED_RANGES {1 2 4 8 16 32 64 128}
add_parameter MASTER_READ_BURST_COUNT_WIDTH Integer "2" "Enable bursting"
set_parameter_property MASTER_READ_BURST_COUNT_WIDTH VISIBLE false
set_parameter_property MASTER_READ_BURST_COUNT_WIDTH DISPLAY_NAME "Burst Count Width"
set_parameter_property MASTER_READ_BURST_COUNT_WIDTH GROUP "Read Master – Burst Properties"
set_parameter_property MASTER_READ_BURST_COUNT_WIDTH AFFECTS_PORT_WIDTHS true
set_parameter_property MASTER_READ_BURST_COUNT_WIDTH ALLOWED_RANGES {1:8}
# Other Settings

```
add_parameter MASTER_READ_FIFO_DEPTH Integer "32" "FIFO depth"
set_parameter_property MASTER_READ_FIFO_DEPTH VISIBLE true
set_parameter_property MASTER_READ_FIFO_DEPTH DISPLAY_NAME "FIFO Depth"
set_parameter_property MASTER_READ_FIFO_DEPTH GROUP "Read Master – Other Properties"
```

# FIFO Settings

```
add_parameter MASTER_WRITE_FIFO_DEPTH Integer "32" "FIFO depth"
set_parameter_property MASTER_WRITE_FIFO_DEPTH VISIBLE true
set_parameter_property MASTER_WRITE_FIFO_DEPTH DISPLAY_NAME "FIFO Depth"
set_parameter_property MASTER_WRITE_FIFO_DEPTH GROUP "Write Master – Other Properties"
set_parameter_property MASTER_WRITE_FIFO_DEPTH AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_WRITE_FIFO_DEPTH ALLOWED_RANGES {4 8 16 32 64 128 256}
add_parameter MASTER_WRITE_FIFO_DEPTH_LOG2 Integer "5" "log2(FIFO Depth)"
set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 VISIBLE false
set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 DISPLAY_NAME "log2(FIFO Depth)"
set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 GROUP "Write Master – Other Properties"
set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_WRITE_FIFO_DEPTH_LOG2 ALLOWED_RANGES {2:8}
add_parameter MASTER_WRITE_MEMORY_BASED_FIFO Integer 1 "Select false if you want register based (0) FIFO instead of memory (1)"
set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO VISIBLE true
set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO DISPLAY_NAME "Memory based FIFO"
set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO GROUP "Write Master – Other Properties"
set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_WRITE_MEMORY_BASED_FIFO ALLOWED_RANGES {"1:Memory" "0:Logic"}
```
set_parameter_property MASTER_READ_FIFODEPTH AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_READ_FIFODEPTH ALLOWED_RANGES [4 8 16 32 64 128 256]
add_parameter MASTER_READ_FIFODEPTH LOG2 Integer 5 "log2 (FIFO Depth)"
set_parameter_property MASTER_READ_FIFODEPTH LOG2 DISPLAY_NAME "log2 (FIFO Depth)"
set_parameter_property MASTER_READ_FIFODEPTH LOG2 GROUP "Read Master − Other Properties"
set_parameter_property MASTER_READ_FIFODEPTH AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_READ_FIFODEPTH ALLOWED_RANGES [2:8]
add_parameter MASTER_READ_MEMORY_BASED_FIFO Integer 1 "Select false if you want register based (0) FIFO instead of memory (1)"
set_parameter_property MASTER_READ_MEMORY_BASED_FIFO VISIBLE true
set_parameter_property MASTER_READ_MEMORY_BASED_FIFO DISPLAY_NAME "Memory based FIFO"
set_parameter_property MASTER_READ_MEMORY_BASED_FIFO GROUP "Read Master − Other Properties"
set_parameter_property MASTER_READ_MEMORY_BASED_FIFO AFFECTS_PORT_WIDTHS false
set_parameter_property MASTER_READ_MEMORY_BASED_FIFO ALLOWED_RANGES [1:Memory "0:Logic"]
# ===========BEGIN SLAVE PARMS=================
# | connection point clk_in
# |
add_interface clk_in clock end
set_interface_property clk_in ptfSchematicName ""
add_interface_port clk_in clk clk Input 1
add_interface_port clk_in reset reset Input 1
# |
# | connection point master_write
# |
add_interface master_write avalon start
set_interface_property master_write adaptsTo "" master_write doStreamReads false
set_interface_property master_write doStreamWrites false
set_interface_property master_write burstOnBurstBoundariesOnly false
set_interface_property master_write ASSOCIATED_CLOCK clk_in
add_interface_port master_write master_write_address address Output -1
add_interface_port master_write master_write_read read read Output 1
add_interface_port master_write master_write_write write Output 1
add_interface_port master_write master_write_byteenable byteenable Output -1
add_interface_port master_write master_write_readdata readdata Input -1
add_interface_port master_write master_write_readdatavalid readdatavalid Input 1
add_interface_port master_write master_write_writedata writedata Output -1
add_interface_port master_write master_write_burstcount burstcount Output -1
add_interface_port master_write master_write_waitrequest waitrequest Input 1
# |
# | connection point master_read

174
# | | | add_interface master_read avalon start
# | | set_interface_property master_read linewrapBursts false
# | | set_interface_property master_read adaptsTo ""
# | | set_interface_property master_read doStreamReads false
# | | set_interface_property master_read burstOnBurstBoundariesOnly false

# | | set_interface_property master_read ASSOCIATED_CLOCK clk_in

# | | add_interface_port master_read master_read_address address Output -1
# | | add_interface_port master_read master_read_read read Output 1
# | | add_interface_port master_read master_read_write write Output 1
# | | add_interface_port master_read master_read_byteenable byteenable Input -1
# | | add_interface_port master_read master_read_readdata readdata Input -1
# | | add_interface_port master_read master_read_readdatavalid readdatavalid Input 1
# | | add_interface_port master_read master_read_writedata writedata Output -1
# | | add_interface_port master_read master_read_burstcount burstcount Output -1
# | | add_interface_port master_read master_read_waitrequest waitrequest Input 1

# | | add_interface csr avalon end

# | | set_interface_property csr holdTime 0
# | | set_interface_property csr linewrapBursts false
# | | set_interface_property csr minimumUninterruptedRunLength 1
# | | set_interface_property csr bridgesToMaster ""
# | | set_interface_property csr isMemoryDevice false
# | | set_interface_property csr burstOnBurstBoundariesOnly false
# | | set_interface_property csr addressSpan 5 1 2
# | | set_interface_property csr timingUnits Cycles
# | | set_interface_property csr setupTime 0
# | | set_interface_property csr writeWaitTime 0
# | | set_interface_property csr isNonVolatileStorage false
# | | set_interface_property csr addressAlignment DYNAMIC
# | | set_interface_property csr maximumPendingReadTransactions 0
# | | set_interface_property csr readWaitTime 0
# | | set_interface_property csr readLatency 3
# | | set_interface_property csr printableDevice false

# | | set_interface_property csr ASSOCIATED_CLOCK clk_in

# | | add_interface_port csr slave_address address Input 9
# | | add_interface_port csr slave_read read Input 1
# | | add_interface_port csr slave_write write Input 1
# | | add_interface_port csr slave_readdata readdata Input 1
# | | add_interface_port csr slave_writedata writedata Input -1

# | | add_interface_port csr slave_irq irq Output 1

# | | proc elaborate_me {} {
# | | # set all the new port widths - this is all based
# | | set the_data_width [get_parameter_value DATA_WIDTH]
set the_address_width [get_parameter_value ADDRESS_WIDTH]
set the_byteenable_width [expr {\$the_data_width / 8}]

# BEGIN MASTER WRITE ELABORATE
set master_write_the_burst_count_width [get_parameter_value MASTER_WRITE_BURST_COUNT_WIDTH]
set_port_property master_write_address WIDTH $the_address_width
set_port_property master_write_byteenable WIDTH $the_byteenable_width
set_port_property master_write_readdata WIDTH $the_data_width
set_port_property master_write_writedata WIDTH $the_data_width
set_port_property master_write_burstcount WIDTH $master_write_the_burst_count_width

# determine the master direction and burst capabilities
# write master
set master_write_the_burst_possible [get_parameter_value MASTER_WRITE_BURST_CAPABLE]

# switch between read and write master signals (excluding burstcount)
set_port_property master_write_read TERMINATION true
set_port_property master_write_write TERMINATION true
set_port_property master_write_readdata TERMINATION true
set_port_property master_write_readdatavalid TERMINATION true
set_port_property master_write_writedata TERMINATION false

# enable/disable the burstcount signal
if { $master_write_the_burst_possible == 0 } {
  set_port_property master_write_burstcount TERMINATION true
} else {
  set_port_property master_write_burstcount TERMINATION false
}
# END MASTER WRITE ELABORATE

# BEGIN MASTER READ ELABORATE
set master_read_the_burst_count_width [get_parameter_value MASTER_READ_BURST_COUNT_WIDTH]
set_port_property master_read_address WIDTH $the_address_width
set_port_property master_read_byteenable WIDTH $the_byteenable_width
set_port_property master_read_readdata WIDTH $the_data_width
set_port_property master_read_writedata WIDTH $the_data_width
set_port_property master_read_burstcount WIDTH $master_read_the_burst_count_width

# determine the master direction and burst capabilities
# write master
set master_read_the_burst_possible [get_parameter_value MASTER_READ_BURST_CAPABLE]

# switch between read and write master signals (excluding burstcount)
set_port_property master_read_read TERMINATION false
set_port_property master_read_write TERMINATION true
set_port_property master_read_readdata TERMINATION false
set_port_property master_read_readdatavalid TERMINATION false
set_port_property master_read_writedata TERMINATION true

# enable/disable the burstcount signal
if { $master_read_the_burst_possible == 0 } {
  set_port_property master_read_burstcount TERMINATION true
} else {
  set_port_property master_read_burstcount TERMINATION false
}
# END MASTER READ ELABORATE

# BEGIN SLAVE ELABORATE
set_port_property slave_readdata WIDTH $the_data_width
set_port_property slave_writedata WIDTH $the_data_width

---
set the_irq_en [get_parameter_value IRQ_EN]

## adding the slave_byteenable and user_byteenable signals only if the data width is
## greater than 8 bits
if { $the_data_width != 8 } {
    add_interface_port csr slave_byteenable byteenable Input [expr { $the_data_width / 8 }]
    #if [ $the_enable_sync_signals == 1 ] {
    # add_interface_port user_interface user_byteenable export Output [expr { $the_data_width / 8 }]
    #}
    #
    ## terminate interrupt signal if it is not turned on
expr { ($the_irq_en == 1) ? [ set_port_property slave_irq TERMINATION false ] : [ set_port_property slave_irq TERMINATION true ]};

# ===============END SLAVE ELABORATE================

proc validate_me {} {

    # ===========BEGIN MASTER WRITE VALIDATE===============
    # read in all the parameter that matter for validation
    set master_write_the_burst_capable [get_parameter_value MASTER_WRITE_BURST_CAPABLE]
    set master_write_the_maximum_burst_count [get_parameter_value MASTER_WRITE_MAXIMUM_BURST_COUNT]
    set master_write_the_fifo_depth [get_parameter_value MASTER_WRITE_FIFO_DEPTH]
    # when burst is enabled check to make sure FIFO depth is at least twice as large (also
    # enable/disable burst count)
    if { $master_write_the_burst_capable == 1 } {
        set_parameter_property MASTER_WRITE_MAXIMUM_BURST_COUNT_ENABLED true
        if { $master_write_the_fifo_depth < [expr { $master_write_the_maximum_burst_count * 2 }] } {
            send_message Error "The FIFO Depth must be at least twice as large as Maximum Burst Count." 
        }
    } else {
        set_parameter_property MASTER_WRITE_MAXIMUM_BURST_COUNT_ENABLED false
    }

    set master_write_the_burst_count [get_parameter_value MASTER_WRITE_MAXIMUM_BURST_COUNT]
    set master_write_the_burst_count_width [expr { (log($master_write_the_burst_count) / log(2)) + 1 }]
    set master_write_the_fifo_depth [get_parameter_value MASTER_WRITE_FIFO_DEPTH]
    set master_write_the_fifo_depth_log2 [expr { log($master_write_the_fifo_depth) / log(2) }]
    set_parameter_value MASTER_WRITE_BURST_COUNT_WIDTH $master_write_the_burst_count_width
    set_parameter_value MASTER_WRITE_FIFODEPTH_LOG2 $master_write_the_fifo_depth_log2
    # ===============END MASTER WRITE VALIDATE===============

    # ===========BEGIN MASTER READ VALIDATE===============
    # read in all the parameter that matter for validation
    set master_read_the_burst_capable [get_parameter_value MASTER_READ_BURST_CAPABLE]
    set master_read_the_maximum_burst_count [get_parameter_value MASTER_READ_MAXIMUM_BURST_COUNT]
    set master_read_the_fifo_depth [get_parameter_value MASTER_READ_FIFO_DEPTH]
    # when burst is enabled check to make sure FIFO depth is at least twice as large (also
    # enable/disable burst count)
    if { $master_read_the_burst_capable == 1 } {
        set_parameter_property MASTER_READ_MAXIMUM_BURST_COUNT_ENABLED true
        if { $master_read_the_fifo_depth < [expr { $master_read_the_maximum_burst_count * 2 }] } {
            send_message Error "The FIFO Depth must be at least twice as large as Maximum Burst Count." 
        }
    }
Listing 16.9: Demo IP_hw.tcl File (IPTest_hw.tcl)
parameter MEMORY_BASED_FIFO = 1; // 0 for LE/ALUT FIFOs, 1 for memory FIFOs (highly recommend 1)
parameter FIFODEPTH = 32;
parameter FIFODEPTH_LOG2 = 5;
parameter ADDRESS_WIDTH = 32;
parameter BURST_CAPABLE = 0; // 1 to enable burst, 0 to disable it
parameter MAXIMUM_BURST_COUNT = 2;
parameter BURST_COUNT_WIDTH = 2;

input clk;
input reset;

// control inputs and outputs
input control_fixed_location;
input [ADDRESS_WIDTH-1:0] control_read_base; // for read master
input [ADDRESS_WIDTH-1:0] control_read_length; // for read master
input [ADDRESS_WIDTH-1:0] control_write_base; // for write master
input [ADDRESS_WIDTH-1:0] control_write_length; // for write master
input control_go;
output wire control_done;
output wire control_early_done; // for read master

// user logic inputs and outputs
input user_read_buffer; // for read master
input user_write_buffer; // for write master
input [DATA_WIDTH-1:0] user_buffer_input_data; // for write master
output wire [DATA_WIDTH-1:0] user_buffer_output_data; // for read master
output wire user_data_available; // for read master
output wire user_buffer_full; // for write master

// master inputs and outputs
output wire [ADDRESS_WIDTH-1:0] master_address;
output wire master_read; // for read master
output wire master_write; // for write master
output wire [(DATA_WIDTH/8)-1:0] master_byteenable;
input [DATA_WIDTH-1:0] master_readdata; // for read master
input master_readdatavalid; // for read master
output wire [DATA_WIDTH-1:0] master_writedata; // for write master
output wire [BURST_COUNT_WIDTH-1:0] master_burstcount; // for bursting read and write masters
input master_waitrequest;

generate // big generate if statement to select the appropriate master depending on the
direction and burst parameters
if (MASTER_DIRECTION == 0)
begin
  if (BURST_CAPABLE == 1)
begin
burst_read_master a_burst_read_master(
  .clk (clk),
  .reset (reset),
  .control_fixed_location (control_fixed_location),
  .control_read_base (control_read_base),
  .control_read_length (control_read_length),
  .control_go (control_go),
  .control_done (control_done),
  .control_early_done (control_early_done),
  .user_read_buffer (user_read_buffer),
  .user_buffer_data (user_buffer_output_data),
  .user_data_available (user_data_available),
  .master_address (master_address),
  .master_read (master_read),
  .master_byteenable (master_byteenable),
  .master_readdata (master_readdata),
  .master_readdatavalid (master_readdatavalid),
  .master_burstcount (master_burstcount),
end
.master_waitrequest ( master_waitrequest )
);
defparam a_burst_read_master .DATAWIDTH = DATA_WIDTH;
defparam a_burst_read_master .MAXBURSTCOUNT = MAXIMUM_BURST_COUNT;
defparam a_burst_read_master .BURSTCOUNTWIDTH = BURST_COUNT_WIDTH;
defparam a_burst_read_master .BYTEENABLEWIDTH = DATA_WIDTH/8;
defparam a_burst_read_master .ADDRESSWIDTH = ADDRESS_WIDTH;
defparam a_burst_read_master .FIFODEPTH = FIFO_DEPTH;
defparam a_burst_read_master .FIFODEPTH_LOG2 = FIFO_DEPTH_LOG2;
defparam a_burst_read_master .FIFOUSEMEMORY = MEMORY_BASED_FIFO;
end
else
begin
  latency_aware_read_master a_latency_aware_read_master (  
    .clk ( clk ),  
    .reset ( reset ),  
    .control_fixed_location ( control_fixed_location ),  
    .control_read_base ( control_read_base ),  
    .control_read_length ( control_read_length ),  
    .control_go ( control_go ),  
    .control_done ( control_done ),  
    .control_early_done ( control_early_done ),  
    .user_read_buffer ( user_read_buffer ),  
    .user_buffer_data ( user_buffer_output_data ),  
    .user_data_available ( user_data_available ),  
    .master_address ( master_address )
    .master_read ( master_read ),  
    .master_byteenable ( master_byteenable ),  
    .master_readdata ( master_readdata ),  
    .master_readdatavalid ( master_readdatavalid ),  
    .master_waitrequest ( master_waitrequest )
  );
defparam a_latency_aware_read_master .DATAWIDTH = DATA_WIDTH;
defparam a_latency_aware_read_master .BYTEENABLEWIDTH = DATA_WIDTH/8;
defparam a_latency_aware_read_master .ADDRESSWIDTH = ADDRESS_WIDTH;
defparam a_latency_aware_read_master .FIFODEPTH = FIFO_DEPTH;
defparam a_latency_aware_read_master .FIFODEPTH_LOG2 = FIFO_DEPTH_LOG2;
defparam a_latency_aware_read_master .FIFOUSEMEMORY = MEMORY_BASED_FIFO;
end
end
else
begin
  if ( BURST_CAPABLE == 1 )
  begin
    burst_write_master a_burst_write_master (  
      .clk ( clk ),  
      .reset ( reset ),  
      .control_fixed_location ( control_fixed_location ),  
      .control_write_base ( control_write_base ),  
      .control_write_length ( control_write_length ),  
      .control_go ( control_go ),  
      .control_done ( control_done ),  
      .user_write_buffer ( user_write_buffer ),  
      .user_buffer_data ( user_buffer_input_data ),  
      .user_buffer_full ( user_buffer_full ),  
      .master_address ( master_address ),  
      .master_write ( master_write ),  
      .master_byteenable ( master_byteenable ),  
      .master_writedata ( master_writedata ),  
      .master_burstcount ( master_burstcount ),  
      .master_waitrequest ( master_waitrequest )
    );
defparam a_burst_write_master .DATAWIDTH = DATA_WIDTH;
defparam a_burst_write_master .MAXBURSTCOUNT = MAXIMUM_BURST_COUNT;
defparam a_burst_write_master .BURSTCOUNTWIDTH = BURST_COUNT_WIDTH;
defparam a_burst_write_master .BYTEENABLEWIDTH = DATA_WIDTH/8;
defparam a_burst_write_master .ADDRESSWIDTH = ADDRESS_WIDTH;
defparam a_burst_write_master .FIFODEPTH = FIFO_DEPTH;
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)

This bursting read master is passed a word aligned address, length in bytes, and a 'go' bit. The master will continue to post full length bursts until the length register reaches a value less than a full burst. A single final burst is then posted and when all the reads return the done bit will be asserted.

To use this master you must simply drive the control signals into this block, 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'.
The signal 'user_data_available' is asserted whenever data is available from the exposed FIFO.

```verilog
module burst_read_master (
    clk,
    reset,
    // control inputs and outputs
    control_fixed_location,
    control_read_base,
    control_read_length,
    control_go,
    control_done,
    control_early_done,
    // user logic inputs and outputs
    user_read_buffer,
    user_buffer_data,
    user_data_available,
    // master inputs and outputs
    master_address,
    master_read,
    master_byteenable,
    master_readdata,
    master_readdatavalid,
    master_burstcount,
    master_waitrequest);

    parameter DATAWIDTH = 32;
    parameter MAXBURSTCOUNT = 4;
    parameter BURSTCOUNTWIDTH = 3;
    parameter BYTEENABLEWIDTH = 4;
    parameter ADDRESSWIDTH = 32;
    parameter FIFODEPTH = 32;
    parameter FIFODEPTH_LOG2 = 5;
    parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead

    input clk;
    input reset;

    // control inputs and outputs
    input control_fixed_location;
    input [ADDRESSWIDTH-1:0] control_read_base;
    input [ADDRESSWIDTH-1:0] control_read_length;
    input control_go;
    output wire control_done;
    output wire control_early_done; // don't use this unless you know what you are doing, it's going to fire when the last read is posted, not when the last data returns!

    // user logic inputs and outputs
    input user_read_buffer;
    output wire [DATAWIDTH-1:0] user_buffer_data;
    output wire user_data_available;

    // master inputs and outputs
    input master_waitrequest;
    input master_readdatavalid;
    input [DATAWIDTH-1:0] master_readdata;
    output wire [ADDRESSWIDTH-1:0] master_address;
    output wire master_read;
```
output wire [BYTEENABLEWIDTH-1:0] master_byteenable;
output wire [BURSTCOUNTWIDTH-1:0] master_burstcount;

// internal control signals
reg control_fixed_location_d1;
wire fifo_empty;
reg [ADDRESSWIDTH-1:0] address;
reg [ADDRESSWIDTH-1:0] length;
reg [FIFODEPTH_LOG2-1:0] reads_pending;
wire increment_address;
wire [BURSTCOUNTWIDTH-1:0] burst_count;
wire [BURSTCOUNTWIDTH-1:0] first_short_burst_count;
wire first_short_burst_enable;
wire [BURSTCOUNTWIDTH-1:0] final_short_burst_count;
wire final_short_burst_enable;
wire [BURSTCOUNTWIDTH-1:0] burst_boundary_word_address;
reg burst_begin;
wire too_many_reads_pending;
wire [FIFODEPTH_LOG2-1:0] fifo_used;

// registering the control_fixed_location bit
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
    begin
      control_fixed_location_d1 <= 0;
    end
  else
    begin
      if (control_go == 1)
        begin
          control_fixed_location_d1 <= control_fixed_location;
        end
    end
end

// master address logic
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
    begin
      address <= 0;
    end
  else
    begin
      if (control_go == 1)
        begin
          address <= control_read_base;
        end
      else if ((increment_address == 1) & (control_fixed_location_d1 == 0))
        begin
          address <= address + (burst_count * BYTEENABLEWIDTH);  // always performing word size accesses, increment by the burst count presented
        end
    end
end

// master length logic
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
    begin
length <= 0;
end
else
begin
  if (control_go == 1)
  begin
    length <= control_read_length;
  end
else if (increment_address == 1)
  begin
    length <= length - (burst_count * BYTEENABLEWIDTH); // always performing word size accesses, decrement by the burst count presented
  end
end

// controlled signals going to the master/control ports
assign master_address = address;
assign master_byteenable = -1; // all ones, always performing word size accesses
assign master_burstcount = burst_count;
assign control_done = (length == 0) & (reads_pending == 0); // need to make sure that the reads have returned before firing the done bit
assign control_early_done = (length == 0); // advanced feature, you should use 'control_done' if you need all the reads to return first
assign burst_boundary_word_address = ((address / BYTEENABLEWIDTH) & (MAXBURSTCOUNT - 1));
assign first_short_burst_enable = (burst_boundary_word_address != 0);
assign final_short_burst_enable = (length < (MAXBURSTCOUNT * BYTEENABLEWIDTH));
assign first_short_burst_count = ((burst_boundary_word_address & 1'b1) == 1'b1) ? 1 : // if the burst boundary isn’t a multiple of 2 then must post a burst of 1 to get to a multiple of 2 for the next burst

  (((MAXBURSTCOUNT - burst_boundary_word_address) < (length / BYTEENABLEWIDTH))
   & (MAXBURSTCOUNT - burst_boundary_word_address) * (length / BYTEENABLEWIDTH)));
assign final_short_burst_count = (length / BYTEENABLEWIDTH);
assign burst_count = (first_short_burst_enable == 1) ? first_short_burst_count : // this will get the transfer back on a burst boundary,
  (final_short_burst_enable == 1) ? final_short_burst_count : MAXBURSTCOUNT;
assign increment_address = (too_many_reads_pending == 0) & (master_waitrequest == 0) & (length != 0);
assign too_many_reads_pending = (reads_pending + fifo_used) >= (FIFODEPTH - MAXBURSTCOUNT - 4); // make sure there are fewer reads posted than room in the FIFO

// tracking FIFO
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
  begin
    reads_pending <= 0;
  end
else
begin
  if (increment_address == 1)
  begin
    if (master_readdatavalid == 0)
    begin
      reads_pending <= reads_pending + burst_count;
    end
  end
end
reads_pending <= reads_pending + burst_count − 1;  // a burst read was posted, but a word returned 
end 
end 
else 
begin 
if ( master_readdatavalid == 0) 
begin 
reads_pending <= reads_pending;  // burst read was not posted and no read returned 
end 
else 
begin 
reads_pending <= reads_pending − 1;  // burst read was not posted but a word returned 
end 
end 
end 
end 
end 

// read data feeding user logic 
assign user_data_available = !fifo_empty; 
scfifo the_master_to_user_fifo ( 
.aclr ( reset ), 
.clock ( clk ), 
.data ( master_readdata ), 
.empty ( fifo_empty ), 
.q ( user_buffer_data ), 
.rdreq ( user_read_buffer ), 
.usedw ( fifo_used ), 
.wrreq ( master_readdatavalid ) ); 
defparam the_master_to_user_fifo.lpm_width = DATAWIDTH; 
defparam the_master_to_user_fifo.lpm_numwords = FIFODEPTH; 
defparam the_master_to_user_fifo.lpm_showahead = "ON"; 
defparam the_master_to_user_fifo.use_eab = ( FIFOUSEMEMORY == 1)? "ON" : "OFF"; 
defparam the_master_to_user_fifo.add_ram_output_register = "OFF"; 
defparam the_master_to_user_fifo.underflow_checking = "OFF"; 
defparam the_master_to_user_fifo.overflow_checking = "OFF"; 
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)
This latency aware read master is passed a word aligned address, length in bytes, and a 'go' bit. The master will continue to post reads until the length register reaches a value of zero. When all the reads return the done bit will be asserted.

To use this master you must simply drive the control signals into this block, 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'. The signal 'user_data_available' is asserted whenever data is available from the exposed FIFO.

```plaintext
module latency_aware_read_master (  
    clk,  
    reset,  
    // control inputs and outputs  
    control_fixed_location,  
    control_read_base,  
    control_read_length,  
    control_go,  
    control_done,  
    control_early_done,  
    // user logic inputs and outputs  
    user_read_buffer,  
    user_buffer_data,  
    user_data_available,  
    // master inputs and outputs  
    master_address,  
    master_read,  
    master_byteenable,  
    master_readdata,  
    master_readdatavalid,  
    master_waitrequest  
);  
parameter DATAWIDTH = 32;  
parameter BYTEENABLEREG = 4;  
parameter ADDRESSWIDTH = 32;  
parameter FIFODEPTH = 32;  
parameter FIFODEPTH_LOG2 = 5;  
parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead  
input clk;  
input reset;  

// control inputs and outputs  
input control_fixed_location;  
input [ADDRESSWIDTH-1:0] control_read_base;  
input [ADDRESSWIDTH-1:0] control_read_length;  
input control_go;  
output wire control_done;  
output wire control_early_done; // don’t use this unless you know what you are doing!  
// user logic inputs and outputs  
input user_read_buffer;  
output wire [DATAWIDTH-1:0] user_buffer_data;  
output wire user_data_available;  
// master inputs and outputs  
input master_waitrequest;  
input master_readdatavalid;  
```
input [DATAWIDTH−1:0] master_readdata;
output wire [ADDRESSWIDTH−1:0] master_address;
output wire master_read;
output wire [BYTEENABLEWIDTH−1:0] master_byteenable;

// internal control signals
reg control_fixed_location_d1;
wire fifo_empty;
reg [ADDRESSWIDTH−1:0] address;
reg [ADDRESSWIDTH−1:0] length;
reg [FIFODEPTH_LOG2−1:0] reads_pending;
wire increment_address;
wire too_many_pending_reads;
reg too_many_pending_reads_d1;
wire [FIFODEPTH_LOG2−1:0] fifo_used;

// registering the control_fixed_location_bit
always @(posedge clk or posedge reset)
begin
  if (reset == 1) begin
    control_fixed_location_d1 <= 0;
    end
  else begin
    if (control_go == 1)
      begin
        control_fixed_location_d1 <= control_fixed_location;
      end
    end
  end
end

// master address logic
assign master_address = address;
assign master_byteenable = −1;  // all ones, always performing word size accesses
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
    begin
      address <= 0;
    end
  else begin
    if (control_go == 1)
      begin
        address <= control_read_base;
      end
    else if ((increment_address == 1) & (control_fixed_location_d1 == 0))
      begin
        address <= address + BYTEENABLEWIDTH;  // always performing word size accesses
      end
  end
end

// master length logic
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
    begin
      length <= 0;
    end
else
begin
  if (control_go == 1)
  begin
    length <= control_read_length;
  end
else if (increment_address == 1)
begin
  length <= length - BYTEENABLE_WIDTH; // always performing word size accesses
end
end

// control logic
assign too_many_pending_reads = (fifo_used + reads_pending) >= (FIFODEPTH - 4);
assign master_read = (length != 0) & (too_many_pending_reads.d1 == 0);
assign increment_address = (length != 0) & (too_many_pending_reads.d1 == 0) & (master_waitrequest == 0);
assign control_done = (reads_pending == 0) & (length == 0); // master done posting
assign control_early_done = (length == 0); // if you need all the pending reads to return then use 'control_done' instead of this signal

always @(posedge clk)
begin
  if (reset == 1)
  begin
    too_many_pending_reads_d1 <= 0;
  end
else
begin
  too_many_pending_reads_d1 <= too_many_pending_reads;
end
end

always @(posedge clk or posedge reset)
begin
  if (reset == 1)
  begin
    reads_pending <= 0;
  end
else
begin
  if (increment_address == 1)
  begin
    if (master_readdatavalid == 0)
    begin
      reads_pending <= reads_pending + 1;
    end
else
begin
  reads_pending <= reads_pending; // a read was posted, but another returned
end
end
else
begin
  if (master_readdatavalid == 0)
  begin
    reads_pending <= reads_pending; // read was not posted and no read returned
  end
else
begin
  reads_pending <= reads_pending - 1; // read was not posted but a read returned
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)

/*
 * Legal Notice: (C)2007 Altera Corporation. All rights reserved. Your
 * use of Altera Corporation’s design tools, logic functions and other
 * software and tools, and its AMPP partner logic functions, and any
 * output files any of the foregoing (including device programming or
 * simulation files), and any associated documentation or information are
 * expressly subject to the terms and conditions of the Altera Program
 * License Subscription Agreement or other applicable license agreement,
 * including, without limitation, that your use is for the sole purpose
 * of programming logic devices manufactured by Altera and sold by Altera
 * or its authorized distributors. Please refer to the applicable
 * agreement for further details.
 */

/*
Author: JCJB
Date: 11/04/2007

This simple write master is passed a word aligned address, length in bytes,
and a 'go' bit. The master will continue to post writes until the length register
reaches zero. When the length register reaches zero the 'done' bit is asserted.

To use this master you must simply drive the control signals into this block,
and also write the data to the exposed write FIFO. To read from the exposed FIFO
use the 'user_write_buffer' signal to push data into the FIFO 'user_buffer_data'.
The signal 'user_buffer_full' is asserted whenever the exposed buffer is full.
You should not attempt to write data to the exposed FIFO if it is full.
*/

// altera message_off 10230
module write_master(
    clk,
    reset,
    // control inputs and outputs
    control_fixed_location,
    control_write_base,
    control_write_length,
    control_go,
    control_done,
    // user logic inputs and outputs
    user_write_buffer,
    user_buffer_data,
    user_buffer_full,
    // master inputs and outputs
    master_address,
    master_write,
    master_byteenable,
    master_writedata,
    master_waitrequest);

parameter DATAWIDTH = 32;
parameter BYTEENABLEWIDTH = 4;
parameter ADDRESSWIDTH = 32;
parameter FIFODEPTH = 32;
parameter FIFODEPTH_LOG2 = 5;
parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead

input clk;
input reset;

// control inputs and outputs
input control_fixed_location; // this only makes sense to enable when MAXBURSTCOUNT = 1
input [ADDRESSWIDTH−1:0] control_write_base;
input [ADDRESSWIDTH−1:0] control_write_length;
input control_go;
output wire control_done;

// user logic inputs and outputs
input user_write_buffer;
input [DATAWIDTH−1:0] user_buffer_data;
output wire user_buffer_full;

// master inputs and outputs
input master_waitrequest;
output wire [ADDRESSWIDTH−1:0] master_address;
output wire master_write;
output wire [BYTEENABLEWIDTH−1:0] master_byteenable;
output wire [DATAWIDTH−1:0] master_writedata;

// internal control signals
reg control_fixed_location_d1;
reg [ADDRESSWIDTH−1:0] address; // this increments for each word
reg [ADDRESSWIDTH−1:0] length;
wire increment_address; // this increments the 'address' register when write is asserted and waitrequest is de-asserted
wire read_fifo;
wire user_buffer_empty;
// registering the control_fixed_location bit
always @(posedge clk or posedge reset)
begin
    if (reset == 1)
        begin
            control_fixed_location_d1 <= 0;
        end
    else
        begin
            if (control_go == 1)
                begin
                    control_fixed_location_d1 <= control_fixed_location;
                end
        end
end

// master word increment counter
always @(posedge clk or posedge reset)
begin
    if (reset == 1)
        begin
            address <= 0;
        end
    else
        begin
            if (control_go == 1)
                begin
                    address <= control_write_base;
                end
            else if ((increment_address == 1) & (control_fixed_location_d1 == 0))
                begin
                    address <= address + BYTEENABLEWIDTH; // always performing word size accesses
                end
        end
end

// master length logic
always @(posedge clk or posedge reset)
begin
    if (reset == 1)
        begin
            length <= 0;
        end
    else
        begin
            if (control_go == 1)
                begin
                    length <= control_write_length;
                end
            else if (increment_address == 1)
                begin
                    length <= length - BYTEENABLEWIDTH; // always performing word size accesses
                end
        end
end

// controlled signals going to the master/control ports
assign master_address = address;
assign master_byteenable = -1; // all ones, always performing word size accesses
assign control_done = (length == 0);
assign master_write = (user_buffer_empty == 0) & (control_done == 0);
```verilog
assign increment_address = (user_buffer_empty == 0) & (master_waitrequest == 0) & (control_done == 0);
assign read_fifo = increment_address;

// write data feed by user logic
scfifo the_user_to_master_fifo (  
    .aclr (reset),  
    .clock (clk),  
    .data (user_buffer_data),  
    .full (user_buffer_full),  
    .empty (user_buffer_empty),  
    .q (master_writedata),  
    .rdreq (read_fifo),  
    .wrreq (user_write_buffer) );
defparam the_user_to_master_fifo.lpm_width = DATAWIDTH;
defparam the_user_to_master_fifo.lpm_numwords = FIFODEPTH;
defparam the_user_to_master_fifo.lpm_showahead = "ON";
defparam the_user_to_master_fifo.use_eab = (FIFOUSEMEMORY == 1)? "ON" : "OFF";
defparam the_user_to_master_fifo.add_ram_output_register = "OFF";
defparam the_user_to_master_fifo.underflow_checking = "OFF";
defparam the_user_to_master_fifo.overflow_checking = "OFF";
endmodule
```

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

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

```verilog
module burst_write_master (  
    aclr (reset),  
    clock (clk),  
    data (user_buffer_data),  
    full (user_buffer_full),  
    empty (user_buffer_empty),  
    q (master_writedata),  
    rdreq (read_fifo),  
    wrreq (user_write_buffer) );
defparam the_user_to_master_fifo.lpm_width = DATAWIDTH;
defparam the_user_to_master_fifo.lpm_numwords = FIFODEPTH;
defparam the_user_to_master_fifo.lpm_showahead = "ON";
defparam the_user_to_master_fifo.use_eab = (FIFOUSEMEMORY == 1)? "ON" : "OFF";
defparam the_user_to_master_fifo.add_ram_output_register = "OFF";
defparam the_user_to_master_fifo.underflow_checking = "OFF";
defparam the_user_to_master_fifo.overflow_checking = "OFF";
endmodule
```

This bursting write master is passed a word aligned address, length in bytes, and a 'go' bit. The master will continue to post full length bursts until the length register reaches a value less than a full burst. A single final burst is then posted when enough data has been buffered and then the done bit will be asserted.

To use this master you must simply drive the control signals into this block, and also write the data to the exposed write FIFO. To read from the exposed FIFO use the 'user_write_buffer' signal to push data into the FIFO 'user_buffer_data'. The signal 'user_buffer_full' is asserted whenever the exposed buffer is full. You should not attempt to write data to the exposed FIFO if it is full.

--

*/

// altera_message_off 10230

module burst_write_master (  
  aclr (reset),  
  clock (clk),  
  data (user_buffer_data),  
  full (user_buffer_full),  
  empty (user_buffer_empty),  
  q (master_writedata),  
  rdreq (read_fifo),  
  wrreq (user_write_buffer) );
defparam the_user_to_master_fifo.lpm_width = DATAWIDTH;
defparam the_user_to_master_fifo.lpm_numwords = FIFODEPTH;
defparam the_user_to_master_fifo.lpm_showahead = "ON";
defparam the_user_to_master_fifo.use_eab = (FIFOUSEMEMORY == 1)? "ON" : "OFF";
defparam the_user_to_master_fifo.add_ram_output_register = "OFF";
defparam the_user_to_master_fifo.underflow_checking = "OFF";
defparam the_user_to_master_fifo.overflow_checking = "OFF";
endmodule
```
// control inputs and outputs
control_fixed_location, control_write_base, control_write_length, control_go, control_done,

// user logic inputs and outputs
user_write_buffer, user_buffer_data, user_buffer_full,

// master inputs and outputs
master_address, master_write, master_byteenable, master_writedata, master_burstcount, master_waitrequest
);

parameter DATAWIDTH = 32;
parameter MAXBURSTCOUNT = 4;
parameter BURSTCOUNTWIDTH = 3;
parameter BYTEENABLEWIDTH = 4;
parameter ADDRESSWIDTH = 32;
parameter FIFODEPTH = 32; // must be at least twice MAXBURSTCOUNT in order to be efficient
parameter FIFODEPTH_LOG2 = 5;
parameter FIFOUSEMEMORY = 1; // set to 0 to use LEs instead

input clk;
input reset;

// control inputs and outputs
input control_fixed_location; // this only makes sense to enable when MAXBURSTCOUNT = 1
input [ADDRESSWIDTH−1:0] control_write_base;
input [ADDRESSWIDTH−1:0] control_write_length;
input control_go;
output wire control_done;

// user logic inputs and outputs
input user_write_buffer;
input [DATAWIDTH−1:0] user_buffer_data;
output wire user_buffer_full;

// master inputs and outputs
input master_waitrequest;
output reg [ADDRESSWIDTH−1:0] master_address;
output wire master_write;
output wire [BYTEENABLEWIDTH−1:0] master_byteenable;
output wire [DATAWIDTH−1:0] master_writedata;
output reg [BURSTCOUNTWIDTH−1:0] master_burstcount;

// internal control signals
reg control_fixed_location_d1;
reg [ADDRESSWIDTH−1:0] length;
wire final_short_burst_enable; // when the length is less than MAXBURSTCOUNT * # of bytes per word (BYTEENABLEWIDTH) (i.e. end of the transfer)
wire final_short_burst_ready; // when there is enough data in the FIFO for the final burst
wire [BURSTCOUNTWIDTH−1:0] burst_boundary_word_address; // represents the word offset within the burst boundary
wire [BURSTCOUNTWIDTH−1:0] first_short_burst_count;
wire [BURSTCOUNTWIDTH−1:0] final_short_burst_count;
wire first_short_burst_ready; // when the transfer doesn’t start on a burst boundary
wire full_burst_ready; // when there is enough data in the FIFO to get the master back into burst alignment
wire increment_address; // this increments the ’address’ register when write is asserted and waitrequest is de−asserted
wire burst_begin; // used to register the registers ’burst_address’ and ’burst_count_d1’ as well as drive the master_address and burst_count muxes
wire read_fifo;
wire [FIFODEPTH_LOG2−1:0] fifo_used; // going to combined used with the full bit
wire [BURSTCOUNTWIDTH−1:0] burst_count; // watermark of the FIFO, it has a latency of 2 cycles
reg [BURSTCOUNTWIDTH−1:0] burst_counter;
reg first_transfer; // need to keep track of the first burst so that we don’t incorrectly increment the address

// registering the control_fixed_location bit
always @(posedge clk or posedge reset)
begin
if (reset == 1)
begin
control_fixed_location_d1 <= 0;
end
else
begin
if (control_go == 1)
begin
control_fixed_location_d1 <= control_fixed_location;
end
end
end

// set when control_go fires. and reset once the first burst starts
always @(posedge clk or posedge reset)
begin
if (reset == 1)
begin
first_transfer <= 0;
end
else
begin
if (control_go == 1)
begin
first_transfer <= 1;
end
else if (burst_begin == 1)
begin
first_transfer <= 0;
end
end
end

// master address (held constant during burst)
always @(posedge clk or posedge reset)
begin
if (reset == 1)
begin
master_address <= 0;
end
else
begin
if (control_go == 1)
begin
master_address <= control_write_base;
end
else if ((first_transfer == 0) & (burst_begin == 1) & (control_fixed_location_d1 == 0))
begin
master_address <= master_address + (master_burstcount * BYTEENABLEWIDTH); // we don’t want address + BYTEENABLEWIDTH for the first access
end
end

// master length logic
always @(posedge clk or posedge reset)
begin
if (reset == 1)
begin
length <= 0;
end
else
begin
if (control_go == 1)
begin
length <= control_write_length;
end
else if (increment_address == 1)
begin
length <= length - BYTEENABLEWIDTH; // always performing word size accesses
end
end
end

// register the master burstcount (held constant during burst)
always @(posedge clk or posedge reset)
begin
if (reset == 1)
begin
master_burstcount <= 0;
end
else
begin
if (burst_begin == 1)
begin
master_burstcount <= burst_count;
end
end
end

// burst counter. This is set to the burst count being posted then counts down when each word
// of data goes out. If it reaches 0 (i.e. not reloaded after 1) then the master stalls
due to
// a lack of data to post a new burst.
always @(posedge clk or posedge reset)
begin
if (reset == 1)
begin
burst_counter <= 0;
end
else
begin

195
if (control_go == 1)
begin
    burst_counter <= 0;
end
else if (burst_begin == 1)
begin
    burst_counter <= burst_count;
end
else if (increment_address == 1) // decrements each write, burst_counter will only
hit 0 if burst begin doesn’t fire on the next cycle
begin
    burst_counter <= burst_counter - 1;
end
end

// burst boundaries are on the master "width * maximum burst count". The burst boundary
word address will be used to determine how far off the boundary the transfer starts from.
assign burst_boundary_word_address = ((master_address / BYTEENABLEWIDTH) & (MAXBURSTCOUNT - 1));

// first short burst enable will only be active on the first transfer (if applicable).
// It will either post the amount of words remaining to reach the end of the burst
// boundary or post the remainder of the transfer whichever is shorter. If the transfer
// is very short and not aligned on a burst boundary then the same logic as the final
// short transfer is used
assign first_short_burst_enable = (burst_boundary_word_address != 0) & (first_transfer == 1);
assign first_short_burst_count = ((burst_boundary_word_address & 1'b1) == 1'b1) ? 1 : //
    if the burst boundary isn’t a multiple of 2 then must post a burst of 1 to get to a
    multiple of 2 for the next burst
    ((MAXBURSTCOUNT - burst_boundary_word_address) < (length / BYTEENABLEWIDTH)) ?
    (MAXBURSTCOUNT - burst_boundary_word_address) : final_short_burst_count;
assign first_short_burst_ready = (fifo_used > first_short_burst_count) | ((fifo_used ==
    first_short_burst_count) & (burst_counter == 0));

// when there isn’t enough data for a full burst at the end of the transfer a short
// burst is sent out instead
assign final_short_burst_enable = (length < (MAXBURSTCOUNT * BYTEENABLEWIDTH));
assign final_short_burst_count = (length / BYTEENABLEWIDTH);
assign final_short_burst_ready = (fifo_used > final_short_burst_count) | ((fifo_used ==
    final_short_burst_count) & (burst_counter == 0)); // this will add a one cycle
stall between bursts, since fifo_used has a cycle of latency, this only affects the
last burst

// since the fifo has a latency of 1 we need to make sure we don’t under flow
assign full_burst_ready = (fifo_used > MAXBURSTCOUNT) | ((fifo_used == MAXBURSTCOUNT) &
    (burst_counter == 0)); // when fifo used watermark equals the burst count the
statemachine must stall for one cycle, this will make sure that when a burst begins
there really is enough data present in the FIFO

assign master_bytenable = -1; // all ones, always performing word size accesses
assign control_done = (length == 0);
assign master_write = (control_done == 0) & (burst_counter != 0); // burst_counter = 0
means the transfer is done, or not enough data in the fifo for a new burst

// fifo controls and the burst_begin responsible for timing most of the circuit,
burst_begin starts the writing statemachine
assign burst_begin = (((first_short_burst_enable == 1) & (first_short_burst_ready == 1)) |
    ((final_short_burst_enable == 1) & (final_short_burst_ready == 1)) |
    (full_burst_ready == 1))
& (control_done == 0) // since the FIFO can have data before the master
starts we need to disable this bit from firing when length = 0
& ((burst_counter == 0) | ((burst_counter == 1) & (master_waitrequest == 0) &
(length > (MAXBURSTCOUNT * BYTEENABLEWIDTH)))); // need to make a short
final burst doesn’t start right after a full burst completes.

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
cover both
(final_short_burst_enable == 1)? final_short_burst_count : MAXBURSTCOUNT;
assign increment_address = (master_write == 1) & (master_waitrequest == 0); // writing
is occurring without wait states
assign read_fifo = increment_address;

// write data feed by user logic
scfifo the_user_to_master_fifo (  
  aclr (reset),  
  usedw (fifo_used),  
  clock (clk),  
  .data (user_buffer.data),  
  .almost_full (user_buffer.full),  
  .q (master_writedata),  
  .rdreq (read_fifo),  
  .wrreq (user_write_buffer)  
);
defparam the_user_to_master_fifo.lpm_width = DATAWIDTH;
defparam the_user_to_master_fifo.lpm_numwords = FIFODEPTH;
defparam the_user_to_master_fifo.lpm_showahead = "ON";
defparam the_user_to_master_fifo.almost_full_value = (FIFODEPTH - 2);
defparam the_user_to_master_fifo.use_eab = (FIFOUSEMEMORY == 1)? "ON" : "OFF";
defparam the_user_to_master_fifo.add_ram_output_register = "OFF"; // makes timing the
burst begin single simplier
defparam the_user_to_master_fifo.underflow_checking = "OFF";
defparam the_user_to_master_fifo.overflow_checking = "OFF";
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)

module interrupt_logic (  
  clk,  
  reset,  
  data_in,  
  write,  
  write_data,  
  address_decode,  
  irq_mask_reg_en,  
  edge_capture_reg_en,  
  read_data,  
  irq_out  
);

parameter DATA_WIDTH = 32;
input clk;
input reset;
input [DATA_WIDTH-1:0] data_in;
input write;
input [DATA_WIDTH-1:0] write_data;
input address_decode;
input irq_mask_reg_en;

197
input edge_capture_reg_en;
output reg [DATA_WIDTH-1:0] read_data;
output wire irq_out;

// internal logic
reg [DATA_WIDTH-1:0] data_in_d1;
reg [DATA_WIDTH-1:0] data_in_d2;
reg [DATA_WIDTH-1:0] data_in_d3;
wire [DATA_WIDTH-1:0] edge_detect;
reg [DATA_WIDTH-1:0] edge_capture;
reg [DATA_WIDTH-1:0] irq_mask;
wire edge_capture_wr_strobe;
wire irq_mask_wr_strobe;
wire [DATA_WIDTH-1:0] readdata_mux;

// interrupt mask register
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
  begin
    irq_mask <= 0;
  end
  else if (irq_mask_wr_strobe)
  begin
    irq_mask <= write_data;
  end
end

// double registers for asynchronous input and assure metastability
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
  begin
    data_in_d1 <= 0;
    data_in_d2 <= 0;
  end
  else
  begin
    data_in_d1 <= data_in;
    data_in_d2 <= data_in_d1;
  end
end

// edge detection logic
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
  begin
    data_in_d3 <= 0;
  end
  else
  begin
    data_in_d3 <= data_in_d2;
  end
end

assign edge_detect = data_in_d2 & ~data_in_d3;

// edge capture registers with separate clear bit
generate
  genvar BIT;
  for (BIT = 0; BIT < DATA_WIDTH; BIT = BIT + 1)
  begin: edge_capture_generation
    always @(posedge clk or posedge reset)
    begin
      if (reset == 1)
      begin
        ...
edge_capture[B] <= 0;
end
else
begin
  if (edge_capture_wr_strobe && write_data[B])
    edge_capture[B] <= 0;
  else if (edge_detect[B])
    edge_capture[B] <= 1;
end
end
endgenerate

// register the readdata_mux
always @(posedge clk or posedge reset)
begin
  if (reset == 1)
    read_data <= 0;
  else
    begin
      read_data <= readdata_mux;
    end
end

assign readdata_mux = (DATA_WIDTH((irq_mask_reg_en == 1'b1)) & irq_mask) | (DATA_WIDTH((edge_capture_reg_en == 1'b1)) & edge_capture);
assign irq_mask_wr_strobe = (write == 1'b1) && (address_decode == 1'b1) && (irq_mask_reg_en == 1'b1);
assign edge_capture_wr_strobe = (write == 1'b1) && (address_decode == 1'b1) && (edge_capture_reg_en == 1'b1);
assign irq_out = (edge_capture & irq_mask);
endmodule

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

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

This slave component has a parameterizable data width and 16 input/output words. There are five modes for each addressable word in this component as follows:

Mode = 0 --> Output only
Mode = 1 --> Input only
Mode = 2 --> Output and input (independent I/O, default)
Mode = 3 --> Output with loopback (software readable output registers)
Mode = 4 --> Disabled
This component is always available so the waitrequest signal is not used and it has fixed read and write latencies. The write latency is 0 and the read latency is 3 cycles. If you attempt to access a location that doesn’t support the necessary functionality then you will either write to a non-existent register (write to space) or will readback 0 as the inputs will be grounded if they are disabled. Inputs or outputs that are removed will be due to the component tcl file stubbing the signals. Disabled outputs will not be exposed at the top of the system and the Quartus II software will optimize the register away. Disabled inputs (except in the loopback mode) will not be exposed at the top and internally be wired to ground. The Quartus II software as a result will optimize the input registers to be hardcoded wires set to ground automatically.

In order for your external logic to know which register is being accessed you will need to enable ‘ENABLE_SYNC_SIGNALS’ by setting it to 1. When enabled, the user_chipselect/byteenable/read/write signals will be exposed to your external logic which you can use to determine which register is being accessed and whether it’s a read or write access.

If you use the synchronization signals use the following to qualify them:

Read: user_chipselect[x] AND user_read
Write: user_chipselect[x] AND user_write AND user_byteenable

Note: Reads return the full word regardless of the byteenables presented.

module slave_template (  
// signals to connect to an Avalon clock source interface  
clk,  
reset,  

// signals to connect to an Avalon-MM slave interface  
slave_address,  
slave_read,  
slave_write,  
slave_readdata,  
slave_writedata,  
slave_byteenable,  

// interrupt signals  
slave_irq,  

// signals to connect to custom user logic (up to 16 input and output pairs)  
user_dataout_0,  
user_dataout_1,  
user_dataout_2,  
user_dataout_3,  
user_dataout_4,  
user_dataout_5,  
user_dataout_6,  
user_dataout_7,  
user_dataout_8,  
user_dataout_9,  
user_dataout_10,  
user_dataout_11,  
user_dataout_12,  
user_dataout_13,  
user_dataout_14,  
user_dataout_15,  
user_datain_0,  
user_datain_1,  
user_datain_2,  
user_datain_3.

200
user_datain_4,
user_datain_5,
user_datain_6,
user_datain_7,
user_datain_8,
user_datain_9,
user_datain_10,
user_datain_11,
user_datain_12,
user_datain_13,
user_datain_14,
user_datain_15,

// optional signals so that your external logic knows what location is being accessed
user_chipselect,
user_byteenable,
user_write,
user_read
);

// most of the set values will only be used by the component .tcl file. The DATA_WIDTH
// and MODEX = 3 influence the hardware created.
// ENABLE_SYNC_SIGNALS isn’t used by this hardware at all but it provided anyway so that
// it can be exposed in the component .tcl file
// to control the stubbing of certain signals.
parameter DATA_WIDTH = 32;   // word size of each input and output register
parameter ENABLE_SYNC_SIGNALS = 0;   // only used by the component .tcl file, 1 to expose
    user_chipselect/write/read. 0 to stub them
parameter MODE_0 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_1 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_2 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_3 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_4 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_5 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_6 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_7 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_8 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_9 = 2;    // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_10 = 2;   // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_11 = 2;   // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_12 = 2;   // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_13 = 2;   // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_14 = 2;   // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter MODE_15 = 2;   // 0 = Output, 1 = Input, 2 = Ouput and Input, 3 =
    Output with loopback, 4 = Disabled
parameter IRQ_EN = 0;  // 0 = Enable interrupt, 1 = Disable interrupt

// clock interface
input clk;
input reset;
// slave interface
input [8:0] slave_address;
input slave_read;
input slave_write;
output wire [DATA_WIDTH-1:0] slave_readdata;
input [DATA_WIDTH-1:0] slave_writedata;
input ((DATA_WIDTH/8)-1:0) slave_byteenable;
output wire slave_irq;

// user interface
output wire [DATA_WIDTH-1:0] user_dataout_0;
output wire [DATA_WIDTH-1:0] user_dataout_1;
output wire [DATA_WIDTH-1:0] user_dataout_2;
output wire [DATA_WIDTH-1:0] user_dataout_3;
output wire [DATA_WIDTH-1:0] user_dataout_4;
output wire [DATA_WIDTH-1:0] user_dataout_5;
output wire [DATA_WIDTH-1:0] user_dataout_6;
output wire [DATA_WIDTH-1:0] user_dataout_7;
output wire [DATA_WIDTH-1:0] user_dataout_8;
output wire [DATA_WIDTH-1:0] user_dataout_9;
output wire [DATA_WIDTH-1:0] user_dataout_10;
output wire [DATA_WIDTH-1:0] user_dataout_11;
output wire [DATA_WIDTH-1:0] user_dataout_12;
output wire [DATA_WIDTH-1:0] user_dataout_13;
output wire [DATA_WIDTH-1:0] user_dataout_14;
output wire [DATA_WIDTH-1:0] user_dataout_15;
input [DATA_WIDTH-1:0] user_datain_0;
input [DATA_WIDTH-1:0] user_datain_1;
input [DATA_WIDTH-1:0] user_datain_2;
input [DATA_WIDTH-1:0] user_datain_3;
input [DATA_WIDTH-1:0] user_datain_4;
input [DATA_WIDTH-1:0] user_datain_5;
input [DATA_WIDTH-1:0] user_datain_6;
input [DATA_WIDTH-1:0] user_datain_7;
input [DATA_WIDTH-1:0] user_datain_8;
input [DATA_WIDTH-1:0] user_datain_9;
input [DATA_WIDTH-1:0] user_datain_10;
input [DATA_WIDTH-1:0] user_datain_11;
input [DATA_WIDTH-1:0] user_datain_12;
input [DATA_WIDTH-1:0] user_datain_13;
input [DATA_WIDTH-1:0] user_datain_14;
input [DATA_WIDTH-1:0] user_datain_15;
output wire [15:0] user_chipselect;
output wire [(DATA_WIDTH/8)-1:0] user_byteenable;
output wire user_write;
output wire user_read;

// internal logic signals
wire [(DATA_WIDTH/8)-1:0] internal_byteenable; // when DATA_WIDTH is 8 bits need to
har core this signal to 1 since the fabric doesn’t support 1 bit byte enables
reg [(DATA_WIDTH/8)-1:0] internal_byteenable_d1;
wire [15:0] address_decode;
reg [15:0] address_decode_d1; // used to select the first stage of mux
pipeline (a, b, c, or d)
wire [3:0] address_bank_decode; // used to select the second stage of mux
pipeline (mux a, b, c, or d)
reg [3:0] address_bank_decode_d1; // used to select the second stage of mux
pipeline (mux a, b, c, or d)
reg slave_read_d1; // used to qualify the first stage of mux
pipeline (a, b, c, or d)
reg slave_read_d2; // used to qualify the second stage of mux
pipeline (slave_readdata)
reg slave_write_d1; // used by the option user_write signal
reg [DATA_WIDTH-1:0] user_datain_0_d1;
reg [DATA_WIDTH-1:0] user_datain_1_d1;
// when the data width is 8 need to hardcode the 1 bit byteenable to high
generate
if (DATA_WIDTH == 8)
begin
    assign internal_byteenable = 1'b1;
end
else
begin
    assign internal_byteenable = slave_byteenable;
end
endgenerate

// sixteen address decodes (using one-hot encoding) A bank is considered to be a
grouping of four addresses.
assign address_decode[0] = (slave_address[7:4] == 4'b0000) & (slave_write | slave_read);
assign address_decode[1] = (slave_address[7:4] == 4'b0001) & (slave_write | slave_read);
assign address_decode[2] = (slave_address[7:4] == 4'b0010) & (slave_write | slave_read);
assign address_decode[3] = (slave_address[7:4] == 4'b0011) & (slave_write | slave_read);
assign address_decode[4] = (slave_address[7:4] == 4'b0100) & (slave_write | slave_read);
assign address_decode[5] = (slave_address[7:4] == 4’b0101) & (slave_write | slave_read);
assign address_decode[6] = (slave_address[7:4] == 4’b0110) & (slave_write | slave_read);
assign address_decode[7] = (slave_address[7:4] == 4’b0111) & (slave_write | slave_read);
assign address_decode[8] = (slave_address[7:4] == 4’b1000) & (slave_write | slave_read);
assign address_decode[9] = (slave_address[7:4] == 4’b1001) & (slave_write | slave_read);
assign address_decode[10] = (slave_address[7:4] == 4’b1010) & (slave_write | slave_read);
assign address_decode[11] = (slave_address[7:4] == 4’b1101) & (slave_write | slave_read);
assign address_decode[12] = (slave_address[7:4] == 4’b1100) & (slave_write | slave_read);
assign address_decode[13] = (slave_address[7:4] == 4’b1101) & (slave_write | slave_read);
assign address_decode[14] = (slave_address[7:4] == 4’b1110) & (slave_write | slave_read);
assign address_decode[15] = (slave_address[7:4] == 4’b1111) & (slave_write | slave_read);
assign address_bank_decode[0] = (address_decode_d1[3:0] != 0) ? 1’b1 : 1’b0;
assign address_bank_decode[1] = (address_decode_d1[7:4] != 0) ? 1’b1 : 1’b0;
assign address_bank_decode[2] = (address_decode_d1[11:8] != 0) ? 1’b1 : 1’b0;
assign address_bank_decode[3] = (address_decode_d1[15:12] != 0) ? 1’b1 : 1’b0;
assign user_readout_reg_en = (slave_address[1:0] == 2’b00);
assign irq_mask_reg_en = (slave_address[1:0] == 2’b01);
assign edge_capture_reg_en = (slave_address[1:0] == 2’b10);
assign edge_capture_reg_en = (slave_address[1:0] == 2’b11);

// registering various address decoding registers and the input data
always @ (posedge clk or posedge reset)
begin
  if (reset == 1)
    begin
      slave_read_d1 <= 0;
      slave_read_d2 <= 0;
      slave_write_d1 <= 0;
      address_decode_d1 <= 0;
      address_bank_decode_d1 <= 0;
      internal_bytencode_d1 <= 0;
      user_datain_0_d1 <= 0;
      user_datain_1_d1 <= 0;
      user_datain_2_d1 <= 0;
      user_datain_3_d1 <= 0;
      user_datain_4_d1 <= 0;
      user_datain_5_d1 <= 0;
      user_datain_6_d1 <= 0;
      user_datain_7_d1 <= 0;
      user_datain_8_d1 <= 0;
      user_datain_9_d1 <= 0;
      user_datain_10_d1 <= 0;
      user_datain_11_d1 <= 0;
      user_datain_12_d1 <= 0;
      user_datain_13_d1 <= 0;
      user_datain_14_d1 <= 0;
      user_datain_15_d1 <= 0;
    end
  else
    begin
      slave_read_d1 <= slave_read;
      slave_read_d2 <= slave_read_d1;
      slave_write_d1 <= slave_write;
      internal_bytencode_d1 <= internal_bytencode;
      if ((slave_read == 1) | (slave_write == 1))
        begin
          address_decode_d1 <= address_decode;
        end
      if (slave_read_d1 == 1)
        begin
          address_bank_decode_d1 <= address_bank_decode;
        end
    end
end
begin
  if ((address_decode[0] == 1) & (slave_read == 1))
  begin
    user_datain_0_d1 <= user_datain_0;
  end
if ((address_decode[1] == 1) & (slave_read == 1))
begin
  user_datain_1_d1 <= user_datain_1;
end
if ((address_decode[2] == 1) & (slave_read == 1))
begin
  user_datain_2_d1 <= user_datain_2;
end
if ((address_decode[3] == 1) & (slave_read == 1))
begin
  user_datain_3_d1 <= user_datain_3;
end
if ((address_decode[4] == 1) & (slave_read == 1))
begin
  user_datain_4_d1 <= user_datain_4;
end
if ((address_decode[5] == 1) & (slave_read == 1))
begin
  user_datain_5_d1 <= user_datain_5;
end
if ((address_decode[6] == 1) & (slave_read == 1))
begin
  user_datain_6_d1 <= user_datain_6;
end
if ((address_decode[7] == 1) & (slave_read == 1))
begin
  user_datain_7_d1 <= user_datain_7;
end
if ((address_decode[8] == 1) & (slave_read == 1))
begin
  user_datain_8_d1 <= user_datain_8;
end
if ((address_decode[9] == 1) & (slave_read == 1))
begin
  user_datain_9_d1 <= user_datain_9;
end
if ((address_decode[10] == 1) & (slave_read == 1))
begin
  user_datain_10_d1 <= user_datain_10;
end
if ((address_decode[11] == 1) & (slave_read == 1))
begin
  user_datain_11_d1 <= user_datain_11;
end
if ((address_decode[12] == 1) & (slave_read == 1))
begin
  user_datain_12_d1 <= user_datain_12;
end
if ((address_decode[13] == 1) & (slave_read == 1))
begin
  user_datain_13_d1 <= user_datain_13;
end
if ((address_decode[14] == 1) & (slave_read == 1))
begin
  user_datain_14_d1 <= user_datain_14;
end
if ((address_decode[15] == 1) & (slave_read == 1))
begin
  user_datain_15_d1 <= user_datain_15;
end
/\* Instantiate interrupt logic according to port type chosen and irq_en \*/

generate
if ((IRQ_EN == 1) && ((MODE_0 == 1) || (MODE_0 == 2)))
begin
    interrupt_logic interrupt_0 (clk, reset, user_datain_0, slave_write, slave_writedata
        , address_decode[0], irq_mask_reg_en, edge_capture_reg_en, readdata_0, irq_out
        [0]);
    defparam interrupt_0.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_0 = 0;
    assign irq_out[0] = 1'b0;
end
end

if ((IRQ_EN == 1) && ((MODE_1 == 1) || (MODE_1 == 2)))
begin
    interrupt_logic interrupt_1 (clk, reset, user_datain_1, slave_write, slave_writedata
        , address_decode[1], irq_mask_reg_en, edge_capture_reg_en, readdata_1, irq_out
        [1]);
    defparam interrupt_1.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_1 = 0;
    assign irq_out[1] = 1'b0;
end
end

if ((IRQ_EN == 1) && ((MODE_2 == 1) || (MODE_2 == 2)))
begin
    interrupt_logic interrupt_2 (clk, reset, user_datain_2, slave_write, slave_writedata
        , address_decode[2], irq_mask_reg_en, edge_capture_reg_en, readdata_2, irq_out
        [2]);
    defparam interrupt_2.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_2 = 0;
    assign irq_out[2] = 1'b0;
end
end

if ((IRQ_EN == 1) && ((MODE_3 == 1) || (MODE_3 == 2)))
begin
    interrupt_logic interrupt_3 (clk, reset, user_datain_3, slave_write, slave_writedata
        , address_decode[3], irq_mask_reg_en, edge_capture_reg_en, readdata_3, irq_out
        [3]);
    defparam interrupt_3.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_3 = 0;
    assign irq_out[3] = 1'b0;
end
end

if ((IRQ_EN == 1) && ((MODE_4 == 1) || (MODE_4 == 2)))
begin
    interrupt_logic interrupt_4 (clk, reset, user_datain_4, slave_write, slave_writedata
        , address_decode[4], irq_mask_reg_en, edge_capture_reg_en, readdata_4, irq_out
        [4]);
    defparam interrupt_4.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_4 = 0;
    assign irq_out[4] = 1'b0;
end
end

if ((IRQ_EN == 1) && ((MODE_5 == 1) || (MODE_5 == 2)))
begin
interrupt_logic interrupt_5 (clk, reset, user_datain_5, slave_write, slave_writedata ,
address_decode[5], irq_mask_reg_en, edge_capture_reg_en, readdata_5, irq_out [5]);
defparam interrupt_5.DATA_WIDTH = DATA_WIDTH;
end
else
begin
assign readdata_5 = 0;
assign irq_out[5] = 1'b0;
end
if ((IRQ_EN == 1) && ((MODE_6 == 1) || (MODE_6 == 2)))
begin
interrupt_logic interrupt_6 (clk, reset, user_datain_6, slave_write, slave_writedata ,
address_decode[6], irq_mask_reg_en, edge_capture_reg_en, readdata_6, irq_out [6]);
defparam interrupt_6.DATA_WIDTH = DATA_WIDTH;
end
else
begin
assign readdata_6 = 0;
assign irq_out[6] = 1'b0;
end
if ((IRQ_EN == 1) && ((MODE_7 == 1) || (MODE_7 == 2)))
begin
interrupt_logic interrupt_7 (clk, reset, user_datain_7, slave_write, slave_writedata ,
address_decode[7], irq_mask_reg_en, edge_capture_reg_en, readdata_7, irq_out [7]);
defparam interrupt_7.DATA_WIDTH = DATA_WIDTH;
end
else
begin
assign readdata_7 = 0;
assign irq_out[7] = 1'b0;
end
if ((IRQ_EN == 1) && ((MODE_8 == 1) || (MODE_8 == 2)))
begin
interrupt_logic interrupt_8 (clk, reset, user_datain_8, slave_write, slave_writedata ,
address_decode[8], irq_mask_reg_en, edge_capture_reg_en, readdata_8, irq_out [8]);
defparam interrupt_8.DATA_WIDTH = DATA_WIDTH;
end
else
begin
assign readdata_8 = 0;
assign irq_out[8] = 1'b0;
end
if ((IRQ_EN == 1) && ((MODE_9 == 1) || (MODE_9 == 2)))
begin
interrupt_logic interrupt_9 (clk, reset, user_datain_9, slave_write, slave_writedata ,
address_decode[9], irq_mask_reg_en, edge_capture_reg_en, readdata_9, irq_out [9]);
defparam interrupt_9.DATA_WIDTH = DATA_WIDTH;
end
else
begin
assign readdata_9 = 0;
assign irq_out[9] = 1'b0;
end
if ((IRQ_EN == 1) && ((MODE_10 == 1) || (MODE_10 == 2)))
begin
interrupt_logic interrupt_10 (clk, reset, user_datain_10, slave_write, slave_writedata ,
address_decode[10], irq_mask_reg_en, edge_capture_reg_en,
readdata_{10}, irq_{out}[10]);
defparam interrupt_{10}.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_{10} = 0;
    assign irq_{out}[10] = 1'b0;
end

if ((IRQ_EN == 1) && ((MODE_{11} == 1) || (MODE_{11} == 2)))
begin
    interrupt_logic interrupt_{11} (clk, reset, user_datain_{11}, slave_write,
        slave_writedata, address_decode[11], irq_mask_reg_en, edge_capture_reg_en,
        readdata_{11}, irq_{out}[11]);
defparam interrupt_{11}.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_{11} = 0;
    assign irq_{out}[11] = 1'b0;
end

if ((IRQ_EN == 1) && ((MODE_{12} == 1) || (MODE_{12} == 2)))
begin
    interrupt_logic interrupt_{12} (clk, reset, user_datain_{12}, slave_write,
        slave_writedata, address_decode[12], irq_mask_reg_en, edge_capture_reg_en,
        readdata_{12}, irq_{out}[12]);
defparam interrupt_{12}.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_{12} = 0;
    assign irq_{out}[12] = 1'b0;
end

if ((IRQ_EN == 1) && ((MODE_{13} == 1) || (MODE_{13} == 2)))
begin
    interrupt_logic interrupt_{13} (clk, reset, user_datain_{13}, slave_write,
        slave_writedata, address_decode[13], irq_mask_reg_en, edge_capture_reg_en,
        readdata_{13}, irq_{out}[13]);
defparam interrupt_{13}.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_{13} = 0;
    assign irq_{out}[13] = 1'b0;
end

if ((IRQ_EN == 1) && ((MODE_{14} == 1) || (MODE_{14} == 2)))
begin
    interrupt_logic interrupt_{14} (clk, reset, user_datain_{14}, slave_write,
        slave_writedata, address_decode[14], irq_mask_reg_en, edge_capture_reg_en,
        readdata_{14}, irq_{out}[14]);
defparam interrupt_{14}.DATA_WIDTH = DATA_WIDTH;
end
else
begin
    assign readdata_{14} = 0;
    assign irq_{out}[14] = 1'b0;
end

if ((IRQ_EN == 1) && ((MODE_{15} == 1) || (MODE_{15} == 2)))
begin
    interrupt_logic interrupt_{15} (clk, reset, user_datain_{15}, slave_write,
        slave_writedata, address_decode[15], irq_mask_reg_en, edge_capture_reg_en,
        readdata_{15}, irq_{out}[15]);
defparam interrupt_{15}.DATA_WIDTH = DATA_WIDTH;
end
else begin
    assign readdata_{15} = 0;
    assign irq_out[15] = 1'b0;
end
endgenerate

// sixtet output registers which use byteenables to register each byte. Disabling the write enable for output registers that were not needed.

register_with_bytelanes register_0 (clk, reset, slave_writedata , slave_write&address_decode [0]&user_dataout_reg_en, internal_byteenable , user_dataout_0);
defparam register_0 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_1 (clk, reset, slave_writedata , slave_write&address_decode [1]&user_dataout_reg_en, internal_byteenable , user_dataout_1);
defparam register_1 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_2 (clk, reset, slave_writedata , slave_write&address_decode [2]&user_dataout_reg_en, internal_byteenable , user_dataout_2);
defparam register_2 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_3 (clk, reset, slave_writedata , slave_write&address_decode [3]&user_dataout_reg_en, internal_byteenable , user_dataout_3);
defparam register_3 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_4 (clk, reset, slave_writedata , slave_write&address_decode [4]&user_dataout_reg_en, internal_byteenable , user_dataout_4);
defparam register_4 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_5 (clk, reset, slave_writedata , slave_write&address_decode [5]&user_dataout_reg_en, internal_byteenable , user_dataout_5);
defparam register_5 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_6 (clk, reset, slave_writedata , slave_write&address_decode [6]&user_dataout_reg_en, internal_byteenable , user_dataout_6);
defparam register_6 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_7 (clk, reset, slave_writedata , slave_write&address_decode [7]&user_dataout_reg_en, internal_byteenable , user_dataout_7);
defparam register_7 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_8 (clk, reset, slave_writedata , slave_write&address_decode [8]&user_dataout_reg_en, internal_byteenable , user_dataout_8);
defparam register_8 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_9 (clk, reset, slave_writedata , slave_write&address_decode [9]&user_dataout_reg_en, internal_byteenable , user_dataout_9);
defparam register_9 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_10 (clk, reset, slave_writedata , slave_write&address_decode[10]&user_dataout_reg_en, internal_byteenable , user_dataout_10);
defparam register_10 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_11 (clk, reset, slave_writedata , slave_write&address_decode[11]&user_dataout_reg_en, internal_byteenable , user_dataout_11);
defparam register_11 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_12 (clk, reset, slave_writedata , slave_write&address_decode[12]&user_dataout_reg_en, internal_byteenable , user_dataout_12);
defparam register_12 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_13 (clk, reset, slave_writedata , slave_write&address_decode[13]&user_dataout_reg_en, internal_byteenable , user_dataout_13);
defparam register_13 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_14 (clk, reset, slave_writedata , slave_write&address_decode[14]&user_dataout_reg_en, internal_byteenable , user_dataout_14);
defparam register_14 .DATA_WIDTH = DATA_WIDTH;
register_with_bytelanes register_15 (clk, reset, slave_writedata , slave_write&address_decode[15]&user_dataout_reg_en, internal_byteenable , user_dataout_15);
defparam register_15 .DATA_WIDTH = DATA_WIDTH;

// registered slave_readdata mux for all the return values, if an input register is disabled then a zero will be passed in
always @(posedge clk or posedge reset) begin
  if (reset == 1) begin
    read_from_user <= 0;
  end
else
begin
if (slave_read_d1 == 1) // first multiplexer stage
begin
// when the mode parameters are set to 3 then the outputs are looped back as inputs

case (address_decode_d1[3:0])
4'b0001: mux_first_stage_a <= (MODE0 == 3)? user_dataout_0 : user_datain_0.d1;
4'b0010: mux_first_stage_a <= (MODE1 == 3)? user_dataout_1 : user_datain_1.d1;
4'b0100: mux_first_stage_a <= (MODE2 == 3)? user_dataout_2 : user_datain_2.d1;
4'b1000: mux_first_stage_a <= (MODE3 == 3)? user_dataout_3 : user_datain_3.d1;
endcase

case (address_decode_d1[7:4])
4'b0001: mux_first_stage_b <= (MODE4 == 3)? user_dataout_4 : user_datain_4.d1;
4'b0010: mux_first_stage_b <= (MODE5 == 3)? user_dataout_5 : user_datain_5.d1;
4'b0100: mux_first_stage_b <= (MODE6 == 3)? user_dataout_6 : user_datain_6.d1;
4'b1000: mux_first_stage_b <= (MODE7 == 3)? user_dataout_7 : user_datain_7.d1;
endcase

case (address_decode_d1[11:8])
4'b0001: mux_first_stage_c <= (MODE8 == 3)? user_dataout_8 : user_datain_8.d1;
4'b0010: mux_first_stage_c <= (MODE9 == 3)? user_dataout_9 : user_datain_9.d1;
4'b0100: mux_first_stage_c <= (MODE10 == 3)? user_dataout_10 : user_datain_10.d1;
4'b1000: mux_first_stage_c <= (MODE11 == 3)? user_dataout_11 : user_datain_11.d1;
endcase

case (address_decode_d1[15:12])
4'b0001: mux_first_stage_d <= (MODE12 == 3)? user_dataout_12 : user_datain_12.d1;
4'b0010: mux_first_stage_d <= (MODE13 == 3)? user_dataout_13 : user_datain_13.d1;
4'b0100: mux_first_stage_d <= (MODE14 == 3)? user_dataout_14 : user_datain_14.d1;
4'b1000: mux_first_stage_d <= (MODE15 == 3)? user_dataout_15 : user_datain_15.d1;
endcase

if (slave_read_d2 == 1) // second multiplexer stage
begin

case (address_bank_decode_d1[3:0])
4'b0001: read_from_user <= mux_first_stage_a;
4'b0010: read_from_user <= mux_first_stage_b;
4'b0100: read_from_user <= mux_first_stage_c;
4'b1000: read_from_user <= mux_first_stage_d;
endcase
end
end

// multiplex the readdata from all I/O interrupt registers
always @(posedge clk or posedge reset)
begin
if (reset == 1)
begin
read_from_int_regs <= 0;
end
else
begin
if (slave_read_d1 == 1) // first multiplexer stage
begin
// when the mode parameters are set to 3 then the outputs are looped back as inputs

case (address_decode_d1[3:0])
4'b0001: int_mux_first_stage_a <= readdata_0;
4'b0010: int_mux_first_stage_a <= readdata_1;
4'b0100: int_mux_first_stage_a <= readdata_2;
4'b1000: int_mux_first_stage_a <= readdata_3;
endcase
end
end
end
//
endcase

case (address_decode_d1[7:4])
    4'b0001: int_mux_first_stage_b <= readdata_4;
    4'b0010: int_mux_first_stage_b <= readdata_5;
    4'b0100: int_mux_first_stage_b <= readdata_6;
    4'b1000: int_mux_first_stage_b <= readdata_7;
endcase

case (address_decode_d1[11:8])
    4'b0001: int_mux_first_stage_c <= readdata_8;
    4'b0010: int_mux_first_stage_c <= readdata_9;
    4'b0100: int_mux_first_stage_c <= readdata_10;
    4'b1000: int_mux_first_stage_c <= readdata_11;
endcase

case (address_decode_d1[15:12])
    4'b0001: int_mux_first_stage_d <= readdata_12;
    4'b0010: int_mux_first_stage_d <= readdata_13;
    4'b0100: int_mux_first_stage_d <= readdata_14;
    4'b1000: int_mux_first_stage_d <= readdata_15;
endcase
end

if (slave_read_d2 == 1) // second multiplexer stage
begin
    case (address_bank_decode_d1[3:0])
        4'b0001: read_from_int_regs <= int_mux_first_stage_a;
        4'b0010: read_from_int_regs <= int_mux_first_stage_b;
        4'b0100: read_from_int_regs <= int_mux_first_stage_c;
        4'b1000: read_from_int_regs <= int_mux_first_stage_d;
    endcase
end
end

assign priority_int_src = (irq_out[0] == 1) ? 1:
    (irq_out[1] == 1) ? 2 :
    (irq_out[2] == 1) ? 3 :
    (irq_out[3] == 1) ? 4 :
    (irq_out[4] == 1) ? 5 :
    (irq_out[5] == 1) ? 6 :
    (irq_out[6] == 1) ? 7 :
    (irq_out[7] == 1) ? 8 :
    (irq_out[8] == 1) ? 9 :
    (irq_out[9] == 1) ? 10 :
    (irq_out[10] == 1) ? 11 :
    (irq_out[11] == 1) ? 12 :
    (irq_out[12] == 1) ? 13 :
    (irq_out[13] == 1) ? 14 :
    (irq_out[14] == 1) ? 15 :
    (irq_out[15] == 1) ? 16 : 0;

assign user_write = slave_write_d1; // outputs are registed so need a delayed copy of the write signal
assign user_read = slave_read;
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
assign user_byteenable = (slave_write_d1 == 1)? internal_byteenable_d1 : internal_byteenable; // for write cycles need the delayed copy of the byteenables , don't use the byteenables for reads since the full word is always sent on read transfers
assign slave_readdata = (slave_address == 9'h100)? priority_int_src : (user_datain_reg_en == 1)? read_from_user : read_from_int_regs;
assign slave_irq = | (irq_out);
endmodule
module register_with_bytelanes (
  clk ,
  reset ,
  data_in ,
  write ,
  byte_enables ,
  data_out
);

parameter DATA_WIDTH = 32;

input clk ;
input reset ;

input [DATA_WIDTH−1:0] data_in ;
input write ;
input [((DATA_WIDTH/8)−1:0] byte_enables ;
output reg [DATA_WIDTH−1:0] data_out ;

// generating write logic for each group of 8 bits for 'data_out'
generate
  genvar LANE ;
  for ( LANE = 0 ; LANE < (DATA_WIDTH/8) ; LANE = LANE+1 )
    begin : register_bytelane_generation
      always @ ( posedge clk or posedge reset )
        begin
          if ( reset == 1 )
            begin
              data_out [((LANE*8)+7:(LANE*8)] <= 0 ;
            end
          else
            begin
              if ( ((byte_enables[LANE] == 1) & (write == 1))
                begin
                  data_out [((LANE*8)+7:(LANE*8)] <= data_in [((LANE*8)+7:(LANE*8)]; // write to each byte lane with write = 1 and the lane byteenable = 1
                end
            end
        end
    end
endgenerate
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)

/*******************************************************************************
 * These are all word addresses. Please shift the
 * offset accordingly when using them in IORD or IOWR.
 * E.g.: Write to a 32-bit user_datain_0 port
 * IOWR_32DIRECT(BASE, DATA_IN_0 = 4, 0XFFFFFFFF)
 * Another example: Read a 16-bit edge capture reg
 * IORD_16DIRECT(BASE, EDGE_CAPTURE_0 = 2)
*******************************************************************************/

#define DATA_IN_0 0x00
#define DATA_IN_1 0x10
#define DATA_IN_2 0x20
#define DATA_IN_3 0x30
#define DATA_IN_4 0x40
#define DATA_IN_0 0x50
#define DATA_IN_1 0x60
#define DATA_IN_2 0x70
#define DATA_IN_3 0x80
#define DATA_IN_4 0x90
#define DATA_IN_5 0xA0
#define DATA_IN_6 0xB0
#define DATA_IN_7 0xC0
#define DATA_IN_8 0xD0
#define DATA_IN_9 0xE0
#define DATA_IN_10 0xF0

#define DATA_OUT_0 0x01
#define DATA_OUT_1 0x11
#define DATA_OUT_2 0x21
#define DATA_OUT_3 0x31
#define DATA_OUT_4 0x41
#define DATA_OUT_5 0x51
#define DATA_OUT_6 0x61
#define DATA_OUT_7 0x71
#define DATA_OUT_8 0x81
#define DATA_OUT_9 0x91
#define DATA_OUT_10 0xA1
#define DATA_OUT_11 0xB1
#define DATA_OUT_12 0xC1
#define DATA_OUT_13 0xD1
#define DATA_OUT_14 0xE1
#define DATA_OUT_15 0xF1

#define IRQ_MASK_0 0x02
#define IRQ_MASK_1 0x12
#define IRQ_MASK_2 0x22
#define IRQ_MASK_3 0x32
#define IRQ_MASK_4 0x42
#define IRQ_MASK_5 0x52
#define IRQ_MASK_6 0x62
#define IRQ_MASK_7 0x72
#define IRQ_MASK_8 0x82
#define IRQ_MASK_9 0x92
#define IRQ_MASK_10 0xA2
#define IRQ_MASK_11 0xB2
#define IRQ_MASK_12 0xC2
#define IRQ_MASK_13 0xD2
#define IRQ_MASK_14 0xE2
#define IRQ_MASK_15 0xF2

#define EDGE_CAPTURE_0 0x03
#define EDGE_CAPTURE_1 0x13
#define EDGE_CAPTURE_2 0x23
#define EDGE_CAPTURE_3 0x33
#define EDGE_CAPTURE_4 0x43
#define EDGE_CAPTURE_5 0x53
#define EDGE_CAPTURE_6 0x63
#define EDGE_CAPTURE_7 0x73
#define EDGE_CAPTURE_8 0x83
#define EDGE_CAPTURE_9 0x93
#define EDGE_CAPTURE_10 0xA3
#define EDGE_CAPTURE_11 0xB3
#define EDGE_CAPTURE_12 0xC3
#define EDGE_CAPTURE_13 0xD3
#define EDGE_CAPTURE_14 0xE3
#define EDGE_CAPTURE_15 0xF3

#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
      ├── build
      │    ├── <FILES>
      │    └── 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-iptest
          └── files
              └── test.c
                  └── smart-cam-iptest_1.0.bb

meta-yocto
  └── <FILES>

meta-yocto-bsp
  └── <FILES>

scripts
  └── <FILES>

altera-init
  └── LICENSE

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

```bash
DESCRIPTION = "Software to Test ARM / FPGA Interface and Shared DDR3"
SECTION = "smart-cam-iptest"
LICENSE = "BSD"
LIC_FILES_CHKSUM = "file ://${COMMON_LICENSE_DIR}/BSD ; md5=3775480a712fc46a69647678acb234cb"
PR = "r0"
FILES_${PN} = "/home/root/snr-des/testIP/*"
FILES_${PN}-dbg = "/usr/home/root/snr-des/testIP/.debug"
SRC_URI = "file ://test.c"
S = "${WORKDIR}" 
#based on http://www.multitech.net/developer/products/multiconnect-ocg/development/writing-bitbake-recipes/

do_compile () {
  $(CC) $(CFLAGS) $(LDFLAGS) test.c -o test
}

do_install () {
  install -d $(D)/home/root/snr-des/testIP
  install -m 0755 test $(D)/home/root/snr-des/testIP/test
}
```

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

16.4.3 Senior Design Test Application Source (test.c)

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <stdbool.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/mman.h>

#define DDR3_ARM_BASE 0xc0000000
#define DDR3_ARM_END 0xcf000000
#define DDR3_IP_BASE 0x00000000
#define DDR3_IP_END 0x0fffffff

//the linux image uses pages of width 4 KB or 0x00001000
//as it turns out, this is the mem space of the entire bridge
//used to connect to the CSR. Since we need to map an entire
//page, we need to take 0xff23000 to 0xff230fff.
//However, the real CSR starts at 0xff23800
#define IP_CSR_OFFSET 0x00000800
#define IP_CSR_BASE 0xff230000
#define IP_CSR_END 0xff230fff

//based on readme_first.txt from the altera template
//each register is the 2nd hexideximal digit (16 place)
//the least significan hex digit represents the access
```

215
// mode such as 0 = read, 1 = write, 2 = interrupt mask
#endif
#define ADDR_OFFSET 0x01 // address (write) (0 = read, 1 = write)
#define MODE_OFFSET 0x11 // mode (write) (0 = read, 1 = write)
#define DATA0W_OFFSET 0x21 // data0 (write)
#define DATA1W_OFFSET 0x31 // data1 (write)
#define DATA2W_OFFSET 0x41 // data2 (write)
#define DATA3W_OFFSET 0x51 // data3 (write)
#define READ_RDY_OFFSET 0x00 // read ready (read) (0 when ready)
#define WRITE_RDY_OFFSET 0x10 // write ready (read) (0 when ready)
#define DATA0R_OFFSET 0x20 // data0 (read)
#define DATA1R_OFFSET 0x30 // data1 (read)
#define DATA2R_OFFSET 0x40 // data2 (read)
#define DATA3R_OFFSET 0x50 // data3 (read)

// Constants used by IP CSR
#define READ_MODE 0
#define WRITE_MODE 1
#define IP_READY 0

// info on how to use open and mmap on /dev/mem was based on http://stackoverflow.com/questions/962193/how-to-access-kernel-space-from-user-space-in-linux
// o_sync is essential to ensure writes to and reads from /dev/mem are not buffered by the OS. Bitwise oring is used to specify multiple parms

int main(int argc, char** argv)
{
    int memSize = DDR3_ARM_END - DDR3_ARM_BASE + 1;
    int csrSize = IP_CSR_END - IP_CSR_BASE + 1;
    int status;
    bool testPass;
    int test0, test1, test2, test3;
    int d0 = 0;
    int d1 = 0;
    int d2 = 0;
    int d3 = 0;
    int memOffset = 0;
    int exitStatus = EXIT_SUCCESS;
    int* ddr3Map;
    int* csrMap;
    int memFile = open("/dev/mem", O_RDWR | O_SYNC);

    if (memFile == -1)
    {
        int errNum = errno;
        char* errorMsg = strerror(errNum);
        printf("Error opening /dev/mem: %s\n", errorMsg);
        return EXIT_FAILURE;
    }
    ddr3Map = (int*) mmap(NULL, memSize, PROT_READ | PROT_WRITE, MAP_SHARED, memFile, DDR3_ARM_BASE);
    if (ddr3Map == MAP_FAILED)
    {
        int errNum = errno;
        char* errorMsg = strerror(errNum);
        printf("Error mapping DDR3: %s\n", errorMsg);
        return EXIT_FAILURE;
    }
}
csrMap = (int *) mmap(NULL, csrSize, PROT_READ | PROT_WRITE, MAP_SHARED, memFile,
   IP_CSR_BASE);

if (csrMap == MAP_FAILED) {
   int errNum = errno;
   char * errorMsg = strerror(errNum);
   printf("Error mapping IP CSR: %s\n", errorMsg);
   return EXIT_FAILURE;
}

// ********** Begin Test Code **********
// Read Test

test0 = 0x0000000F;
test1 = 0x000000F0;
test2 = 0x00000F00;
test3 = 0x0000F000;

*(ddr3Map + memOffset) = test0;
*(ddr3Map + memOffset+1) = test1;
*(ddr3Map + memOffset+2) = test2;
*(ddr3Map + memOffset+3) = test3;

printf("Wrote into DDR3 from ARM\n");
printf("0x %x\n", test0);
printf("0x %x\n", test1);
printf("0x %x\n", test2);
printf("0x %x\n", test3);

printf("\n");

// read using IP
*(csrMap + IP_CSR_OFFSET + ADDR_OFFSET) = DDR3_IP_BASE+memOffset;
*(csrMap + IP_CSR_OFFSET + MODE_OFFSET) = READMODE;

while ((*(csrMap + IP_CSR_OFFSET + READ_RDY_OFFSET) & 1) != IP_READY) {
   ; // wait for IP to finish read operation
   // this is a spin lock, I know, but this is a simple test of IP
   // and not a high speed application
}

d0 = *(csrMap + IP_CSR_OFFSET + DATA0R_OFFSET);
d1 = *(csrMap + IP_CSR_OFFSET + DATA1R_OFFSET);
d2 = *(csrMap + IP_CSR_OFFSET + DATA2R_OFFSET);
d3 = *(csrMap + IP_CSR_OFFSET + DATA3R_OFFSET);

printf("Read from DDR3 in IP\n");
printf("0x%lx\n", d0);
printf("0x%lx\n", d1);
printf("0x%lx\n", d2);
printf("0x%lx\n", d3);

testPass = test0==d0 && test1==d1 && test2==d2 && test3==d3;

if (testPass){
   printf("***** Read Test: Success *****\n");
} else {
   printf("***** Read Test: Fail *****\n");
}

// Write test
memOffset = 0x00000010;

test0 = 0x11111111;
test1 = 0x22222222;
test2 = 0x33333333;
test3 = 0x44444444;
*(csrMap + IP_CSR_OFFSET + ADDR_OFFSET) = DDR3_IP_BASE+memOffset;
*(csrMap + IP_CSR_OFFSET + DATA0W_OFFSET) = test0;
*(csrMap + IP_CSR_OFFSET + DATA1W_OFFSET) = test1;
*(csrMap + IP_CSR_OFFSET + DATA2W_OFFSET) = test2;
*(csrMap + IP_CSR_OFFSET + DATA3W_OFFSET) = test3;
*(csrMap + IP_CSR_OFFSET + MODE_OFFSET) = WRITE_MODE;
while ((*(csrMap + IP_CSR_OFFSET + WRITE_RDY_OFFSET) & 1) != IP_READY) {
    // wait for IP to finish write operation
    // this is a spin lock, I know, but this is a simple test of IP
    // and not a high speed application
}
printf("Wrote into DDR3 from IP\n");
printf("0x %x\n", test0);
printf("0x %x\n", test1);
printf("0x %x\n", test2);
printf("0x %x\n", test3);
printf("\n");
d0 = *(ddr3Map + memOffset);
d1 = *(ddr3Map + memOffset + 1);
d2 = *(ddr3Map + memOffset + 2);
d3 = *(ddr3Map + memOffset + 3);
printf("Read from DDR3 in ARM\n");
printf("0x%x\n", d0);
printf("0x%x\n", d1);
printf("0x%x\n", d2);
printf("0x%x\n", d3);
testPass = test0==d0 && test1==d1 && test2==d2 && test3==d3;
if (testPass) {
    printf("***** Write Test: Success *****");
} else {
    printf("***** Write Test: Fail *****");
}

/********** End Test Code ***********/
status = munmap(ddr3Map, memSize);
if (status == -1) {
    int errNum = errno;
    char* errorMsg = strerror(errNum);
    printf("Error un-mapping DDR3: %s\n", errorMsg);
    exitStatus = EXIT_FAILURE;
}
status = munmap(csrMap, csrSize);
if (status == -1) {
    int errNum = errno;
    char* errorMsg = strerror(errNum);
    printf("Error un-mapping CSR: %s\n", errorMsg);
    exitStatus = EXIT_FAILURE;
}
status = close(memFile);
if (status == -1) {
    int errNum = errno;
    char* errorMsg = strerror(errNum);
}
<table>
<thead>
<tr>
<th>Listing 16.19: Senior Design Test Application Source (test.c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>227 printf(&quot;Error closing /dev/mem: %s\n&quot;, errorMsg);</td>
</tr>
<tr>
<td>228 exitStatus = EXIT_FAILURE;</td>
</tr>
<tr>
<td>229 return exitStatus;</td>
</tr>
</tbody>
</table>
|}

16.4.4 Senior Design Layer Configuration File (layer.conf)

```plaintext
# Based on OE Layer.conf file
# We have a conf and classes directory, append to BBPATH
BBPATH := "${{LAYERDIR}}"

# We have a recipes directory, add to BBFILES
BBFILES += "${{LAYERDIR}}/recipes -*/*.bb ${{LAYERDIR}}/recipes -*/*/*.bbappend"

BBFILE_COLLECTIONS += "snr-des-layer"
BBFILE_PATTERN_snr-des-layer := "^${{LAYERDIR}}/"

# Define the priority for recipes (.bb files) from this layer, choosing carefully how this layer interacts with all of the other layers.
BBFILE_PRIORITY_openembedded-layer = "1"
```

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

16.4.5 Modified bblayers File (bblayers.conf)

```plaintext
# LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf changes incompatibly
LCONF_VERSION = "6"

BBPATH = "${{TOPDIR}}"
BBFILES ?? = ""
BBLAYERS = "% 
/home/christopher/yocto-ghrd/meta-snr-des 
/home/christopher/yocto-ghrd/meta-oe 
/home/christopher/yocto-ghrd/meta 
/home/christopher/yocto-ghrd/meta-yocto 
/home/christopher/yocto-ghrd/meta-yocto-bsp 
/home/christopher/yocto-ghrd/meta-yocto-bsp 
/home/christopher/yocto-ghrd/meta-linaro 
/home/christopher/yocto-ghrd/meta-altera 
"
```

Listing 16.21: Modified bblayers File (bblayers.conf)

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

```plaintext
DESCRIPTION = "The set of packages for development and testing provided by Altera"
LICENSE = "MIT"
```

219

IMAGE_INSTALL="$(ALTERA_IMAGE_INSTALL)"

inherit core

# altera-image.inc must be included after inherit core-image to override functionality
include altera-image.inc

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

### 16.4.7 Modified Device Tree File (socfpga.dts)

```dts
/*
 * This devicetree is generated by sopc2dts on Sat May 03 22:38:06 PDT 2014
 * Sopc2dts is written by Walter Goossens <waltergoossens@home.nl>
 * in cooperation with the nios2 community <Nios2-dev@sopc.et.ntust.edu.tw>
 */
/dts-v1/;
{
  model = "ALTR, socfpga-cyclone5";
  compatible = "ALTR, socfpga-cyclone5";
  #address-cells = < 1 >;
  #size-cells = < 1 >;
  aliases {
    ethernet0 = "/sopc/ethernet@0xff702000";
    serial0 = "/sopc/serial@0xffc02000";
    serial1 = "/sopc/serial@0xffc03000";
    timer0 = "/sopc/timer@0xffc08000";
    timer1 = "/sopc/timer@0xffc09000";
    timer2 = "/sopc/timer@0xffd00000";
    timer3 = "/sopc/timer@0xffd01000";
  }; // end aliases

cpus {
  #address-cells = < 1 >;
  #size-cells = < 0 >;
  hps_0_arm_a9_0: cpu@0x0 {
    device_type = "cpu";
    compatible = "arm,cortex-a9-1.0", "arm,cortex-a9";
    reg = < 0x00000000 >;
  }; // end cpu@0x0 (hps_0_arm_a9_0)
  hps_0_arm_a9_1: cpu@0x1 {
    device_type = "cpu";
    compatible = "arm,cortex-a9-1.0", "arm,cortex-a9";
    reg = < 0x00000001 >;
  }; // end cpu@0x1 (hps_0_arm_a9_1)
  }; // end cpus

cpus {
  device_type = "memory";
  reg = < 0xC0000000 0x00000000 0xFFF00000 0x00010000 >;
  }; // end memory@0

sopc@0 {
```

220
device_type = "soc";
ranges;
#address-cells = < 1 >;
#size-cells = < 1 >;
compatible = "ALTR,avalon", "simple-bus";
buss-frequency = < 50000000 >;

hps_0,h2f,lw: bridge@0xff200000 {
    compatible = "altr,h2f,lw_bridge-1.0", "simple-bus";
    reg = < 0x00010040 0x00000020 >;
    address-cells = < 1 >;
    size-cells = < 1 >;
    ranges = < 0x00010040 0x00010080 0x000100c0 0x00010040 0x00010080 0x000100c0 0x00020000 0x00030000 >
}

ledpio: gpio@0x10040 {
    compatible = "ALTR,pio-13.0.1.99.2", "ALTR,pio-1.0", "altr,pio-1.0";
    reg = < 0x00010040 0x00010080 0x000100c0 0x00010080 0x000100c0 0x00020000 0x00030000 >
    gpio-cells = < 2 >;
    gpio-controller;
}

dipswpio: gpio@0x10080 {
    compatible = "ALTR,pio-13.0.1.99.2", "ALTR,pio-1.0", "altr,pio-1.0";
    reg = < 0x00010040 0x00010080 0x000100c0 0x000100c0 0x00020000 0x00030000 >
    gpio-cells = < 2 >;
    gpio-controller;
}

buttonpio: gpio@0x100c0 {
    compatible = "ALTR,pio-13.0.1.99.2", "ALTR,pio-1.0", "altr,pio-1.0";
    reg = < 0x00010040 0x00010080 0x000100c0 0x00020000 0x00030000 >
    gpio-cells = < 2 >;
    gpio-controller;
}

jtag_uart: serial@0x200000 {
    compatible = "ALTR,avalon-13.0", "simple-bus";
    reg = < 0x00030000 0x00001000 >;
    address-cells = < 1 >;
    size-cells = < 1 >;
    ranges = < 0x000000800 0x00030000 0x000000800 >;
}

mm_clock_crossing_bridge_1: bridge@0x300000 {
    compatible = "ALTR,avalon", "simple-bus";
    reg = < 0x00030000 0x00001000 >;
    address-cells = < 1 >;
    size-cells = < 1 >;
    ranges = < 0x00030000 0x000000800 >;
}
IPTest_0: unknown@0x800 {
    compatible = "unknown.unknown-1.0", "simple-bus";
    reg = < 0x00000000 0x00000800 >;
}; //end unknown@0x800 (IPTest_0)
}; //end bridge@0x30000 (mm_clock_crossing_bridge_1)
}; //end bridge@0x0ff200000 (hps_0_h2f_lw)

hps_0_arm_gic_0: intc@0xfffed000 {
    compatible = "arm, cortex-a9-gic-1.0", "arm, cortex-a9-gic";
    reg = < 0xFFFFE0000 0x00001000 0xFFFFEC100 0x00000100 >;
    interrupt-controller:
    #interrupt-cells = < 3 >;
}; //end intc@0xfffed000 (hps_0_arm_gic_0)

hps_0_L2: L2-cache@0xfffe0000 {
    compatible = "arm.pl310-cache-1.0", "arm.pl310-cache";
    reg = < 0xFFFEEF0000 0x00001000 >;
    interrupt-parent = < &hps_0_arm_gic_0 >;
    interrupts = < 0 38 4 >;
    cache-level = < 2 >; /* embeddedsw.dts.params.cache-level type NUMBER */
    cache-unified; /* appended from boardinfo */
    arm.setTagLatency = < 1 1 >;
}; //end L2-cache@0xfffe0000 (hps_0_L2)

hps_0_dma: dma@0xffe01000 {
    compatible = "arm.pl330-1.0", "arm.pl330", "arm,primecell";
    reg = < 0xFFFFE01000 0x00001000 >;
    interrupt-parent = < &hps_0_dma >;
    interrupts = < 0 104 4 >;
}; //end dma@0xffe01000 (hps_0_dma)

hps_0_sysmgr: sysmgr@0xffd08000 {
    compatible = "altr, sys-mgr-1.0", "altr, sys-mgr";
    reg = < 0xFFFFD08000 0x00004000 >;
}; //end sysmgr@0xffd08000 (hps_0_sysmgr)

hps_0_clkmgr: clkmgr@0xffd04000 {
    compatible = "altr, clkt-mgr-1.0", "altr, clkt-mgr";
    reg = < 0xFFFFD04000 0x00001000 >;
}; //end clkmgr@0xffd04000 (hps_0_clkmgr)

hps_0_rstmgr: rstmgr@0xffd05000 {
    compatible = "altr, rstmgr-1.0", "altr, rstmgr";
    reg = < 0xFFFFD05000 0x00001000 >;
}; //end rstmgr@0xffd05000 (hps_0_rstmgr)

hps_0_fpgamgr: fpgamgr@0xff706000 {
    compatible = "altr, fpga-mgr-1.0", "altr, fpga-mgr";
    reg = < 0xFF7060000 0x00001000 0xFFF90000 0x00001000 >;
    interrupt-parent = < &hps_0_arm_gic_0 >;
    interrupts = < 0 175 4 >;
    transport = "mmio"; /* embeddedsw.dts.params.transport type STRING */
}; //end fpgamgr@0xff706000 (hps_0_fpgamgr)

hps_0_uart0: serial@0xffc02000 {
    compatible = "snps,dw-apb-uart-1.0", "snps,dw-apb-uart";
    reg = < 0xFFC02000 0x00001000 >;
    interrupt-parent = < &hps_0_uart0 >;
    interrupts = < 0 162 4 >;
    reg-io-width = < 4 >; /* embeddedsw.dts.params.reg-io-width type NUMBER */
    reg-shift = < 2 >; /* embeddedsw.dts.params.reg-shift type NUMBER */
    clock-frequency = < 1000000000 >; /* appended from boardinfo */
}; //end serial@0xffc02000 (hps_0_uart0)

hps_0_uart1: serial@0xffc03000 {

compatible = "snps,dw-apb-uart-1.0", "snps,dw-apb-uart";
reg = < 0xFFC03000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 163 4 >; /* appended from boardinfo */
clock-frequency = < 100000000 >;
}; // end timer0

hps_0_timer0: timer@0xfff08000 {
compatible = "snps,dw-apb-timer-sp-1.0", "snps,dw-apb-timer-sp";
reg = < 0xFFF08000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 168 4 >;
clock-frequency = < 100000000 >; /* appended from boardinfo */
}; // end timer0

hps_0_timer1: timer@0xfff09000 {
compatible = "snps,dw-apb-timer-sp-1.0", "snps,dw-apb-timer-sp";
reg = < 0xFFF09000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 167 4 >;
clock-frequency = < 100000000 >; /* appended from boardinfo */
}; // end timer1

hps_0_timer2: timer@0xffd00000 {
compatible = "snps,dw-apb-timer-osc-1.0", "snps,dw-apb-timer-osc";
reg = < 0xFFD00000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 169 4 >;
clock-frequency = < 250000000 >; /* appended from boardinfo */
}; // end timer2

hps_0_timer3: timer@0xffd01000 {
compatible = "snps,dw-apb-timer-osc-1.0", "snps,dw-apb-timer-osc";
reg = < 0xFFD01000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 170 4 >;
clock-frequency = < 250000000 >; /* appended from boardinfo */
}; // end timer3

hps_0_gpio0: gpio@0xff708000 {
compatible = "snps,dw-gpio-1.0", "snps,dw-gpio";
reg = < 0xFF708000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 164 4 >;
#gpio-cells = < 2 >;
gpio-controller;
}; // end gpio0

hps_0_gpio1: gpio@0xff709000 {
compatible = "snps,dw-gpio-1.0", "snps,dw-gpio";
reg = < 0xFF709000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 165 4 >;
#gpio-cells = < 2 >;
gpio-controller;
}; // end gpio1

hps_0_gpio2: gpio@0xff70a000 {
compatible = "snps,dw-gpio-1.0", "snps,dw-gpio";
reg = < 0xFF70A000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 166 4 >;
#gpio-cells = < 2 >;
gpio-controller;
}; // end gpio2
hps_0_i2c0: i2c@0xffc04000 {
  compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
  reg = < 0xffc04000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 158 4 >;
  emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type NUMBER */
  #address-cells = < 1 >;
  #size-cells = < 0 >;
  speed-mode = < 0 >; /* appended from boardinfo */
}

lcd: newhaven,nhd-0216k3z-nsw-bbw@0x28 {
  compatible = "newhaven,nhd-0216k3z-nsw-bbw";
  reg = < 0x00000028 >;
  height = < 2 >; /* appended from boardinfo */
  width = < 16 >; /* appended from boardinfo */
}; //end newhaven.nhd-0216k3z-nsw-bbw@0x28 (lcd)

eeprom: atmel,24c32@0x51 {
  compatible = "atmel,24c32";
  reg = < 0x00000051 >;
  pagesize = < 32 >; /* appended from boardinfo */
}; //end atmel,24c32@0x51 (eeprom)

hps_0_i2c1: i2c@0xffc05000 {
  compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
  reg = < 0xffc05000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 159 4 >;
  emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type NUMBER */
}; //end i2c@0xffc05000 (hps_0_i2c1)

hps_0_i2c2: i2c@0xffc06000 {
  compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
  reg = < 0xffc06000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 160 4 >;
  emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type NUMBER */
}; //end i2c@0xffc06000 (hps_0_i2c2)

hps_0_i2c3: i2c@0xffc07000 {
  compatible = "snps,designware-i2c-1.0", "snps,designware-i2c";
  reg = < 0xffc07000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 161 4 >;
  emptyfifo_hold_master = < 1 >; /* embeddedsw.dts.params.emptyfifo_hold_master type NUMBER */
}; //end i2c@0xffc07000 (hps_0_i2c3)

hps_0_nand0: flash@0xff900000 {
  compatible = "denali,nand-1.0", "denali,denali-nand-dt";
  reg = < 0xff900000 0x01000000 >
  0xfff80000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 144 4 >;
  #address-cells = < 1 >; /* embeddedsw.dts.params.#address-cells type NUMBER */
  #size-cells = < 1 >; /* embeddedsw.dts.params.#size-cells type NUMBER */
  reg-names = "nand_data", "denali_reg"; /* embeddedsw.dts.params.reg-names type STRING */
  bank-width = < 2 >;
  device-width = < 1 >;
}; //end flash@0xff900000 (hps_0_nand0)

hps_0_qspi: flash@0xff705000 {
  compatible = "cadence,qspi-1.0", "cadence,qspi";
reg = < 0xFF705000 0x00001000
  0xFFA00000 0x00001000 >;
interrupt-parent = < &hps_0_arm_gic_0 >;
interrupts = < 0 151 4 >;
bus-num = < 2 >; /* embeddedsw.dts.params.bus-num type NUMBER */
fifo-depth = < 128 >; /* embeddedsw.dts.params.fifo-depth type NUMBER */
num-chipselect = < 4 >; /* embeddedsw.dts.params.num-chipselect type NUMBER */
bank-width = < 2 >;
device-width = < 1 >;
master-ref-clk = < 400000000 >; /* appended from boardinfo */
ext-decoder = < 0 >; /* appended from boardinfo */

flash0: n25q128@0 {
  #address-cells = < 1 >; /* appended from boardinfo */
  #size-cells = < 1 >; /* appended from boardinfo */
  compatible = "n25q128"; /* appended from boardinfo */
  reg = < 0 >; /* appended from boardinfo */
  spi-max-frequency = < 1000000000 >; /* appended from boardinfo */
  page-size = < 256 >; /* appended from boardinfo */
  block-size = < 16 >; /* appended from boardinfo */
  quad = < 1 >; /* appended from boardinfo */
  tshs1-ns = < 200 >; /* appended from boardinfo */
  tsd2d-ns = < 255 >; /* appended from boardinfo */
  tchsh-ns = < 20 >; /* appended from boardinfo */
  tslch-ns = < 20 >; /* appended from boardinfo */
}

part0: partition@0 {
  label = "Flash 0 Raw Data"; /* appended from boardinfo */
}; /* end partition@0 (part0) */

partl: partition@800000 {
  label = "Flash 1 JFFS2 Filesystem"; /* appended from boardinfo */
}; /* end partition@800000 (partl) */

}; /* end flash@0xff705000 (hps_0_qspi) */

hps_0_sdmcc: flash@0xff704000 {
  compatible = "snps,mmc~1.0", "snps,mmc~1.0";
  reg = < 0xFF704000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 139 4 >;
  fifo-depth = < 1024 >; /* embeddedsw.dts.params.fifo-depth type NUMBER */
  num-slots = < 1 >; /* embeddedsw.dts.params.num-slots type NUMBER */
  bank-width = < 2 >;
  device-width = < 1 >;
  bus-hz = < 12500000 >; /* appended from boardinfo */
  #address-cells = < 1 >; /* appended from boardinfo */
  #size-cells = < 0 >; /* appended from boardinfo */
  compatible = "snps,dwc~1.0", "snps,dwc~1.0";

}; /* end flash@0xff704000 (hps_0_sdmcc) */

hps_0_usb0: usb@0xffb00000 {
  compatible = "snps,dwc~1.0", "snps,dwc~1.0";
  reg = < 0xFFB00000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 125 4 >;
  dev-nperio-tx-fifo-size = < 4096 >; /* appended from boardinfo */
  num-slots = < 1 >; /* appended from boardinfo */
  compatible = "snps,dwc~1.0", "snps,dwc~1.0";

}; /* end flash@0xff704000 (hps_0_sdmcc) */
dev_rx_fifo-size = < 512 >; /* embeddedsw.dts.params.dev_rx_fifo-size type NUMBER */
dma-mask = < 268435455 >; /* embeddedsw.dts.params.dma-mask type NUMBER */
host_rx_fifo-size = < 512 >; /* embeddedsw.dts.params.host_rx_fifo-size type NUMBER */
ulpi-ddr = < 0 >; /* embeddedsw.dts.params.ulpi-ddr type NUMBER */
voltage-switch = < 0 >; /* embeddedsw.dts.params.voltage-switch type NUMBER */
}; // end usb@0xffb00000 (hps_0_usb0)

hps_0_usb1: usb@0xffb40000 {
  compatible = "snps,dwc-otg-1.0", "snps,dwc-otg";
  reg = < 0xFFFFB40000 0x00001000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 128 4 >;
  dev-nperio-tx-fifo-size = < 4096 >; /* embeddedsw.dts.params.dev-nperio-tx-fifo-size type NUMBER */
  dev-perio-tx-fifo-size = <512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 >=; /* embeddedsw.dts.params.dev-perio-tx-fifo-size type STRING */
  dev-tx-fifo-size = "<512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 512 >="; /* embeddedsw.dts.params.dev-tx-fifo-size type STRING */
  dev_rx_fifo-size = < 512 >; /* embeddedsw.dts.params.dev_rx_fifo-size type NUMBER */
  dma-mask = < 268435455 >; /* embeddedsw.dts.params.dma-mask type NUMBER */
  host_rx_fifo-size = < 512 >; /* embeddedsw.dts.params.host_rx_fifo-size type NUMBER */
  ulpi-ddr = < 0 >; /* embeddedsw.dts.params.ulpi-ddr type NUMBER */
  voltage-switch = < 0 >; /* embeddedsw.dts.params.voltage-switch type NUMBER */
}; // end usb@0xffb40000 (hps_0_usb1)

hps_0_gmac0: ethernet@0xff700000 {
  compatible = "synopsys,dmac-1.0", "altr,sofpga-stmmac", "snps,dmac-3.70a", "snps,dmac";
  reg = < 0xFF700000 0x000002000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 115 4 >;
  interrupt-names = "macirq"; /* embeddedsw.dts.params.interrupt-names type STRING */
  mac-address = "[00 00 00 00 00 00]"; /* embeddedsw.dts.params.mac-address type STRING */
  address-bits = < 48 >;
  max-frame-size = < 1518 >;
  local-mac-address = [ 00 00 00 00 00 00 ];
  status = "disabled"; /* appended from boardinfo */
}; // end ethernet@0xff700000 (hps_0_gmac0)

hps_0_gmac1: ethernet@0xff702000 {
  compatible = "synopsys,dmac-1.0", "altr,sofpga-stmmac", "snps,dmac-3.70a", "snps,dmac";
  reg = < 0xFF702000 0x00002000 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 0 120 4 >;
  interrupt-names = "macirq"; /* embeddedsw.dts.params.interrupt-names type STRING */
  mac-address = "[00 00 00 00 00 00]"; /* embeddedsw.dts.params.mac-address type STRING */
  address-bits = < 48 >;
  max-frame-size = < 1518 >;
  local-mac-address = [ 00 00 00 00 00 00 ];
  phy-mode = "rgmi"; /* appended from boardinfo */
  phy-addr = < 0xFFFFFFFF >; /* appended from boardinfo */
}; // end ethernet@0xff702000 (hps_0_gmac1)

hps_0_timer: timer@0xfffec600 {
  compatible = "arm,cortex-a9-twd-timer-1.0", "arm,cortex-a9-twd-timer";
  reg = < 0xFFFEC600 0x00000100 >;
  interrupt-parent = < &hps_0_arm_gic_0 >;
  interrupts = < 1 13 3844 >;
}; // end timer@0xfffec600 (hps_0_timer)

leds {
  compatible = "gpio-leds";

226
Listing 16.23: Modified Device Tree File (socfpga.dts)

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

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

16.5 Database - SQL Table Creation Code
DROP TABLE Anomaly
END

IF object_id('DetectionEvent') is not null
BEGIN
DROP TABLE DetectionEvent
END

IF object_id('CameraStats') is not null
BEGIN
DROP TABLE CameraStats
END

IF object_id('Camera') is not null
BEGIN
DROP TABLE Camera
END

IF object_id('KnownImgs') is not null
BEGIN
DROP TABLE KnownImgs
END

IF object_id('Individual') is not null
BEGIN
DROP TABLE Individual
END

CREATE TABLE Individual
(
  ID Integer IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Name Varchar(40) NOT NULL,
  Details Varchar(100) NULL
);

CREATE TABLE KnownImgs
(
  KnownImgID Integer IDENTITY (1,1) NOT NULL PRIMARY KEY,
  IDImg Integer NOT NULL,
  Img Image NULL,
  FOREIGN KEY(IDImg) REFERENCES Individual(ID) ON DELETE CASCADE
);

CREATE TABLE Camera
(
  CameraID SmallInt NOT NULL PRIMARY KEY,
  CameraName Varchar(20) NOT NULL,
  CameraIP Varchar(40) NULL,
  FPGAIP Varchar(40) NULL,
  CameraLoc Varchar(40) NULL,
  CameraDetails Varchar(100) NULL,
  imgIndex smallint NULL,
  ThumbPath Varchar(10) NULL
);

CREATE TABLE CameraStats
(
  TimeID SmallInt NOT NULL PRIMARY KEY,
  CameraID SmallInt NOT NULL,
  LoTimeRange Time(2) NULL,
  HiTimeRange Time(2) NULL,
  Simga Float NULL,
  Mean Float NULL,
  NValue Integer Default(0),
  FOREIGN KEY(CameraID) REFERENCES Camera(CameraID) ON DELETE CASCADE
);

CREATE TABLE DetectionEvent

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

16.6 Behavioral Analysis Engine - Primary File

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Data.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SDBehavioralAnalysis
{
    class Program
    {
        // SeniorDesignCamStatsDataContext SDDatabase;
        static void Main(string[] args)
        {
            SeniorDesignCamStatsDataContext SDDatabase = new
            SeniorDesignCamStatsDataContext();
            DateTime CurrentTime = new DateTime();

            while (SDDatabase.DatabaseExists())
            {
                CurrentTime = DateTime.Now;
                var NewInEvents = (from events in SDDatabase.GetTable<DetectionEvent>()
                                    where (events.ExitType == null && events.isChecked == 0) select events);
                var NewOutEvents = (from events in SDDatabase.GetTable<DetectionEvent>()
                                     where (events.ExitType != null && events.EventTimeOut != null &&
                                               events.isChecked == 0) select events);
                var CamStats = (from camtables in SDDatabase.GetTable<CameraStat>()
                                 where (camtables.HitTimeRange > CurrentTime.TimeOfDay &&
                                        CurrentTime.TimeOfDay > camtables.LoTimeRange) select camtables);

                foreach (DetectionEvent Event in NewOutEvents)
                {
                    // Process NewOutEvents
                }

                // Process NewInEvents
            }
        }
    }
}
```
if (Event.ExitType == 0 || Event.EntryType == 0)
    ReportAnomaly(SDDatabase, Event);
foreach (CameraStat Camera in CamStats)
{
    if (Camera.CameraID == Event.EventID)
    {
        double x = (((TimeSpan)Event.EventTimeOut).TotalMinutes - Event.EventTimeIn.TotalMinutes);
        if (Camera.NValue > 36)
        {
            if (Math.Abs(x) > (Camera.Mean + (Camera.Sigma * 3)))
                ReportAnomaly(SDDatabase, Event);
            continue;
        }
        if (checkZVal(Camera, x) > 1.96)
            ReportAnomaly(SDDatabase, Event);
        UpdateStats(SDDatabase, Camera, Event, x);
    }
}
} //end of foreach Out

foreach (DetectionEvent Event in NewInEvents)
{
    foreach (CameraStat Camera in CamStats)
    {
        double x = (DateTime.Now.TimeOfDay.TotalMinutes - Event.EventTimeIn.TotalMinutes);
        if (Camera.NValue > 36 && checkZVal(Camera, x) > 1.96)
        {
            ReportAnomaly(SDDatabase, Event);
            Event.isChecked = 1;
            try
            {
                SDDatabase.SubmitChanges();
            }
            catch (Exception e)
            {
                Console.WriteLine("Database Connection Failed. Please Check Connection and Restart Program. Details:");
            }
        }
    }
} //end of while

private static double checkZVal(CameraStat CamInfo, double x)
{
    double z = ((x - (double)CamInfo.Mean) / (double)CamInfo.Sigma) * Math.Sqrt((double)CamInfo.NValue);
    return z;
}

private static void UpdateStats(SeniorDesignCamStatsDataContext SDDatabase, CameraStat CameraInfo, DetectionEvent Event, double x)
{
    int NewNVal = (int)CameraInfo.NValue + 1;
    double NewMean = (((double)CameraInfo.Mean * (NewNVal - 1)) + x) / NewNVal;
    CameraInfo.NValue = NewNVal;
    CameraInfo.Mean = NewMean;
```csharp
CameraInfo.Simga = NewSigma;
Event.isChecked = 1;
try
{
    SDDatabase.SubmitChanges();
    Console.WriteLine("Camera " + CameraInfo.CameraID + ": Updated with new values!");
}
catch (Exception e)
{
    Console.WriteLine("Database Connection Failed, Please Check Connection and Restart Program. Details: " + e);
    return;
}

private static void ReportAnomaly(SeniorDesignCamStatsDataContext SDDatabase, DetectionEvent Event)
{
    Anomaly newAnom = new Anomaly
    {
        EventID = Event.EventID,
        Severity = 0.0,
        AnomalyType = null,
        isChecked = 0
    };
    SDDatabase.Anomalies.InsertOnSubmit(newAnom);
    Event.isChecked = 1;
    try
    {
        SDDatabase.SubmitChanges();
        Console.WriteLine("Anomaly Detected ! ! ! ! ! ");
    }
    catch (Exception e)
    {
        Console.WriteLine("Database Connection Failed, Please Check Connection and Restart Program. Details: " + e);
        return;
    }
}

Listing 16.26: Primary File for the Behavioral Analysis Engine

16.7 Behavioral Analysis Engine - Database Configuration File

```
using System.Data.Linq;
using System.Data;
using System.Collections.Generic;
using System.Reflection;
using System.Linq;
using System.Linq.Expressions;
using System.ComponentModel;
using System;

[global::System.Data.Linq.Mapping.DatabaseAttribute(Name="SeniorDesignDB2")]
public partial class SeniorDesignCamStatsDataContext : System.Data.Linq.DataContext
{

    #region Extensibility Method Definitions
    partial void OnCreated();
    partial void InsertCamera(Camera instance);
    partial void UpdateCamera(Camera instance);
    partial void DeleteCamera(Camera instance);
    partial void InsertCameraStat(CameraStat instance);
    partial void UpdateCameraStat(CameraStat instance);
    partial void DeleteCameraStat(CameraStat instance);
    partial void InsertDetectionEvent(DetectionEvent instance);
    partial void UpdateDetectionEvent(DetectionEvent instance);
    partial void DeleteDetectionEvent(DetectionEvent instance);
    partial void InsertAnomaly(Anomaly instance);
    partial void UpdateAnomaly(Anomaly instance);
    partial void DeleteAnomaly(Anomaly instance);
    #endregion

    public SeniorDesignCamStatsDataContext():
    base(global::SBBehavioralAnalysis.Properties.Settings.Default.SeniorDesignDB2ConnectionString, mappingSource)
    |
    OnCreated();
    |
    public SeniorDesignCamStatsDataContext(string connection): base(connection, mappingSource)
    |
    OnCreated();
    |
    public SeniorDesignCamStatsDataContext(System.Data.IDbConnection connection):
    base(connection, mappingSource)
    |
    OnCreated();
    |
    base(connection, mappingSource)
    |
    OnCreated();
    |
    base(connection, mappingSource)
    |
    OnCreated();
    |
    public System.Data.Linq.Table<Camera> Cameras
}
public System.Data.Linq.Table<CameraStat> CameraStats
{
    get
    {
        return this.GetTable<CameraStat>();
    }
}

public System.Data.Linq.Table<DetectionEvent> DetectionEvents
{
    get
    {
        return this.GetTable<DetectionEvent>();
    }
}

public System.Data.Linq.Table<Anomaly> Anomalies
{
    get
    {
        return this.GetTable<Anomaly>();
    }
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Camera")]
public partial class Camera : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

    private short _CameraID;
    private string _CameraName;
    private string _CameraIP;
    private string _FPGAIP;
    private string _CameraLoc;
    private string _CameraDetails;
    private System.Nullable<short> _imgIndex;
    private string _ThumbPath;
    private EntitySet<CameraStat> _CameraStats;
    private EntitySet<DetectionEvent> _DetectionEvents;

    #region Extensibility Method Definitions
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
    partial void OnCameraIDChanging(short value);
    partial void OnCameraIDChanged();
    partial void OnCameraNameChanging(string value);
    partial void OnCameraNameChanged();
    partial void OnCameraIPChanging(string value);
    partial void OnCameraIPChanged();
    #endregion
}
partial void OnCameraIPChanged();
partial void OnFPGAIPChanging(string value);
partial void OnFPGAIPChanged();
partial void OnCameraLocChanging(string value);
partial void OnCameraLocChanged();
partial void OnCameraDetailsChanging(string value);
partial void OnCameraDetailsChanged();
partial void OnimgIndexChanging(System.Nullable<short> value);
partial void OnimgIndexChanged();
partial void OnThumbPathChanging(string value);
partial void OnThumbPathChanged();

#endregion

public Camera() {
    this._CameraStats = new EntitySet<CameraStat>(
        new Action<CameraStat>(
            this.attach_CameraStats),
        new Action<CameraStat>(
            this.detach_CameraStats));
    this._DetectionEvents = new EntitySet<DetectionEvent>(
        new Action<DetectionEvent>(
            this.attach_DetectionEvents),
        new Action<DetectionEvent>(
            this.detach_DetectionEvents));
    OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraID", DbType="SmallInt NOT NULL", IsPrimaryKey=true)]
public short CameraID {
    get {
        return this._CameraID;
    }
    set {
        if ((this._CameraID != value)) {
            this.OnCameraIDChanging(value);
            this.SendPropertyChanging();
            this._CameraID = value;
            this.SendPropertyChanged("CameraID");
            this.OnCameraIDChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraName", DbType="VarChar(20) NOT NULL", CanBeNull=false)]
public string CameraName {
    get {
        return this._CameraName;
    }
    set {
        if ((this._CameraName != value)) {
            this.OnCameraNameChanging(value);
            this.SendPropertyChanging();
            this._CameraName = value;
            this.SendPropertyChanged("CameraName");
            this.OnCameraNameChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraIP", DbType="VarChar(40)")]
public string CameraIP
get
{
    return this._CameraIP;
}

set
{
    if ((this._CameraIP != value))
    {
        this.OnCameraIPChanging(value);
        this.SendPropertyChanging();
        this._CameraIP = value;
        this.SendPropertyChanged("CameraIP");
        this.OnCameraIPChanged();
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FPGAIP", DbType="VarChar(40)")]
public string FPGAIP
{
    get
    {
        return this._FPGAIP;
    }
    set
    {
        if ((this._FPGAIP != value))
        {
            this.OnFPGAIPChanging(value);
            this.SendPropertyChanging();
            this._FPGAIP = value;
            this.SendPropertyChanged("FPGAIP");
            this.OnFPGAIPChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraLoc", DbType="VarChar(40)")]
public string CameraLoc
{
    get
    {
        return this._CameraLoc;
    }
    set
    {
        if ((this._CameraLoc != value))
        {
            this.OnCameraLocChanging(value);
            this.SendPropertyChanging();
            this._CameraLoc = value;
            this.SendPropertyChanged("CameraLoc");
            this.OnCameraLocChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraDetails", DbType="VarChar(100)")]
public string CameraDetails
{
    get
    {
        return this._CameraDetails;
    }
}
```csharp
set
[
    if (this._CameraDetails != value)
    {
        this.OnCameraDetailsChanging(value);
        this.SendPropertyChanging();
        this._CameraDetails = value;
        this.SendPropertyChanged("CameraDetails");
        this.OnCameraDetailsChanged();
    }
]

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_imgIndex", DbType="SmallInt")]
public System.Nullable<short> imgIndex
{
    get
    { return this._imgIndex; }
    set
    {
        if ((this._imgIndex != value))
        {
            this.OnImgIndexChanging(value);
            this.SendPropertyChanging();
            this._imgIndex = value;
            this.SendPropertyChanged("imgIndex");
            this.OnImgIndexChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ThumbPath", DbType="VarChar(10)")]
public string ThumbPath
{
    get
    { return this._ThumbPath; }
    set
    {
        if ((this._ThumbPath != value))
        {
            this.OnThumbPathChanging(value);
            this.SendPropertyChanging();
            this._ThumbPath = value;
            this.SendPropertyChanged("ThumbPath");
            this.OnThumbPathChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat", Storage="_CameraStats", ThisKey="CameraID", OtherKey="CameraID")]
public EntitySet<CameraStat> CameraStats
{
    get
    { return this._CameraStats; }
    set
    {
        this._CameraStats.Assign(value);
    }
}
```
public EntitySet<DetectionEvent> DetectionEvents
{
  get
  {
    return this._DetectionEvents;
  }
  set
  {
    this._DetectionEvents.Assign(value);
  }
}

class CameraStat : INotifyPropertyChanging, INotifyPropertyChanged
{
private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private short _TimeID;
}
private short _CameraID;
private System.Nullable<System.TimeSpan> _LoTimeRange;
private System.Nullable<System.TimeSpan> _HiTimeRange;
private System.Nullable<double> _Simga;
private System.Nullable<double> _Mean;
private System.Nullable<int> _NValue;
private EntityRef<Camera> _Camera;

#region Extensibility Method Definitions
partial void OnLoaded();
partial void OnValidate(System.Data.Linq.ChangeAction action);
partial void OnCreated();
partial void OnTimeIDChanging(short value);
partial void OnTimeIDChanged();
partial void OnCameraIDChanging(short value);
partial void OnCameraIDChanged();
partial void OnLoTimeRangeChanging(System.Nullable<System.TimeSpan> value);
partial void OnLoTimeRangeChanged();
partial void OnHiTimeRangeChanging(System.Nullable<System.TimeSpan> value);
partial void OnHiTimeRangeChanged();
partial void OnSimgaChanging(System.Nullable<double> value);
partial void OnSimgaChanged();
partial void OnMeanChanging(System.Nullable<double> value);
partial void OnMeanChanged();
partial void OnNValueChanging(System.Nullable<int> value);
partial void OnNValueChanged();
#endregion

public CameraStat() {
   this._Camera = default(EntityRef<Camera>);
   OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_TimeID", DbType="SmallInt NOT NULL", IsPrimaryKey=true)]
public short TimeID {
   get {
      return this._TimeID;
   }
   set {
      if ((this._TimeID != value)) {
         this.OnTimeIDChanging(value);
         this.SendPropertyChanging();
         this._TimeID = value;
         this.SendPropertyChanged("TimeID");
         this.OnTimeIDChanged();
      }
   }
}

public short CameraID {
   get {
      return this._CameraID;
   }
return this._CameraID;
} set
{
  if ((this._CameraID != value))
  {
    if (this._Camera.HasLoadedOrAssignedValue)
    {
      throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
    }
    this.OnCameraIDChanging(value);
    this.SendPropertyChanging();
    this._CameraID = value;
    this.SendPropertyChanged("CameraID");
    this.OnCameraIDChanged();
  }
}

{
  get
  {
    return this._LoTimeRange;
  }
  set
  {
    if ((this._LoTimeRange != value))
    {
      this.OnLoTimeRangeChanging(value);
      this.SendPropertyChanging();
      this._LoTimeRange = value;
      this.SendPropertyChanged("LoTimeRange");
      this.OnLoTimeRangeChanged();
    }
  }
}

{
  get
  {
    return this._HiTimeRange;
  }
  set
  {
    if ((this._HiTimeRange != value))
    {
      this.OnHiTimeRangeChanging(value);
      this.SendPropertyChanging();
      this._HiTimeRange = value;
      this.SendPropertyChanged("HiTimeRange");
      this.OnHiTimeRangeChanged();
    }
  }
}

{
  get
  {
    return this._Simga;
  }
}
set
[
    if ((this._Simga != value))
    {
        this.OnSimgaChanging(value);
        this.SendPropertyChanging();
        this._Simga = value;
        this.SendPropertyChanged("Simga");
        this.OnSimgaChanged();
    }
]

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Mean", DbType="Float")]
public System.Nullable<double> Mean
{
    get
    {
        return this._Mean;
    }
    set
    {
        if ((this._Mean != value))
        {
            this.OnMeanChanging(value);
            this.SendPropertyChanging();
            this._Mean = value;
            this.SendPropertyChanged("Mean");
            this.OnMeanChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_NValue", DbType="Int")]
public System.Nullable<int> NValue
{
    get
    {
        return this._NValue;
    }
    set
    {
        if ((this._NValue != value))
        {
            this.OnNValueChanging(value);
            this.SendPropertyChanging();
            this._NValue = value;
            this.SendPropertyChanged("NValue");
            this.OnNValueChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat", Storage="_Camera", ThisKey="CameraID", OtherKey="CameraID", IsForeignKey=true, DeleteOnNull=true, DeleteRule="CASCADE")]
public Camera Camera
{
    get
    {
        return this._Camera.Entity;
    }
    set
    {
        Camera previousValue = this._Camera.Entity;
        if (((previousValue != value) || (this._Camera.HasLoadedOrAssignedValue == false)))
        {
        }
    }
}
```csharp
if ((previousValue != null))
{
    this._Camera.Entity = null;
    previousValue.CameraStats.Remove(this);
}
this._Camera.Entity = value;
if ((value != null))
{
    value.CameraStats.Add(this);
    this._CameraID = value.CameraID;
}
else
{
    this._CameraID = default(short);
}
this.SendPropertyChanged("Camera");
}
}
}
}
}
public event PropertyChangingEventHandler PropertyChanging;
public event PropertyChangedEventHandler PropertyChanged;
protected virtual void SendPropertyChanging()
{
    if ((this.PropertyChanging != null))
    {
        this.PropertyChanging(this, emptyChangingEventArgs);
    }
}
protected virtual void SendPropertyChanged(String propertyName)
{
    if ((this.PropertyChanged != null))
    {
        this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}
[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.DetectionEvent")]
public partial class DetectionEvent : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);
    private int _EventID;
    private short _EventCamera;
    private System.TimeSpan _EventTimeIn;
    private System.Nullable<System.TimeSpan> _EventTimeOut;
    private System.Nullable<int> _isChecked;
    private System.Nullable<short> _EntryType;
    private System.Nullable<short> _ExitType;
    private int _SuspectID;
    private double _Confidence;
    private EntitySet<Anomaly> _Anomalies;
}
private EntityRef<Camera>_Camera;

#region Extensibility Method Definitions
partial void OnLoaded();
partial void OnValidate(System.Data.Linq.ChangeAction action);
partial void OnCreated();
partial void OnEventIDChanging(int value);
partial void OnEventIDChanged();
partial void OnEventCameraChanging(short value);
partial void OnEventCameraChanged();
partial void OnEventTimeInChanging(System.TimeSpan value);
partial void OnEventTimeInChanged();
partial void OnEventTimeOutChanging(System.Nullable<System.TimeSpan> value);
partial void OnEventTimeOutChanged();
partial void OnIsCheckedChanging(System.Nullable<int> value);
partial void OnIsCheckedChanged();
partial void OnEntryTypeChanging(System.Nullable<short> value);
partial void OnEntryTypeChanged();
partial void OnExitTypeChanging(System.Nullable<short> value);
partial void OnExitTypeChanged();
partial void OnSuspectIDChanging(int value);
partial void OnSuspectIDChanged();
partial void OnConfidenceChanging(double value);
partial void OnConfidenceChanged();
#endregion

public DetectionEvent()
{
    this._Anomalies = new EntitySet<Anomaly>(
        new Action<Anomaly>(this.attach_Anomalies),
        new Action<Anomaly>(this.detach_Anomalies));
    this._Camera = default(EntityRef<Camera>);
    OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EventID", DbType="Int NOT NULL", IsPrimaryKey=true)]
public int EventID
{
    get
    {
        return this._EventID;
    }
    set
    {
        if ((this._EventID != value))
        {
            this.OnEventIDChanging(value);
            this.SendPropertyChanging();
            this._EventID = value;
            this.SendPropertyChanged("EventID");
            this.OnEventIDChanged();
        }
    }
}

{
    get
    {
        return this._EventCamera;
    }
    set
    {
        if ((this._EventCamera != value))
        {

if (this._Camera.HasLoadedOrAssignedValue)
{
    throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
}
this._OnEventCameraChanging(value);
this._SendPropertyChanging();
this._EventCamera = value;
this._SendPropertyChanged("EventCamera");
this._OnEventCameraChanged();
}
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage=""EventTimeIn", DbType=""Time NOT NULL")]
public System.TimeSpan EventTimeIn
{
    get
    {
        return this._EventTimeIn;
    }
    set
    {
        if ((this._EventTimeIn != value))
        {
            this._OnEventTimeInChanging(value);
            this._SendPropertyChanging();
            this._EventTimeIn = value;
            this._SendPropertyChanged("EventTimeIn");
            this._OnEventTimeInChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage=""EventTimeOut", DbType=""Time")]
{
    get
    {
        return this._EventTimeOut;
    }
    set
    {
        if ((this._EventTimeOut != value))
        {
            this._OnEventTimeOutChanging(value);
            this._SendPropertyChanging();
            this._EventTimeOut = value;
            this._SendPropertyChanged("EventTimeOut");
            this._OnEventTimeOutChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage=""isChecked", DbType=""Int")]
public System.Nullable<int> isChecked
{
    get
    {
        return this._isChecked;
    }
    set
    {
        if ((this._isChecked != value))
        {
            this._OnisCheckedChanging(value);
            this._SendPropertyChanging();
            this._SendPropertyChanged("isChecked");
            this._OnisCheckedChanged();
        }
    }
}
this._isChecked = value;
this.SendPropertyChanged("isChecked");
this.OnisCheckedChanged();
}
}
}
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EntryType", DbType="SmallInt")]
public System.Nullable<short> EntryType
{
get
{
    return this._EntryType;
}
set
{
    if ((this._EntryType != value))
    {
        this.OnEntryTypeChanging(value);
        this.SendPropertyChanging();
        this._EntryType = value;
        this.SendPropertyChanged("EntryType");
        this.OnEntryTypeChanged();
    }
}
}
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ExitType", DbType="SmallInt")]
public System.Nullable<short> ExitType
{
get
{
    return this._ExitType;
}
set
{
    if ((this._ExitType != value))
    {
        this.OnExitTypeChanging(value);
        this.SendPropertyChanging();
        this._ExitType = value;
        this.SendPropertyChanged("ExitType");
        this.OnExitTypeChanged();
    }
}
}
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_SuspectID", DbType="Int NOT NULL")]
public int SuspectID
{
get
{
    return this._SuspectID;
}
set
{
    if ((this._SuspectID != value))
    {
        this.OnSuspectIDChanging(value);
        this.SendPropertyChanging();
        this._SuspectID = value;
        this.SendPropertyChanged("SuspectID");
        this.OnSuspectIDChanged();
    }
}
public double Confidence
{
    get
    {
        return this._Confidence;
    }
    set
    {
        if ((this._Confidence != value))
        {
            this.OnConfidenceChanging(value);
            this.SendPropertyChanging();
            this._Confidence = value;
            this.SendPropertyChanged("Confidence");
            this.OnConfidenceChanged();
        }
    }
}

public EntitySet<Anomaly> Anomalies
{
    get
    {
        return this._Anomalies;
    }
    set
    {
        this._Anomalies.Assign(value);
    }
}

public Camera Camera
{
    get
    {
        return this._Camera.Entity;
    }
    set
    {
        Camera previousValue = this._Camera.Entity;
        if (((previousValue != value) || (this._Camera.HasLoadedOrAssignedValue == false)))
        {
            this.SendPropertyChanging();
            if ((previousValue != null))
            {
                this._Camera.Entity = null;
                previousValue.DetectionEvents.Remove(this);
            }
            this._Camera.Entity = value;
            if ((value != null))
            {
                value.DetectionEvents.Add(this);
                this._EventCamera = value.CameraID;
            }
            else
            {
                this._EventCamera = default(short);
            }
        }
    }
}
protected virtual void SendPropertyChanging()
{
    if ((this.PropertyChanging != null))
    {
        this.PropertyChanging(this, emptyChangingEventArgs);
    }
}

protected virtual void SendPropertyChanged(String propertyName)
{
    if ((this.PropertyChanged != null))
    {
        this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}

private void attachAnomalies(Anomaly entity)
{
    this.SendPropertyChanging();
    entity.DetectionEvent = this;
}

private void detachAnomalies(Anomaly entity)
{
    this.SendPropertyChanging();
    entity.DetectionEvent = null;
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Anomaly")]
public partial class Anomaly : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

    private int _AnomalyID;
    private int _EventID;
    private System.Nullable<double> _Severity;
    private System.Nullable<short> _isChecked;
    private System.Nullable<short> _AnomalyType;
    private EntityRef<DetectionEvent> _DetectionEvent;

    #region Extensibility Method Definitions
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
    partial void OnAnomalyIDChanging(int value);
    partial void OnAnomalyIDChanged();
    partial void OnEventIDChanging(int value);
    partial void OnEventIDChanged();
    partial void OnSeverityChanging(System.Nullable<double> value);
    partial void OnSeverityChanged();
    partial void OnisCheckedChanging(System.Nullable<short> value);
    partial void OnisCheckedChanged();
    #endregion
}
partial void OnisCheckedChanged();
partial void OnAnomalyTypeChanging(System.Nullable<short> value);
partial void OnAnomalyTypeChanged();
#endregion

public Anomaly()
{
this._DetectionEvent = default(EntityRef<DetectionEvent>);
OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_AnomalyID", AutoSync=AutoSync.OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true, IsDbGenerated=true)]
public int AnomalyID
{
get
{
return this._AnomalyID;
}
set
{
if ((this._AnomalyID != value))
{
this.OnAnomalyIDChanging(value);
this.SendPropertyChanging();
this._AnomalyID = value;
this.SendPropertyChanged("AnomalyID");
this.OnAnomalyIDChanged();
}
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EventID", DbType="Int NOT NULL")]
public int EventID
{
get
{
return this._EventID;
}
set
{
if ((this._EventID != value))
{
if (this._DetectionEvent.HasLoadedOrAssignedValue)
{
throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
}
this.OnEventIDChanging(value);
this.SendPropertyChanging();
this._EventID = value;
this.SendPropertyChanged("EventID");
this.OnEventIDChanged();
}
}

public System.Nullable<double> Severity
{
get
{
return this._Severity;
}
set
{
if ((this._Severity != value))
{
    this.OnSeverityChanging(value);
    this.SendPropertyChanging();
    this._Severity = value;
    this.SendPropertyChanged("Severity");
    this.OnSeverityChanged();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_isChecked", DbType="SmallInt")]
public System.Nullable<short> isChecked
{
    get
    {
        return this._isChecked;
    }
    set
    {
        if ((this._isChecked != value))
        {
            this.OnisCheckedChanging(value);
            this.SendPropertyChanging();
            this._isChecked = value;
            this.SendPropertyChanged("isChecked");
            this.OnisCheckedChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_AnomalyType", DbType="SmallInt")]
public System.Nullable<short> AnomalyType
{
    get
    {
        return this._AnomalyType;
    }
    set
    {
        if ((this._AnomalyType != value))
        {
            this.OnAnomalyTypeChanging(value);
            this.SendPropertyChanging();
            this._AnomalyType = value;
            this.SendPropertyChanged("AnomalyType");
            this.OnAnomalyTypeChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="DetectionEvent_Anomaly", Storage="_DetectionEvent", ThisKey="EventID", OtherKey="EventID", IsForeignKey=true, DeleteOnNull=true, DeleteRule="CASCADE")]
public DetectionEvent DetectionEvent
{
    get
    {
        return this._DetectionEvent.Entity;
    }
    set
    {
        DetectionEvent previousValue = this._DetectionEvent.Entity;
        if (((previousValue != value) || (this._DetectionEvent.HasLoadedOrAssignedValue == false)))
        {
            
        }
Listing 16.27: Database Configuration File

16.8 User Interface - Primary Form

```csharp
namespace Senior.Design.UI
{
    public partial class Form1 : Form
    {
```
delegate void SetTextCallback (string text);
delegate void SetUICallback();
System.Threading.Thread DBThread;

/************************************** LOADING FUNCTIONS
***************************************/
public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    /** Basic Options and links being established **/
    Globals.StreamThumbsList.ImageSize = new Size(95, 60);
    Globals.EventThumbsList.ImageSize = new Size(95, 60);
    VideoStream.CtlVisible = false;
    VideoStream.Update();
    DBThread = new System.Threading.Thread(StartDBCalls);
    DBThread.Start();

    /** Setting up resizing options and different screen options **/
    StreamListMenu.Size = new Size(StreamListMenu.Size.Width, (RightMenu.Size.Height / 2) - 12);
    ObjectList.Size = new Size(ObjectList.Size.Width, (RightMenu.Size.Height / 2) - 12);
    NoteLog.Location = new Point(NoteLog.Location.X, (RightMenu.Size.Height + menuStrip1.Size.Height - NoteLog.Size.Height) - 5);
    VideoStream.Size = new Size(RightMenu.Location.X - 24, NoteLog.Location.Y - 12 - menuStrip1.Size.Height);
    Identity.Location = new Point(NoteLog.Size.Width + NoteLog.Location.X + 12, NoteLog.Location.Y);
    if (RightMenu.Location.X > 800)
    {
        NoteLog.Size = new Size((RightMenu.Location.X - Identity.Size.Width), NoteLog.Size.Height);
        Identity.Size = new Size((RightMenu.Location.X - NoteLog.Size.Width) - 36, Identity.Size.Height);
        Identity.Location = new Point(NoteLog.Size.Width + NoteLog.Location.X + 12, Identity.Location.Y);
        Identity.Show();
    } else
    {
        NoteLog.Size = new Size(RightMenu.Location.X - 12, NoteLog.Size.Height);
        Identity.Hide();
    }
    LoadDBCams();
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    DBThread.Abort();
}

private void LoadDBCams()
{
    SDDBDataContext DB = new SDDBDdataContext();
    var Cameras = (from cams in DB.GetTable<Camera>() select cams);
    foreach (Camera DBCam in Cameras)
    {
try
{
    Image NewThumb = Image.FromFile(DBCam.ThumbPath);
    Globals.StreamThumbsList.Images.Add(NewThumb);
}
catch
{
    Image FailedImg = Image.FromFile("C:/Users/nfox/Source/Repos/
                                        SmartCameras/Senior Design UI/Senior Design UI/noImageAvailable.
                                        jpg");
    Globals.StreamThumbsList.Images.Add(FailedImg);
}
RSTPStream CamFromDB = new RSTPStream(DBCam.CameraIP, DBCam.CameraDetails,
                                        DBCam.CameraLoc, Globals.StreamList.Count + 1);
Globals.StreamList.Add(CamFromDB);
UpdateStreamListBox();
*
******************************************************************************
******************************************************************************
private void addStreamToolStripMenuClick(object sender, EventArgs e)
{
    StreamAddForm = new AddStream();
    StreamAddForm.ShowDialog();
    if (StreamAddForm.DialogResult == DialogResult.OK)
    {
        StreamThumbRefresh(Globals.StreamList[Globals.StreamList.Count - 1]);
    }
}
private void StreamThumbRefresh(RSTPStream thisStream)
{
    // Image Newthumb = Image.FromFile();
    // Globals.StreamThumbsList.Images[thisStream.get_imgIndex()] = NewThumb;
    /* Not Working Yet
    Uri ThumbUrl = new Uri(thisStream.get_url() + "/dsm?");
    WebRequest ThumbNailRequest = WebRequest.Create(ThumbUrl);
    WebResponse ThumbNailReponse = ThumbNailRequest.GetResponse();
    if (ThumbNailReponse.ResponseUri.AbsoluteUri != null)
    {
       Globals.StreamThumbsList.Images[thisStream.get_imgIndex()] = NewThumb;
    }*/
    UpdateStreamListBox();
}
private void UpdateStreamListBox()
{
    for (int i = 0; i < Globals.StreamList.Count; i++)
    {
        ListViewItem thisColumn = new ListViewItem("", i);
        thisColumn.SubItems.Add(Globals.StreamList[i].get_loc());
        thisColumn.SubItems.Add(Globals.StreamList[i].get_details());
        try
        {
            StreamListView.Items[i] = thisColumn;
        }
        catch (ArgumentOutOfRangeException e)
        {
            StreamListView.Items.Add(thisColumn);
        }
}
private void Form1_ResizeBegin(object sender, EventArgs e)
{
    StreamListMenu.Size = new Size(StreamListMenu.Size.Width, (RightMenu.Size.Height / 2) - 12);
    ObjectList.Size = new Size(ObjectList.Size.Width, (RightMenu.Size.Height / 2) - 12);
    NoteLog.Location = new Point(NoteLog.Location.X, (RightMenu.Size.Height + menuStrip1.Size.Height + NoteLog.Size.Height) - 5);
    if (RightMenu.Location.X > 800)
    {
        NoteLog.Size = new Size((RightMenu.Location.X - Identity.Size.Width) - 36, NoteLog.Size.Height);
        Identity.Size = new Size(300, Identity.Size.Height);
        Identity.Location = new Point(NoteLog.Size.Width + NoteLog.Location.X + 12, Identity.Location.Y);
        Identity.Show();
    }
    else
    {
        NoteLog.Size = new Size(RightMenu.Location.X - 12, NoteLog.Size.Height);
        Identity.Hide();
    }
}

private void refreshIconsToolStripMenuItem_Click(object sender, EventArgs e)
{
    UpdateStreamListBox();
}

private void StreamListView_SelectedIndexChanged(object sender, EventArgs e)
{
    ListView.SelectedItemsCollection indexes = this.StreamListView.SelectedItems;
    int index = 0;
    foreach (ListViewItem item in indexes)
    {
        if (item.Selected)
        {
            index = item.Index;
            break;
        }
    }

    String VideoUrl = "rtsp://" + Globals.StreamList[index].get_url() + "/live1.sdp";
    VideoStream.playlist.add(VideoUrl);
    VideoStream.playlist.next();
    VideoStream.playlist.play();
}

/****************************************************************************
** DATABASE FUNCTIONS
****************************************************************************/

// Database Continuous Calls checking for anomalies
private void StartDBCalls()
{
    SDDBDataContext SDDatabase = new SDDBDataContext();
    DateTime CurrTime = DateTime.Now;
    while (SDDatabase.DatabaseExists())
    {
        if (CurrTime <= DateTime.Now)
// SafeSetText (DateTime.Now.TimeOfDay + "Queries started");
UpdateAnoms (SDDatabase);
UpdateEvents (SDDatabase);
CurrTime = CurrTime.AddMilliseconds (1500);
// SafeSetText (DateTime.Now.TimeOfDay + "Queries completed");
else
{
    System.Threading.Thread.Sleep (500);
}
else
{
    System.Threading.Thread.Sleep (500);
}

private void UpdateAnoms (SDDBDataContext SDB)
{
    var newAnomalies = (from anom in SDB.GetTable<Anomaly>() where (anom.
isChecked == 0) select anom);
    string LogOutput = "";
    foreach (Anomaly anom in newAnomalies)
    {
        try
        {
            anom.isChecked = 1;
            SDB.SubmitChanges ();
            var eventID = SDB.DetectionEvents.Single (x => x.EventID == anom.
EventID);
            LogOutput = DateTime.Now.TimeOfDay + ": Anomaly Detected in Camera: "+
CameraLoc + " with individual: " + SDB.Individuals.Single (x => x.
ID == eventID.SuspectID).Name + "\n";
        } catch (Exception e)
        {
            LogOutput = DateTime.Now.TimeOfDay + ": Database Connection Failed, 
Please Check Connection. Details: " + e + "\n";
        }
        SafeSetText (LogOutput);
    }
}

private void UpdateEvents (SDDBDataContext SDB)
{
    var OngoingEvents = (from events in SDB.GetTable <DetectionEvent>() where (events.EventTimeIn != null && events.EventTimeOut == null) select events);
    bool UpdateFlag = false;
    foreach (DetectionEvent events in OngoingEvents)
    {
        string IDName = SDB.Individuals.Single (v => v.ID == events.SuspectID).
Name;
        if (Globals.EventList.Exists (x => x.EventID == events.EventID))
            continue;
        UpdateFlag = true;
        EventObject newEvent = new EventObject (IDName, (Int32)events.EventCamera, 
Globals.EventList.Count, (Int32)events.EventID, 0, 0);
        Globals.EventList.Add (newEvent);
        Image FailedImg = Image.FromFile ("C:/Users/nfox/Source/Repos/SmartCameras/ 
Senior Design UI/Senior Design UI/noImageAvailable.jpg");
        Globals.EventThumbsList.Images.Add (FailedImg);
    }
    if (UpdateFlag)
        UpdateUISafe () ;
}

private void UpdateUISafe ()
{
}
if (this.ObjectListView.InvokeRequired)
{
    SetUICallback d = new SetUICallback(UpdateUISafe);
    this.Invoke(d, new object[]{})
}
else
{
    for (int i = 0; i < Globals.EventList.Count; i++)
    {
        ListViewItem thisColumn = new ListViewItem("", i);
        thisColumn.SubItems.Add(Globals.EventList[i].Individual);
        thisColumn.SubItems.Add(Globals.EventList[i].CameraID - 1].getLoc());
        thisColumn.SubItems.Add(Globals.EventList[i].X.ToString());
        thisColumn.SubItems.Add(Globals.EventList[i].Y.ToString());
        thisColumn.SubItems.Add("0");
        try
        {
            ObjectListView.Items[i] = thisColumn;
        }
        catch (ArgumentOutOfRangeException e)
        {
            ObjectListView.Items.Add(thisColumn);
        }
    }
}
private void SafeSetText(StringSetText)
{
    if (this.NotificationLog.InvokeRequired)
    {
        SetTextCallback d = new SetTextCallback(SafeSetText);
        this.Invoke(d, new object[]{SetText});
    }
    else
    {
        this.NotificationLog.AppendText(SetText + "\n");
    }
}
private void textBox1_Enter(object sender, EventArgs e)
{
    NotificationLog.AppendText(DateTime.Now.TimeOfDay + ": " + textBox1.Text + "\n");
    textBox1.Text = "";
}
private void NotificationLog_TextChanged(object sender, EventArgs e)
{
    SDDBDatContext SDDB = new SDDBDatContext();
    var Array = NotificationLog.Text.Split('\n');
    if (SDDB.Log == null)
    {
        return;
    }
    try
    {
        SDDB.Log.NewLine = Array[Array.Length - 1];
    }
    catch
    {
        SDDB.Log.NewLine = "Bug in TextLog, revisit later";
    }
}
public class Globals
public partial class Form1
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));
        this.NoteLog = new System.Windows.Forms.GroupBox();
        this.textEnterLine = new System.Windows.Forms.TextBox();
        this.label1 = new System.Windows.Forms.Label();
        this.VideoStream = new AxAXVLC.AxVLCPlugin2();
        this.menuStrip1 = new System.Windows.Forms.MenuStrip();
        this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.streamsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.addStreamToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.refreshIconsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.imageList1 = new System.Windows.Forms.ImageList();
        this_STREAMListView = new System.Windows.Forms.ListView();
        this.StreamListView = new System.Windows.Forms.ListView();
    
    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));
        this.NoteLog = new System.Windows.Forms.GroupBox();
        this.textEnterLine = new System.Windows.Forms.TextBox();
        this.label1 = new System.Windows.Forms.Label();
        this.VideoStream = new AxAXVLC.AxVLCPlugin2();
        this.menuStrip1 = new System.Windows.Forms.MenuStrip();
        this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.streamsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.addStreamToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.refreshIconsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.imageList1 = new System.Windows.Forms.ImageList();
        this_STREAMListView = new System.Windows.Forms.ListView();
        this.StreamListView = new System.Windows.Forms.ListView();
    }
this.StreamListMenu = new System.Windows.Forms.GroupBox();
this.ObjectList = new System.Windows.Forms.GroupBox();
this.RightMenu = new System.Windows.Forms.GroupBox();
this.pictureBox1 = new System.Windows.Forms.PictureBox();
this.Identity = new System.Windows.Forms.GroupBox();
this.NoteLog.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.VideoStream)).EndInit();
this.menuStrip1.SuspendLayout();
this.StreamListMenu.SuspendLayout();
this.ObjectList.SuspendLayout();
this.RightMenu.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.Identity.SuspendLayout();
this.SuspendLayout();
// NoteLog
//
this.NoteLog.AutoSize = true;
this.NoteLog.Controls.Add(this.NotificationLog);
this.NoteLog.Controls.Add(this.textEnterLine);
this.NoteLog.Location = new System.Drawing.Point(9, 596);
this.NoteLog.Margin = new System.Windows.Forms.Padding(2);
this.NoteLog.Name = "NoteLog";
this.NoteLog.Padding = new System.Windows.Forms.Padding(2);
this.NoteLog.Size = new System.Drawing.Size(721, 193);
this.NoteLog.TabIndex = 2;
this.NoteLog.TabStop = false;
this.NoteLog.Text = "Notification Log";
// NotificationLog
//
this.NotificationLog.Location = new System.Drawing.Point(9, 15);
this.NotificationLog.Margin = new System.Windows.Forms.Padding(2);
this.NotificationLog.MaxLength = 50000;
this.NotificationLog.Name = "NotificationLog";
this.NotificationLog.ReadOnly = true;
this.NotificationLog.Size = new System.Drawing.Size(717, 154);
this.NotificationLog.TabIndex = 1;
this.NotificationLog.Text = "";
this.NotificationLog.TextChanged += new System.EventHandler(this.NotificationLog_TextChanged);
// textEnterLine
//
this.textEnterLine.Location = new System.Drawing.Point(2, 171);
this.textEnterLine.Margin = new System.Windows.Forms.Padding(2);
```csharp
this.textEnterLine.Name = "textEnterLine";
this.textEnterLine.Size = new System.Drawing.Size(717, 20);
this.textEnterLine.TabIndex = 0;
this.textEnterLine.Leave += new System.EventHandler(this.textBox1_Leave);

// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(18, 25);
this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(56, 13);
this.label1.TabIndex = 5;
this.label1.Text = "Testlabel1";
this.label1.Visible = false;

// VideoStream
//
this.VideoStream.Enabled = true;
this.VideoStream.Location = new System.Drawing.Point(11, 27);
this.VideoStream.Margin = new System.Windows.Forms.Padding(2);
this.VideoStream.Name = "VideoStream";
this.VideoStream.Size = new System.Drawing.Size(1441, 700);
this.VideoStream.TabIndex = 8;

// menuStrip1
//
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] { this.fileToolStripMenuItem, this.streamsToolStripMenuItem });
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Padding = new System.Windows.Forms.Padding(4, 2, 0, 2);
this.menuStrip1.Size = new System.Drawing.Size(1426, 24);
this.menuStrip1.TabIndex = 10;
this.menuStrip1.Text = "menuStrip1";

// fileToolStripMenuItem
//
this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);
this.fileToolStripMenuItem.Text = "File";
this.fileToolStripMenuItem.Click += new System.EventHandler(this.fileToolStripMenuItem_Click);

// streamsToolStripMenuItem
//
this.streamsToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] { this.addStreamToolStripMenuItem, this.refreshIconsToolStripMenuItem });
this.streamsToolStripMenuItem.Name = "streamsToolStripMenuItem";
this.streamsToolStripMenuItem.Size = new System.Drawing.Size(61, 20);
this.streamsToolStripMenuItem.Text = "Streams";
this.streamsToolStripMenuItem.Click += new System.EventHandler(this.addStreamToolStripMenuItem_Click);

// refreshIconsToolStripMenuItem
//
this.refreshIconsToolStripMenuItem.Name = "refreshIconsToolStripMenuItem";
this.refreshIconsToolStripMenuItem.Size = new System.Drawing.Size(144, 22);
this.refreshIconsToolStripMenuItem.Text = "Add Stream";
this.refreshIconsToolStripMenuItem.Click += new System.EventHandler(this.refreshIconsToolStripMenuItem_Click);
```

257
// refreshIconsToolStripMenuItem.Name = "refreshIconsToolStripMenuItem";
this.refreshIconsToolStripMenuItem.Size = new System.Drawing.Size(144, 22);
this.refreshIconsToolStripMenuItem.Text = "Refresh Icons";
this.refreshIconsToolStripMenuItem.Click += new System.EventHandler(this.refreshIconsToolStripMenuItem_Click);

// imageList1
//
this.imageList1.ImageSize = new System.Drawing.Size(80, 80);
this.imageList1.TransparentColor = System.Drawing.Color.Transparent;

// StreamListView
//
this.StreamListView.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] { this.columnHeader6, this.columnHeader7, this.columnHeader8 });
this.StreamListView.FullRowSelect = true;
this.StreamListView.GridLines = true;
this.StreamListView.Location = new System.Drawing.Point(2, 15);
this.StreamListView.Margin = new System.Windows.Forms.Padding(2);
this.StreamListView.MultiSelect = false;
this.StreamListView.Name = "StreamListView";
this.StreamListView.Size = new System.Drawing.Size(420, 347);
this.StreamListView.SmallImageList = this.imageList1;
this.StreamListView.TabIndex = 11;
this.StreamListView.TileSize = new System.Drawing.Size(400, 65);
this.StreamListView.UseCompatibleStateImageBehavior = false;
this.StreamListView.ItemActivate += new System.EventHandler(this.StreamListView_SelectedIndexChanged);
this.StreamListView.SelectedIndexChanged += new System.EventHandler(this.StreamListView_SelectedIndexChanged);
this.StreamListView.Click += new System.EventHandler(this.StreamListView_SelectedIndexChanged);

// columnHeader6
//
this.columnHeader6.Text = "Thumb Nail";
this.columnHeader6.Width = 100;

// columnHeader7
//
this.columnHeader7.Text = "Stream Name";
this.columnHeader7.Width = 100;

// columnHeader8
//
this.columnHeader8.Text = "Details";
this.columnHeader8.Width = 216;

// StreamListMenu
//
this.StreamListMenu.Controls.Add(this.StreamListView);
this.StreamListMenu.Location = new System.Drawing.Point(2, 15);
this.StreamListMenu.Margin = new System.Windows.Forms.Padding(2);
this.StreamListMenu.Name = "StreamListMenu";
this.StreamListMenu.Padding = new System.Windows.Forms.Padding(2);
this.StreamListMenu.Size = new System.Drawing.Size(420, 364);
this.StreamListMenu.TabIndex = 12;
this.StreamListMenu.TabStop = false;
this.StreamListMenu.Text = "Stream List";

// ObjectList
this.ObjectList.Controls.Add(this.ObjectListView);
this.ObjectList.Location = new System.Drawing.Point(2, 382);
this.ObjectList.Margin = new System.Windows.Forms.Padding(2);
this.ObjectList.Name = "ObjectList";
this.ObjectList.Padding = new System.Windows.Forms.Padding(2);
this.ObjectList.Size = new System.Drawing.Size(424, 387);
this.ObjectList.TabIndex = 13;
this.ObjectList.TabStop = false;
this.ObjectList.Text = "Object List";

// ObjectListView
    columnHeader1,
    columnHeader2,
    columnHeader3,
    columnHeader4,
    columnHeader5 });
this.ObjectListView.FullRowSelect = true;
this.ObjectListView.GridLines = true;
this.ObjectListView.Location = new System.Drawing.Point(2, 15);
this.ObjectListView.Margin = new System.Windows.Forms.Padding(2);
this.ObjectListView.Name = "ObjectListView";
this.ObjectListView.Size = new System.Drawing.Size(420, 370);
this.ObjectListView.SmallImageList = this.imageList1;
this.ObjectListView.TabIndex = 11;
this.ObjectListView.TileSize = new System.Drawing.Size(400, 100);
this.ObjectListView.UseCompatibleStateImageBehavior = false;

// columnhead1
// this.columnHeader1.Text = "Thumbnail";
this.columnHeader1.Width = 100;

// columnHeader2
// this.columnHeader2.Text = "Name";
this.columnHeader2.Width = 100;

// columnHeader1
// this.columnHeader1.Text = "Stream";
this.columnHeader1.Width = 90;

// columnHeader3
// this.columnHeader3.Text = "X";
this.columnHeader3.Width = 20;

// columnHeader4
// this.columnHeader4.Text = "Y";
this.columnHeader4.Width = 20;

// columnHeader5
// this.columnHeader5.Text = "Last Seen";
this.columnHeader5.Width = 80;

// RightMenu
// this.RightMenu.Controls.Add(this.StreamListMenu);
this.RightMenu.Controls.Add(this.ObjectList);
this.RightMenu.Location = new System.Drawing.Point(998, 24);
this.RightMenu.Margin = new System.Windows.Forms.Padding(2);
this.RightMenu.Name = "RightMenu";
this.RightMenu.Padding = new System.Windows.Forms.Padding(2);
this.RightMenu.Size = new System.Drawing.Size(428, 771);
this.RightMenu.TabIndex = 14;
this.RightMenu.TabStop = false;
this.RightMenu.Text = "Side Menu";
//
// pictureBox1
//
this.pictureBox1.BackgroundImage = global::Senior_Design_UI.Properties.Resources.sceasel;
this.pictureBox1.Dock = System.Windows.Forms.DockStyle.Fill;
this.pictureBox1.Location = new System.Drawing.Point(2, 15);
this.pictureBox1.Margin = new System.Windows.Forms.Padding(2);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(351, 178);
this.pictureBox1.TabIndex = 15;
this.pictureBox1.TabStop = false;
//
// Identity
//
this.Identity.Controls.Add(this.label1);
this.Identity.Controls.Add(this.pictureBox1);
this.Identity.Location = new System.Drawing.Point(734, 596);
this.Identity.Margin = new System.Windows.Forms.Padding(2);
this.Identity.Name = "Identity";
this.Identity.Padding = new System.Windows.Forms.Padding(2);
this.Identity.Size = new System.Drawing.Size(355, 195);
this.Identity.TabIndex = 16;
this.Identity.TabStop = false;
this.Identity.Text = "Placeholder";
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.ClientSize = new System.Drawing.Size(1426, 795);
this.Controls.Add(this.Identity);
this.Controls.Add(this.NoteLog);
this.Controls.Add(this.Videostream);
this.Controls.Add(this.menuStrip1);
this.MainMenuStrip = this.menuStrip1;
this.Margin = new System.Windows.Forms.Padding(2);
this.MaximizeBox = false;
this.MinimizeBox = false;
this.Name = "Form1";
this.Text = "Senior Design UI";
this.FormClosing += new System.Windows.Forms.FormClosingEventHandler(this.Form1_FormClosing);
this.Load += new System.EventHandler(this.Form1_Load);
this.ResizeBegin += new System.EventHandler(this.Form1_ResizeBegin);
this.ResizeEnd += new System.EventHandler(this.Form1_ResizeBegin);
this.Resize += new System.EventHandler(this.Form1_ResizeBegin);
this.NoteLog.ResumeLayout(false);
this.NoteLog.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.NoteLog)).EndInit();
this.menuStrip1.ResumeLayout(false);
Listing 16.29: Code for the Designer of Primary Form of the User Interface

16.10  User Interface - Add Stream Interface
Listing 16.30: Code for the Add Stream Interface

16.11 User Interface - Add Stream Interface Designer File

262
namespace Senior_Design_UI
{
    partial class AddStream
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.button1 = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.urlTextBox = new System.Windows.Forms.TextBox();
            this.locTextBox = new System.Windows.Forms.TextBox();
            this.detTextBox = new System.Windows.Forms.TextBox();
            this.labelTestResults = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.button2 = new System.Windows.Forms.Button();
            this.cancelButton = new System.Windows.Forms.Button();
            this.label5 = new System.Windows.Forms.Label();
            this.nameTextBox = new System.Windows.Forms.TextBox();
            this.ThumbnailText = new System.Windows.Forms.TextBox();
            this.SuspendLayout();

            // button1
            //
            this.button1.Location = new System.Drawing.Point(11, 139);
            this.button1.Margin = new System.Windows.Forms.Padding(2);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(74, 19);
            this.button1.TabIndex = 0;
            this.button1.Text = "Test Stream";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Visible = false;
            this.button1.Click += new System.EventHandler(this.button1_Click);

            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(8, 37);
            this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(65, 13);
            this.label1.TabIndex = 1;
        }
        #endregion
    }
}
this.label1.Text = "Camera IP :
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(8, 63);
this.label2.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(57, 13);
this.label2.TabIndex = 2;
this.label2.Text = "Location :
//
// urlTextBox
//
this.urlTextBox.Location = new System.Drawing.Point(77, 34);
this.urlTextBox.Margin = new System.Windows.Forms.Padding(2);
this.urlTextBox.Name = "urlTextBox";
this.urlTextBox.Size = new System.Drawing.Size(162, 20);
this.urlTextBox.TabIndex = 3;
this.urlTextBox.Text = "192.168.11.32";
//
// locTextBox
//
this.locTextBox.Location = new System.Drawing.Point(69, 60);
this.locTextBox.Margin = new System.Windows.Forms.Padding(2);
this.locTextBox.Name = "locTextBox";
this.locTextBox.Size = new System.Drawing.Size(170, 20);
this.locTextBox.TabIndex = 4;
this.locTextBox.Text = "608C";
//
// detTextBox
//
this.detTextBox.Location = new System.Drawing.Point(60, 85);
this.detTextBox.Margin = new System.Windows.Forms.Padding(2);
this.detTextBox.Multiline = true;
this.detTextBox.Name = "detTextBox";
this.detTextBox.Size = new System.Drawing.Size(179, 49);
this.detTextBox.TabIndex = 5;
this.detTextBox.Text = "Bannon Engineering Outside facing benson quad ";
//
// labelTestResults
//
this.labelTestResults.AutoSize = true;
this.labelTestResults.Location = new System.Drawing.Point(91, 142);
this.labelTestResults.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
this.labelTestResults.Name = "labelTestResults";
this.labelTestResults.Size = new System.Drawing.Size(118, 13);
this.labelTestResults.TabIndex = 6;
this.labelTestResults.Text = "Test Results : Untested";
this.labelTestResults.Visible = false;
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(8, 88);
this.label4.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(48, 13);
this.label4.TabIndex = 7;
this.label4.Text = "Details : ";
//
// button2
//
this.button2.Location = new System.Drawing.Point(34, 211);
this.button2.Margin = new System.Windows.Forms.Padding(2);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(74, 19);
this.button2.TabIndex = 8;
this.button2.Text = "Add Stream";
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new System.EventHandler(this.StreamAddAccept);
//
// cancelButton
//
this.cancelButton.Location = new System.Drawing.Point(136, 211);
this.cancelButton.Margin = new System.Windows.Forms.Padding(2);
this.cancelButton.Name = "cancelButton";
this.cancelButton.Size = new System.Drawing.Size(65, 19);
this.cancelButton.TabIndex = 9;
this.cancelButton.Text = "Cancel";
this.cancelButton.UseVisualStyleBackColor = true;
this.cancelButton.Click += new System.EventHandler(this.StreamAddCancel);
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(8, 12);
this.label5.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(80, 13);
this.label5.TabIndex = 10;
this.label5.Text = "Stream Name : ";
//
// nameTextBox
//
this.nameTextBox.Location = new System.Drawing.Point(94, 9);
this.nameTextBox.Margin = new System.Windows.Forms.Padding(2);
this.nameTextBox.Name = "nameTextBox";
this.nameTextBox.Size = new System.Drawing.Size(145, 20);
this.nameTextBox.TabIndex = 11;
this.nameTextBox.Text = "Home";
//
// button3
//
this.button3.Location = new System.Drawing.Point(11, 163);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(228, 19);
this.button3.TabIndex = 12;
this.button3.UseVisualStyleBackColor = true;
this.button3.Click += new System.EventHandler(this.button3_Click);
//
// ThumbnailText
//
this.ThumbnailText.Location = new System.Drawing.Point(11, 187);
this.ThumbnailText.Name = "ThumbnailText";
this.ThumbnailText.ReadOnly = true;
this.ThumbnailText.Size = new System.Drawing.Size(228, 20);
this.ThumbnailText.TabIndex = 13;
//
// AddStream
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.CancelButton = this.cancelButton;
this.ClientSize = new System.Drawing.Size(247, 234);
this.Controls.Add(this.ThumbnailText);
this.Controls.Add(this.button3);
this.Controls.Add(this.nameTextBox);
this.Controls.Add(this.label5);
this.Controls.Add(this.cancelButton);
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.label2);
207 this.Controls.Add(this.label1);
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";
214 this.Text = "AddStream";
215 this.ResumeLayout(false);
216 this.PerformLayout();
217 }
218 #endregion

private System.Windows.Forms.Button button1;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.TextBox urlTextBox;
private System.Windows.Forms.TextBox locTextBox;
private System.Windows.Forms.TextBox detTextBox;
private System.Windows.Forms.Label labelTestResults;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.TextBox nameTextBox;
private System.Windows.Forms.TextBox ThumbnailText;

Listing 16.31: Code for the Designer of Add Stream Interface

16.12 User Interface - Event Object Class

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SeniorDesign_UI
{

class EventObject
{
    public EventObject()
    {
        individual = ";
        imgindex = 0;
        camera = 0;
        Xpos = 0;
        Ypos = 0;
    }

    public EventObject(String individ, Int32 cameraid, Int32 img, Int32 eventID,
    Int32 x, Int32 y)
    {
        individual = individ;
        camera = cameraid;
        imgindex = img;
    
}
Listing 16.32: Code for the Event Object Class

16.13 User Interface - RSTPStream Class

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SeniorDesign_UI
{
    class RSTPStream
    {
        public RSTPStream()
        {
            url = "";
            details = "";
```
public RSTPStream(String urls, String dis, String locs, Int32 name)
{
    url = urls;
    details = dis;
    loc = locs;
    imgIndex = name;
    //options = "showdisplay=false"; //Depricated
}

dictionary get_url()
{
    return url;
}

dictionary get_details()
{
    return details;
}

dictionary get_loc()
{
    return loc;
}

Int32 get_imgIndex()
{
    return imgIndex;
}

void set_url(String urls)
{
    url = urls;
}

void set_details(String dets)
{
    details = dets;
}

void set_loc(String locs)
{
    loc = locs;
}

void set_imgIndex(Int32 img)
{
    imgIndex = img;
}

// Variables
String url, name, details, loc;
Int32 imgIndex;
}

Listing 16.33: Code for the RSTPStream Class

16.14 User Interface - Database Configuration File
using System.Data.Linq;
using System.Data;
using System.Collections.Generic;
using System.Reflection;
using System.Linq;
using System.Linq.Expressions;
using System.ComponentModel;
using System;

namespace Senior.Design.UI
{
    [global::System.Data.Linq.Mapping.DatabaseAttribute(Name="SeniorDesignDB2")]
    {

        #region Extensibility Method Definitions
        partial void OnCreated();
        partial void InsertCamera(Camera instance);
        partial void UpdateCamera(Camera instance);
        partial void DeleteCamera(Camera instance);
        partial void InsertCameraStat(CameraStat instance);
        partial void UpdateCameraStat(CameraStat instance);
        partial void DeleteCameraStat(CameraStat instance);
        partial void InsertDetectionEvent(DetectionEvent instance);
        partial void UpdateDetectionEvent(DetectionEvent instance);
        partial void DeleteDetectionEvent(DetectionEvent instance);
        partial void InsertIndividual(Individual instance);
        partial void UpdateIndividual(Individual instance);
        partial void DeleteIndividual(Individual instance);
        partial void InsertKnownImg(KnownImg instance);
        partial void UpdateKnownImg(KnownImg instance);
        partial void DeleteKnownImg(KnownImg instance);
        partial void InsertAnomaly(Anomaly instance);
        partial void UpdateAnomaly(Anomaly instance);
        partial void DeleteAnomaly(Anomaly instance);
        #endregion

        public SDDBDataContext() :
            base(global::Senior_Design_UI.Properties.Settings.Default.
                SeniorDesignDB2ConnectionString, mappingSource)
        {
            OnCreated();
        }

        public SDDBDataContext(string connection) :
            base(connection, mappingSource)
        {
            OnCreated();
        }

        public SDDBDataContext(System.Data.IDbConnection connection) :
            base(connection, mappingSource)
        {
        }
    }
}
{
    base(connection, mappingSource);
    OnCreated();
}

{
    base(connection, mappingSource);
    OnCreated();
}

public System.Data.Linq.Table<Camera> Cameras
{
    get
    {
        return this.GetTable<Camera>();
    }
}

public System.Data.Linq.Table<CameraStat> CameraStats
{
    get
    {
        return this.GetTable<CameraStat>();
    }
}

public System.Data.Linq.Table<DetectionEvent> DetectionEvents
{
    get
    {
        return this.GetTable<DetectionEvent>();
    }
}

public System.Data.Linq.Table<Individual> Individuals
{
    get
    {
        return this.GetTable<Individual>();
    }
}

public System.Data.Linq.Table<KnownImg> KnownImgs
{
    get
    {
        return this.GetTable<KnownImg>();
    }
}

public System.Data.Linq.Table<Anomaly> Anomalies
{
    get
    {
        return this.GetTable<Anomaly>();
    }
}

[DllImport("System.Data.Linq", CallingConvention = CallingConvention.Cdecl)]
public partial class Camera : INotifyPropertyChanging, INotifyPropertyChanged
private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

private short _CameraID;
private string _CameraName;
private string _CameraIP;
private string _FPGAIP;
private string _CameraLoc;
private string _CameraDetails;
private System.Nullable<short>_imgIndex;
private string _ThumbPath;
private EntitySet<CameraStat> _CameraStats;
private EntitySet<DetectionEvent> _DetectionEvents;

#region Extensibility Method Definitions
partial void OnLoaded();
partial void OnValidate(System.Data.Linq.ChangeAction action);
partial void OnCreated();
partial void OnCameraIDChanging(short value);
partial void OnCameraIDChanged();
partial void OnCameraNameChanging(string value);
partial void OnCameraNameChanged();
partial void OnCameraIPChanging(string value);
partial void OnCameraIPChanged();
partial void OnFPGAIPChanging(string value);
partial void OnFPGAIPChanged();
partial void OnCameraLocChanging(string value);
partial void OnCameraLocChanged();
partial void OnCameraDetailsChanging(string value);
partial void OnCameraDetailsChanged();
partial void OnImgIndexChanging(System.Nullable<short> value);
partial void OnImgIndexChanged();
partial void OnThumbPathChanging(string value);
partial void OnThumbPathChanged();
#endregion

public Camera()
{
    this._CameraStats = new EntitySet<CameraStat>(new Action<CameraStat>(this.attach_CameraStats), new Action<CameraStat>(this.detach_CameraStats));
    this._DetectionEvents = new EntitySet<DetectionEvent>(new Action<DetectionEvent>(this.attach_DetectionEvents), new Action<DetectionEvent>(this.detach_DetectionEvents));
    OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraID", DbType="SmallInt NOT NULL", IsPrimaryKey=true)]
public short CameraID
{
    get
    {
        return this._CameraID;
    }
    set
    {
        if ((this._CameraID != value))
        {
            // Set new value
            this._CameraID = value;
            // Notify change
            OnCameraIDChanging(EventArgs.Empty);
        }
    }
}
{  
    this.OnCameraIDChanging(value);
    this.SendPropertyChanging();
    this.CameraID = value;
    this.SendPropertyChanged("CameraID");
    this.OnCameraIDChanged();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraName", DbType="VarChar(20) NOT NULL", CanBeNull=false)]
public string CameraName
{
    get
    {
        return this._CameraName;
    }
    set
    {
        if ((this._CameraName != value))
        {
            this.OnCameraNameChanging(value);
            this.SendPropertyChanging();
            this._CameraName = value;
            this.SendPropertyChanged("CameraName");
            this.OnCameraNameChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraIP", DbType="VarChar(40)")]
public string CameraIP
{
    get
    {
        return this._CameraIP;
    }
    set
    {
        if ((this._CameraIP != value))
        {
            this.OnCameraIPChanging(value);
            this.SendPropertyChanging();
            this._CameraIP = value;
            this.SendPropertyChanged("CameraIP");
            this.OnCameraIPChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FPGAIP", DbType="VarChar(40)")]
public string FPGAIP
{
    get
    {
        return this._FPGAIP;
    }
    set
    {
        if ((this._FPGAIP != value))
        {
            this.OnFPGAIPChanging(value);
            this.SendPropertyChanging();
            this._FPGAIP = value;
            this.SendPropertyChanged("FPGAIP");
        }
    }
}
public string CameraLoc
{
    get
    {
        return this._CameraLoc;
    }
    set
    {
        if ((this._CameraLoc != value))
        {
            this.OnCameraLocChanging(value);
            this.SendPropertyChanging();
            this._CameraLoc = value;
            this.SendPropertyChanged("CameraLoc");
            this.OnCameraLocChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CameraDetails", DbType="VarChar(100)")]
public string CameraDetails
{
    get
    {
        return this._CameraDetails;
    }
    set
    {
        if ((this._CameraDetails != value))
        {
            this.OnCameraDetailsChanging(value);
            this.SendPropertyChanging();
            this._CameraDetails = value;
            this.SendPropertyChanged("CameraDetails");
            this.OnCameraDetailsChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_imgIndex", DbType="SmallInt")]
public System.Nullable<short> imgIndex
{
    get
    {
        return this._imgIndex;
    }
    set
    {
        if ((this._imgIndex != value))
        {
            this.OnimgIndexChanging(value);
            this.SendPropertyChanging();
            this._imgIndex = value;
            this.SendPropertyChanged("imgIndex");
            this.OnimgIndexChanged();
        }
    }
}
```csharp
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ThumbPath", DbType="VarChar(10)")]
public string ThumbPath
{
    get
    {
        return this._ThumbPath;
    }
    set
    {
        if ((this._ThumbPath != value))
        {
            this.OnThumbPathChanging(value);
            this.SendPropertyChanging();
            this._ThumbPath = value;
            this.SendPropertyChanged("ThumbPath");
            this.OnThumbPathChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat", Storage="_CameraStats", ThisKey="CameraID", OtherKey="CameraID")]
public EntitySet<CameraStat> CameraStats
{
    get
    {
        return this._CameraStats;
    }
    set
    {
        this._CameraStats.Assign(value);
    }
}

public EntitySet<DetectionEvent> DetectionEvents
{
    get
    {
        return this._DetectionEvents;
    }
    set
    {
        this._DetectionEvents.Assign(value);
    }
}

public event PropertyChangingEventHandler PropertyChanging;
public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()
{
    if ((this.PropertyChanging != null))
    {
        this.PropertyChanging(this, emptyChangingEventArgs);
    }
}

protected virtual void SendPropertyChanged(String propertyName)
{
    if ((this.PropertyChanged != null))
    {
        this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}
```
private void attach_CameraStats(CameraStat entity)
{
    this.SendPropertyChanging();
    entity.Camera = this;
}

private void detach_CameraStats(CameraStat entity)
{
    this.SendPropertyChanging();
    entity.Camera = null;
}

private void attach_DetectionEvents(DetectionEvent entity)
{
    this.SendPropertyChanging();
    entity.Camera = this;
}

private void detach_DetectionEvents(DetectionEvent entity)
{
    this.SendPropertyChanging();
    entity.Camera = null;
}

[global::System.Data.Linq.Mapping.ToTable(Name="dbo.CameraStats")]
public partial class CameraStat : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

    private short _TimeID;
    private short _CameraID;
    private Nullable<System.TimeSpan> _LoTimeRange;
    private Nullable<System.TimeSpan> _HiTimeRange;
    private Nullable<double> _Simga;
    private Nullable<double> _Mean;
    private Nullable<int> _NValue;
    private EntityRef<Camera> _Camera;

    #region Extensibility Method Definitions
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
    partial void OnTimeIDChanging(short value);
    partial void OnTimeIDChanged();
    partial void OnCameraIDChanging(short value);
    partial void OnCameraIDChanged();
    partial void OnLoTimeRangeChanging(System.Nullable<System.TimeSpan> value);
    partial void OnLoTimeRangeChanged();
    partial void OnHiTimeRangeChanging(System.Nullable<System.TimeSpan> value);
    partial void OnHiTimeRangeChanged();
    partial void OnSimgaChanging(System.Nullable<double> value);
    partial void OnSimgaChanged();
    partial void OnMeanChanging(System.Nullable<double> value);
    partial void OnMeanChanged();
    partial void OnNValueChanged();
    partial void OnNValueChanging(System.Nullable<int> value);
    partial void OnNValueChanged();
    #endregion
}
public CameraStat()
{
    this._Camera = default(EntityRef<Camera>);
    OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_TimeID", DbType="SmallInt NOT NULL", IsPrimaryKey=true)]
public short TimeID
{
    get
    {
        return this._TimeID;
    }
    set
    {
        if ((this._TimeID != value))
        {
            this.OnTimeIDChanging(value);
            this.SendPropertyChanging();
            this._TimeID = value;
            this.SendPropertyChanged("TimeID");
            this.OnTimeIDChanged();
        }
    }
}

public short CameraID
{
    get
    {
        return this._CameraID;
    }
    set
    {
        if ((this._CameraID != value))
        {
            if (this._Camera.HasLoadedOrAssignedValue)
            {
                throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnCameraIDChanging(value);
            this.SendPropertyChanging();
            this._CameraID = value;
            this.SendPropertyChanged("CameraID");
            this.OnCameraIDChanged();
        }
    }
}

public System.Nullable<System.TimeSpan> LoTimeRange
{
    get
    {
        return this._LoTimeRange;
    }
    set
    {
        if ((this._LoTimeRange != value))
        {
            this.OnLoTimeRangeChanging(value);
            this.SendPropertyChanging();
            this._LoTimeRange = value;
        }
    }
}
this.SendPropertyChanged("LoTimeRange");
this.OnLoTimeRangeChanged();

public System.Nullable<System.TimeSpan> HiTimeRange
{
    get
    {
        return this._HiTimeRange;
    }
    set
    {
        if ((this._HiTimeRange != value))
        {
            this.OnHiTimeRangeChanging(value);
            this.SendPropertyChanging();
            this._HiTimeRange = value;
            this.SendPropertyChaged("HiTimeRange");
            this.OnHiTimeRangeChanged();
        }
    }
}

public System.Nullable<double> Simgag
{
    get
    {
        return this._Simga;
    }
    set
    {
        if ((this._Simga != value))
        {
            this.OnSimgagChanging(value);
            this.SendPropertyChanging();
            this._Simga = value;
            this.SendPropertyChaged("Simgag");
            this.OnSimgagChaged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="Mean", DbType="Float")]
public System.Nullable<double> Mean
{
    get
    {
        return this._Mean;
    }
    set
    {
        if ((this._Mean != value))
        {
            this.OnMeanChanging(value);
            this.SendPropertyChanging();
            this._Mean = value;
            this.SendPropertyChaged("Mean");
            this.OnMeanChaged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="NValue", DbType="Int")]

277
```csharp
public System.Nullable<int> NValue
{
    get
    {
        return this._NValue;
    }
    set
    {
        if ((this._NValue != value))
        {
            this.OnNValueChanging(value);
            this.SendPropertyChanging();
            this._NValue = value;
            this.SendPropertyChanged("NValue");
            this.OnNValueChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_CameraStat", Storage="_Camera", ThisKey="CameraID", OtherKey="CameraID", IsForeignKey=true, DeleteOnNull=true, DeleteRule="CASCADE")]
public Camera Camera
{
    get
    {
        return this._Camera.Entity;
    }
    set
    {
        Camera previousValue = this._Camera.Entity;
        if (((previousValue != value) || (this._Camera.HasLoadedOrAssignedValue == false)))
        {
            this.SendPropertyChanging();
            if ((previousValue != null))
            {
                this._Camera.Entity = null;
                previousValue.CameraStats.Remove(this);
            }
            this._Camera.Entity = value;
            if ((value != null))
            {
                value.CameraStats.Add(this);
                this._CameraID = value.CameraID;
            }
            else
            {
                this._CameraID = default(short);
            }
            this.SendPropertyChanged("Camera");
        }
    }
}

public event PropertyChangingEventHandler PropertyChanging;
public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()
{
    if ((this.PropertyChanging != null))
    {
        this.PropertyChanging(this, emptyChangingEventArgs);
    }
}

protected virtual void SendPropertyChanged(String propertyName)
```

278
if ((this.PropertyChanged != null))
    this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
}
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.DetectionEvent")]
public partial class DetectionEvent : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);
    private int _EventID;
    private short _EventCamera;
    private System.TimeSpan _EventTimeIn;
    private System.Nullable<System.TimeSpan> _EventTimeOut;
    private System.Nullable<int> _isChecked;
    private System.Nullable<short> _EntryType;
    private System.Nullable<short> _ExitType;
    private int _SuspectID;
    private double _Confidence;
    private EntitySet<Anomaly> _Anomalies;
    private EntityRef<Camera> _Camera;
    private EntityRef<Individual> _Individual;

    #region Extensibility Method Definitions
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
    partial void OnEventIDChanging(int value);
    partial void OnEventIDChanged();
    partial void OnEventCameraChanging(short value);
    partial void OnEventCameraChanged();
    partial void OnEventTimeInChanging(System.TimeSpan value);
    partial void OnEventTimeInChanged();
    partial void OnEventTimeOutChanging(System.Nullable<System.TimeSpan> value);
    partial void OnEventTimeOutChanged();
    partial void OnIsCheckedChanging(System.Nullable<int> value);
    partial void OnIsCheckedChanged();
    partial void OnEntryTypeChanging(System.Nullable<short> value);
    partial void OnEntryTypeChanged();
    partial void OnExitTypeChanging(System.Nullable<short> value);
    partial void OnExitTypeChanged();
    partial void OnSuspectIDChanging(int value);
    partial void OnSuspectIDChanged();
    partial void OnConfidenceChanging(double value);
    partial void OnConfidenceChanged();
    #endregion

    public DetectionEvent()
    {
        this._Anomalies = new EntitySet<Anomaly>((new Action<Anomaly>(this.attachAnomalies),
            new Action<Anomaly>(this.detachAnomalies));
        this._Camera = default(EntityRef<Camera>);
Individual = default(EntityRef<Individual>);
OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EventID", DbType="Int NOT NULL", IsPrimaryKey=true)]
public int EventID
{
    get
    {
        return this._EventID;
    }
    set
    {
        if ((this._EventID != value))
        {
            this.OnEventIDChanging(value);
            this.SendPropertyChanging();
            this._EventID = value;
            this.SendPropertyChanged("EventID");
            this.OnEventIDChanged();
        }
    }
}

public short EventCamera
{
    get
    {
        return this._EventCamera;
    }
    set
    {
        if ((this._EventCamera != value))
        {
            if (this._Camera.HasLoadedOr_assignedValue)
            |
                throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnEventCameraChanging(value);
            this.SendPropertyChanging();
            this._EventCamera = value;
            this.SendPropertyChanged("EventCamera");
            this.OnEventCameraChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EventTimeIn", DbType="Time NOT NULL")]}
public System.TimeSpan EventTimeIn
{
    get
    {
        return this._EventTimeIn;
    }
    set
    {
        if ((this._EventTimeIn != value))
        {
            this.OnEventTimeInChanging(value);
            this.SendPropertyChanging();
            this._EventTimeIn = value;
            this.SendPropertyChanged("EventTimeIn");
            this.OnEventTimeInChanged();
        }
    }
}
public Nullable<System.TimeSpan> EventTimeOut
{
    get
    {
        return _EventTimeOut;
    }
    set
    {
        if ((_EventTimeOut != value))
        {
            _EventTimeOut = value;
        }
    }
}

public Nullable<int> isChecked
{
    get
    {
        return _isChecked;
    }
    set
    {
        if ((_isChecked != value))
        {
            _isChecked = value;
        }
    }
}

public Nullable<short> EntryType
{
    get
    {
        return _EntryType;
    }
    set
    {
        if ((_EntryType != value))
        {
            _EntryType = value;
        }
    }
}

public Nullable<short> ExitType

get
{
    return this._ExitType;
}

set
{
    if ((this._ExitType != value))
    {
        this.OnExitTypeChanging(value);
        this.SendPropertyChanging();
        this._ExitType = value;
        this.SendPropertyChanged("ExitType");
        this.OnExitTypeChanged();
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_SuspectID", DbType="Int NOT NULL")]
public int SuspectID
{
    get
    {
        return this._SuspectID;
    }
    set
    {
        if ((this._SuspectID != value))
        {
            if (this._Individual.HasLoadedOrAssignedValue)
            {
                throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnSuspectIDChanging(value);
            this.SendPropertyChanging();
            this._SuspectID = value;
            this.SendPropertyChanged("SuspectID");
            this.OnSuspectIDChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Confidence", DbType="Float NOT NULL")]
public double Confidence
{
    get
    {
        return this._Confidence;
    }
    set
    {
        if ((this._Confidence != value))
        {
            this.OnConfidenceChanging(value);
            this.SendPropertyChanging();
            this._Confidence = value;
            this.SendPropertyChanged("Confidence");
            this.OnConfidenceChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="DetectionEvent_Anomaly", Storage="_Anomalies", ThisKey="EventID", OtherKey="EventID")]
public EntitySet<Anomaly> Anomalies
{
get
{
    return this._Anomalies;
}

set
{
    this._Anomalies.Assign(value);
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Camera_DetectionEvent", Storage="_Camera", ThisKey="EventCamera", OtherKey="CameraID", IsForeignKey=true, DeleteOnNull=true, DeleteRule="CASCADE")]
public Camera Camera
{
    get
    {
        return this._Camera.Entity;
    }
    set
    {
        Camera previousValue = this._Camera.Entity;
        if (((previousValue != value) || (this._Camera.HasLoadedOrAssignedValue == false)))
        {
            this.SendPropertyChanging();
            if ((previousValue != null))
            {
                this._Camera.Entity = null;
                previousValue.DetectionEvents.Remove(this);
            }
            this._Camera.Entity = value;
            if ((value != null))
            {
                value.DetectionEvents.Add(this);
                this._EventCamera = value.CameraID;
            }
        }
        else
        {
            this._EventCamera = default(short);
        }
        this.SendPropertyChanged("Camera");
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Individual_DetectionEvent", Storage="_Individual", ThisKey="SuspectID", OtherKey="ID", IsForeignKey=true, DeleteOnNull=true, DeleteRule="CASCADE")]
public Individual Individual
{
    get
    {
        return this._Individual.Entity;
    }
    set
    {
        Individual previousValue = this._Individual.Entity;
        if (((previousValue != value) || (this._Individual.HasLoadedOrAssignedValue == false)))
        {
            this.SendPropertyChanging();
            if ((previousValue != null))
            {
                this._Individual.Entity = null;
                previousValue.DetectionEvents.Remove(this);
            }
            this._Individual.Entity = value;
        }
    }
}
if ((value != null)) {
    value.DetectionEvents.Add(this);
    this._SuspectID = value.ID;
} else {
    this._SuspectID = default(int);
} this.SendPropertyChanged("Individual");
}
public event PropertyChangingEventHandler PropertyChanging;
public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging() {
    if ((this.PropertyChanging != null)) {
        this.PropertyChanging(this, emptyChangingEventArgs);
    }
}
protected virtual void SendPropertyChanged(String propertyName) {
    if ((this.PropertyChanged != null)) {
        this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}

private void attachAnomalies(Anomaly entity) {
    this.SendPropertyChanging();
    entity.DetectionEvent = this;
}
private void detachAnomalies(Anomaly entity) {
    this.SendPropertyChanging();
    entity.DetectionEvent = null;
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Individual")]
public partial class Individual : INotifyPropertyChanging, INotifyPropertyChanged {
    private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);
    private int _ID;
    private string _Name;
    private string _Details;
    private EntitySet<DetectionEvent> _DetectionEvents;
    private EntitySet<KnownImg> _KnownImgs;
    #region Extensibility Method Definitions
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
}
Partial void OnIDChanging (int value);
Partial void OnIDChanged ();
Partial void OnNameChanging (string value);
Partial void OnNameChanged ();
Partial void OnDetailsChanging (string value);
Partial void OnDetailsChanged ();
#endregion

public Individual ()
{
    this.DetectionEvents = new EntitySet<DetectionEvent>(
        new Action<DetectionEvent>(this.attach_DetectionEvents),
        new Action<DetectionEvent>(this.detach_DetectionEvents));
    this.KnownImgs = new EntitySet<KnownImg>(
        new Action<KnownImg>(this.attach_KnownImgs),
        new Action<KnownImg>(this.detach_KnownImgs));
    OnCreated ();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="ID", DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true, IsDbGenerated=true)]
public int ID
{
    get
    {
        return this._ID;
    }
    set
    {
        if ((this._ID != value))
        {
            this.OnIDChanging (value);
            this.SendPropertyChanging ();
            this._ID = value;
            this.SendPropertyChanged("ID");
            this.OnIDChanged ();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="Name", DbType="VarChar(40) NOT NULL", CanBeNull=false)]
public string Name
{
    get
    {
        return this._Name;
    }
    set
    {
        if ((this._Name != value))
        {
            this.OnNameChanging (value);
            this.SendPropertyChanging ();
            this._Name = value;
            this.SendPropertyChanged("Name");
            this.OnNameChanged ();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="Details", DbType="VarChar(100)")]
public string Details
{
    get
    {
        return this._Details;
    }
}
```csharp
    set
    {
        if ((this._Details != value))
        {
            this.OnDetailsChanging(value);
            this.SendPropertyChanging();
            this._Details = value;
            this.SendPropertyChanged("Details");
            this.OnDetailsChanged();
        }
    }

    [global::System.Data.Linq.Mapping.AssociationAttribute(Name="Individual_DetectionEvent", Storage="_DetectionEvents", ThisKey="ID", OtherKey="SuspectID")]
    public EntitySet<DetectionEvent> DetectionEvents
    {
        get
        {
            return this._DetectionEvents;
        }
        set
        {
            this._DetectionEvents.Assign(value);
        }
    }

    [global::System.Data.Linq.Mapping.AssociationAttribute(Name="IndividualKNOWN_IMG", Storage="_KnownImgs", ThisKey="ID", OtherKey="IDIMG")]
    public EntitySet<KnownImg> KnownImgs
    {
        get
        {
            return this._KnownImgs;
        }
        set
        {
            this._KnownImgs.Assign(value);
        }
    }

    public event PropertyChangingEventHandler PropertyChanging;
    public event PropertyChangedEventHandler PropertyChanged;

    protected virtual void SendPropertyChanging()
    {
        if ((this.PropertyChanging != null))
        {
            this.PropertyChanging(this, emptyChangingEventArgs);
        }
    }

    protected virtual void SendPropertyChanged(String propertyName)
    {
        if ((this.PropertyChanged != null))
        {
            this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
        }
    }

    private void attach_DetectionEvents(DetectionEvent entity)
    {
        this.SendPropertyChanging();
        entity.Individual = this;
    }

    private void detach_DetectionEvents(DetectionEvent entity)
```
private void attach_KnownImg(KnownImg entity)
{
    this.SendPropertyChanging();
    entity.Individual = this;
}

private void detach_KnownImg(KnownImg entity)
{
    this.SendPropertyChanging();
    entity.Individual = null;
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.KnownImg")]
public partial class KnownImg : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new
        PropertyChangingEventArgs(String.Empty);
    private int _KnownImgID;
    private int _IDImg;
    private System.Data.Linq.Binary _Img;
    private EntityRef<Individual> _Individual;

    #region Extensibility Method Definitions
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
    partial void OnKnownImgIDChanging(int value);
    partial void OnKnownImgIDChanged();
    partial void OnIDImgChanging(int value);
    partial void OnIDImgChanged();
    partial void OnImgChanging(System.Data.Linq.Binary value);
    partial void OnImgChanged();
    #endregion

    public KnownImg() {
        this._Individual = default(EntityRef<Individual>);
        OnCreated();
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_KnownImgID", AutoSync=
        AutoSync.OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true, 
        IsDbGenerated=true)]
    public int KnownImgID
    {
        get
        {
            return this._KnownImgID;
        }
        set
        {
            if ((this._KnownImgID != value))
            {
                this.OnKnownImgIDChanging(value);
                this.SendPropertyChanging();
                this._KnownImgID = value;
                this.SendPropertyChanged("KnownImgID");
            }
        }
    }
}
        this.OnKnownImgIDChanged();
    }
    
} [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_IDImg", DbType="Int NOT NULL")]

public int IDImg
{
    get
    {
        return this._IDImg;
    }
    
    set
    {
        if ((this._IDImg != value))
        {
            if (this._Individual.HasLoadedOrAssignedValue)
            {
                throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnIDImgChanging(value);
            this.SendPropertyChanging();
            this._IDImg = value;
            this.SendPropertyChanged("IDImg");
            this.OnIDImgChanged();
        }
    }
}


public System.Data.Linq.Binary Img
{
    get
    {
        return this._Img;
    }
    
    set
    {
        if ((this._Img != value))
        {
            this.OnImgChanging(value);
            this.SendPropertyChanging();
            this._Img = value;
            this.SendPropertyChanged("Img");
            this.OnImgChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="Individual_KnownImg", Storage="_Individual", ThisKey="IDImg", OtherKey="ID", IsForeignKey=true, DeleteOnNull=true, DeleteRule="CASCADE")]

public Individual Individual
{
    get
    {
        return this._Individual.Entity;
    }
    
    set
    {
        Individual previousValue = this._Individual.Entity;
        if (((previousValue != value) || (this._Individual.HasLoadedOrAssignedValue == false)))
        {
            this.SendPropertyChanging();
            if ((previousValue != null))
            {
this._Individual.Entity = null;
previousValue.Knownimgs.Remove(this);
if ((value != null))
    value.Knownimgs.Add(this);
this._IDEv = value.ID;
else
    this._IDEv = default(int);
this.SendPropertyChanged("Individual");
}

public event PropertyChangingEventHandler PropertyChanging;
public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()
{
    if ((this.PropertyChanging != null))
        this.PropertyChanging(this, emptyChangingEventArgs);
}

protected virtual void SendPropertyChanged(String propertyName)
{
    if ((this.PropertyChanged != null))
        this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
}

[global::System.Data.Linq.Mapping.TableName(Name="dbo.Anomaly")]
public partial class Anomaly : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArg emptyChangingEventArgs = new
PropertyChangingEventArg(String.Empty);

    private int _AnomalyID;
    private int _EventID;
    private Nullable<double> _Severity;
    private Nullable<short> _isChecked;
    private Nullable<short> _AnomalyType;
    private EntityRef<DetectionEvent> _DetectionEvent;

    #region Extensibility Method Definitions
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
    partial void OnAnomalyIDChanging(int value);
    partial void OnAnomalyIDChanged();
    partial void OnEventIDChanging(int value);
    partial void OnEventIDChanged();
    partial void OnSeverityChanging(System.Nullable<double> value);
partial void OnSeverityChanged();
partial void OnisCheckedChanging(System.Nullable<short> value);
partial void OnisCheckedChanged();
partial void OnAnomalyTypeChanging(System.Nullable<short> value);
partial void OnAnomalyTypeChanged();
#endregion

public Anomaly()
{
    this._DetectionEvent = default(EntityRef<DetectionEvent>);
    OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_AnomalyID", AutoSync=AutoSync.OnInsert, DbType="Int NOT NULL IDENTITY", IsPrimaryKey=true, IsDbGenerated=true)]
public int AnomalyID
{
    get
    {
        return this._AnomalyID;
    }
    set
    {
        if ((this._AnomalyID != value))
        {
            this.OnAnomalyIDChanging(value);
            this.SendPropertyChanging();
            this._AnomalyID = value;
            this.SendPropertyChanged("AnomalyID");
            this.OnAnomalyIDChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_EventID", DbType="Int NOT NULL")]
public int EventID
{
    get
    {
        return this._EventID;
    }
    set
    {
        if ((this._EventID != value))
        {
            if (this._DetectionEvent.HasLoadedOrAssignedValue)
            {
                throw new System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnEventIDChanging(value);
            this.SendPropertyChanging();
            this._EventID = value;
            this.SendPropertyChanged("EventID");
            this.OnEventIDChanged();
        }
    }
}

public System.Nullable<double> Severity
{
    get
    {
        return this._Severity;
    }
}
```csharp
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     [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="is Checked", DbType="SmallInt")]
1450     public System.Nullable<short> isChecked
1451     {
1452         get
1453         {
1454             return this._isChecked;
1455         }
1456         set
1457         {
1458             if ((this._isChecked != value))
1459             {
1460                 this.OnisCheckedChanging(value);
1461                 this.SendPropertyChanging();
1462                 this._isChecked = value;
1463                 this.SendPropertyChanged("isChecked");
1464                 this.OnisCheckedChanged();
1465             }
1466         }
1467         [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="AnomalyType", DbType="SmallInt")]
1468         public System.Nullable<short> AnomalyType
1469         {
1470             get
1471             {
1472                 return this._AnomalyType;
1473             }
1474             set
1475             {
1476                 if ((this._AnomalyType != value))
1477                 {
1478                     this.OnAnomalyTypeChanging(value);
1479                     this.SendPropertyChanging();
1480                     this._AnomalyType = value;
1481                     this.SendPropertyChanged("AnomalyType");
1482                     this.OnAnomalyTypeChanged();
1483                 }
1484             }
1485         }
1487         public DetectionEvent DetectionEvent
1488         {
1489             get
1490             {
1491                 return this._DetectionEvent.Entity;
1492             }
1493             set
1494             {
1495                 DetectionEvent previousValue = this._DetectionEvent.Entity;
1496                 if (((previousValue != value)
1497                     ...```
|| (this.
| DetectionEvent.
| HasLoadedOrAssignedValue == false)))
| this.SendPropertyChanging();
| if ((previousValue != null))
| this.
| DetectionEvent.
| Entity = null;
| previousValue.
| Anomalies.Remove(this);
| this.
| DetectionEvent.
| Entity = value;
| if ((value != null))
| value.
| Anomalies.Add(this);
| this.
| _EventID = value.
| EventID;
| else
| this.
| _EventID = default(int);
| this.
| SendPropertyChanged("DetectionEvent");
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
| public event PropertyChangingEventHandler PropertyChanging;
| public event PropertyChangedEventHandler PropertyChanged;
| protected virtual void SendPropertyChanging()
| { if ((this.PropertyChanging != null))
| { this.PropertyChanging(this, emptyChangingEventArgs);
| }
| protected virtual void SendPropertyChanged(String propertyName)
| { if ((this.PropertyChanged != null))
| { this.PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
| }
| }
| #pragma warning restore 1591
|

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

16.15 Identity Engine - Matlab Demo Code

function [name, cam, frameNum, action] = identity_engine(frame, camera, xpos, ypos, h, w, img)
% input = struct {
% int objID,
% int frame,
% int camera,
% int xpos,
% int ypos,
% int h,
% int w,
% image img }
persistent prevIDs;
persistent lastCam;
persistent lastFrame;
persistent lastPos;
delay = 6;

% read image
figure
subplot(2,2,1)
imshow(img);

% create grayscale image
g_img = rgb2gray(img);
subplot(2,2,2)
imshow(g_img);

% edge detection
edgy_img = edge(g_img, 'canny');
subplot(2,2,3)
imshow(edgy_img);

% find target area
x = w;
y = h;

min_x = x;
min_y = y;
max_x = 1;
max_y = 1;

% find max and min x and y coordinates
for j = 1:y
    for i = 1:x
        if edgy_img(i,j) == 1
            if i < min_x
                min_x = i;
            elseif i > max_x
                max_x = i;
            end
            if j < min_y
                min_y = j;
            elseif j > max_y
                max_y = j;
            end
        end
    end
end

% MATLAB demo only
demo_img = img;

% find outer radius
r = (abs(max_x-min_x)/2 + abs(max_y-min_y)/2)/2;
r_out = 0.69 * r;
r_mid_out = 0.49 * r;
r_mid_in = 0.45 * r;
r_in = 0.25 * r;

for i = 1:x
    for j = 1:y
        p = sqrt((i-(x/2))^2 + (j-(y/2))^2);
        if p <= (r_out * 1.01) && p >= (r_out * 0.99)
            demo_img(i, j, 1) = 255;
demo_img(i, j, 2) = 140;
demo_img(i, j, 3) = 0;
        elseif p <= (r_mid_out * 1.01) && p >= (r_mid_out * 0.99)
            demo_img(i, j, 1) = 255;
demo_img(i, j, 2) = 140;
demo_img(i, j, 3) = 0;
        elseif p <= (r_mid_in * 1.01) && p >= (r_mid_in * 0.99)
demo_img(i, j, 1) = 255;
demo_img(i, j, 2) = 140;
demo_img(i, j, 3) = 0;
else if p <= (r_in * 1.01) && p >= (r_in * 0.99)
    demo_img(i, j, 1) = 255;
demo_img(i, j, 2) = 140;
demo_img(i, j, 3) = 0;
end
end

subplot(2,2,4)
imshow(demo_img);

% ID color bands
% vars for each band: pixel count, sum, avg
band1_pix = 0;
bnd1_sum = [0, 0, 0];
bnd1_avg = [0, 0, 0];

band2_pix = 0;
bnd2_sum = [0, 0, 0];
bnd2_avg = [0, 0, 0];

for j = 1:y
    for i = 1:x
        p = sqrt(((i-(x/2))^2 + (j-(y/2))^2);
        % inside band 1
        if p <= r_out && p >= r_mid_out
            band1_pix = band1_pix + 1;
            bnd1_sum(1) = bnd1_sum(1) + cast(img(i,j,1), 'uint32');
            bnd1_sum(2) = bnd1_sum(2) + cast(img(i,j,2), 'uint32');
            bnd1_sum(3) = bnd1_sum(3) + cast(img(i,j,3), 'uint32');
        end
        % inside band 2
        elseif p <= r_mid_in && p >= r_in
            band2_pix = band2_pix + 1;
            bnd2_sum(1) = bnd2_sum(1) + cast(img(i,j,1), 'uint32');
            bnd2_sum(2) = bnd2_sum(2) + cast(img(i,j,2), 'uint32');
            bnd2_sum(3) = bnd2_sum(3) + cast(img(i,j,3), 'uint32');
        end
    end
end

bnd1_avg(1) = cast(ceil(bnd1_sum(1) / bnd1_pix), 'uint8');
bnd1_avg(2) = cast(ceil(bnd1_sum(2) / bnd1_pix), 'uint8');
bnd1_avg(3) = cast(ceil(bnd1_sum(3) / bnd1_pix), 'uint8');
bnd2_avg(1) = cast(ceil(bnd2_sum(1) / bnd2_pix), 'uint8');
bnd2_avg(2) = cast(ceil(bnd2_sum(2) / bnd2_pix), 'uint8');
bnd2_avg(3) = cast(ceil(bnd2_sum(3) / bnd2_pix), 'uint8');

bnd1 = [0, 0, 0];
bnd2 = [0, 0, 0];

% threshold
for i = 1:3
    if bnd1_avg(i) <= 32
        bnd1(i) = 0;
    elseif bnd1_avg(i) > 32 && bnd1_avg(i) <= 96
        bnd1(i) = 64;
    elseif bnd1_avg(i) > 96 && bnd1_avg(i) <= 160
        bnd1(i) = 128;
    elseif bnd1_avg(i) > 160 && bnd1_avg(i) <= 224
        bnd1(i) = 192;
    elseif bnd1_avg(i) > 224
        bnd1(i) = 255;
    end
end

if bnd2_avg(i) <= 32
band2(i) = 0;
elseif band2_avg(i) > 32 && band2_avg(i) <= 96
    band2(i) = 64;
elseif band2_avg(i) > 96 && band2_avg(i) <= 160
    band2(i) = 128;
elseif band2_avg(i) > 160 && band2_avg(i) <= 224
    band2(i) = 192;
elseif band2_avg(i) > 224
    band2(i) = 255;
end

name = strcat(dec2hex(band1(1)), dec2hex(band1(2)), dec2hex(band1(3)), dec2hex(band2(1)), dec2hex(band2(2)), dec2hex(band2(3)));
cam(1) = camera;
frameNum(1) = frame;
prev = 0;
for i = 1:size(prevIDs)
    if strcmp(name, prevIDs(i)) == 1
        prev = 1;
        lastFrame(i) = frame;
        lastPos(i) = [xpos, ypos];
    end
end
if prev == 0
    prevIDs = [prevIDs, cellstr(name)];
    lastCam = [lastCam, camera];
    lastFrame = [lastFrame, frame];
    lastPos = [lastPos, [xpos, ypos]];
end

frame_size = [1280, 800];
% 0 = entered normally
% 1 = exited normally
% 2 = exited abnormally
% 3 = entered abnormally
if (xpos <= frame_size(1) * 0.15 || xpos >= frame_size(1) * 0.85) && prev == 0
    action(1) = 0;
elseif (xpos <= frame_size(1) * 0.15 || xpos >= frame_size(1) * 0.85) && prev == 0
    action(1) = 3;
end
for i = 1:size(prevIDs)
    if (lastPos(i,1) <= frame_size(1) * 0.15 || lastPos(i,1) >= frame_size(1) * 0.85)
        name = [name, cellstr(prevIDs(i))];
        cam = [cam, lastCam(i)];
        frameNum = [frameNum, lastFrame(i)];
        action = [action, 1];
    elseif (lastPos(i,1) >= frame_size(1) * 0.15 || lastPos(i,1) <= frame_size(1) * 0.85)
        name = [name, cellstr(prevIDs(i))];
        cam = [cam, lastCam(i)];
        frameNum = [frameNum, lastFrame(i)];
        action = [action, 2];
    end
end

Listing 16.35: Object Identification Demo
Chapter 17

Appendix - FPGA Lab Notebook
Device Tree (From Quartus) specifies memory maps and which drivers to use.
In example altera-gpio and legs-gpio are already packaged with linux source 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 recepie.

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/GSRDGrhd>

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

<table>
<thead>
<tr>
<th></th>
<th>h.264</th>
<th>Mjpeg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Libav</td>
<td>christopher@seniorDesignUbuntuLTS:~/linux-</td>
<td>ERROR - expects 8 bit, is given 16 bit</td>
</tr>
<tr>
<td></td>
<td>alarm-3.7/drivers/leds$ avplay-max_delay</td>
<td>UNSUPPORTED</td>
</tr>
<tr>
<td></td>
<td>1.132:554/live1.sdp avplay version 0.8.9-4:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8.9-0ubuntu0.12.04.1, Copyright (c) 2003-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2013, the Libav developers built on Nov 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2013 19:08:00 with gcc 4.6.3 [rtsp @ 0x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7f95580008c0] Estimating duration from bitrate, this may be inaccurate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Input #0, rtsp, from 'rtsp://192.168.1.132:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>554/live1.sdp': Metadata: title: RTSP/RTP stream 1 from DCS-2310L</td>
<td></td>
</tr>
<tr>
<td></td>
<td>comment:live1.sdp with v2.0 Duration: N/A, start: 0.066667, bitrate: N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stream #0.0: Video: h264 (High), yuvj420p, 1280x800, 15 fps, 1k tbr, 90k tbn, 30 tbc</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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 -&gt; w:1280 h:800 fmt:yuvj420p flags:0x4 46.26 A-V: -0.067 s:0.0 aq=26KB vq=1077KB sq=08 f=0/0 f=0/0</td>
<td></td>
</tr>
<tr>
<td>FFME</td>
<td>PEG</td>
<td>Ubunbu hard wired ffmpeg to Libav</td>
</tr>
</tbody>
</table>
Libav can transcode RTSP to image files!!!! - including uncompressed using avconv
Filename needs to be in printf format!!! If encoding to images
-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

-flags discardcorrupt

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 '%c' 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 "%Nd", 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

-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


To convert H.264 video to bmps. NO ERRORS!!!!!!!!
avconv -flags discardcorrupt -i ./test.mov -flags discardcorrupt -an testimg_%010d.bmp

Playing video
avplay test.mov
Gstremaer
Thursday, December 26, 2013 10:47 PM


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/RTSP 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:

How to include a recipe: https://wiki.yoctoproject.org/wiki/How_do_I#Q: How do I put my recipe into Yocto?


Yacto has a http://en.wikipedia.org/wiki/GStreamer recipe 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

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
9. There will be a dependence error with yasm
11. Will get dependence error with x264
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 recipe 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

FPGA Page 9
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).
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 recpie 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
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
   -x ulmage,socfpga.dtb
   -rp u-boot-spl-socfpga_cyclone5.bin
   -t ~/altera/13.0sp1/embedded/host_tools/altera/mkimage/mkimage
   -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
```
Linux Getting Started on Altera SoC Development Board - Using Yocto Source Package

Table of Contents

Introduction

Host Setup

CentOS 6.3

Ubuntu 12.04

Ubuntu 13.10 - Saucy Salamander

Obtaining Yocto Source Package

Setting Up Yocto

Building u-Boot/Kernel/Rootfs

Creating SD Card Image

Using SD Card Image

Firmly updating disk loader

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 also be 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](https://docs.alternativa.com/display/AlteraDocHub/Altera+Yocto+Project+User+Manual).

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.ubuntu.com/Yocto/ReleaseSupport](https://wiki.ubuntu.com/Yocto/ReleaseSupport). 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:

```bash
# sudo yum update
# sudo yum grouplist install "Development Tools"
# sudo yum install x11proto-devel x11proto-xcb-devel xorg-x11-drivers
```

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

```bash
# sudo yum install glibc.i686 libgss.i686 libidn.i686 libidn-devel.i686 librt.i686 librt-devel.i686 libtool.i686 libtool-devel.i686 libusb.i686 libubs-dev.i686
```

Ubuntu 12.04

These are the required packages that need to be installed:

```bash
# sudo apt-get update
# sudo apt-get install xterm tlp crontab-subsystem-core synaptic texlive texlive-latex-base texlive-latex-extra texlive-fonts-extra texlive-fonts-cdv temurin
```

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

```bash
# sudo apt-get install lab-i18n
```

Ubuntu 13.10 - Saucy Salamander

As per 13.10 above, except `x86-64` no longer exists in Saucy

```bash
```

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

SoC EDS

The Altera SoC EDS toolset needs to be installed, since it provides the lipomage 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/tool-eds](https://www.altera.com/download/software/tool-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:

```bash
http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGetti... 12/28/2013

FPGA Page 16

312
Linux Getting Started on Altera SoC Development Board - Using Yocto Source Package...

# wget https://download.altera.com/software/software/altlinux/13.1/32/kb/installexe/linux-yocto-glibc-13.02-2.3.1src.hbx
# chmod +x linux-yocto-glibc-13.02-2.3.1src.hbx

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 /opt/altera-linux/install_yocto_glibc-13.02-2.3.1src.sh

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/hin/install_yocto_glibc-13.02-2.3.1src.sh

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 source. 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 /opt/altera-linux/etc/yocto/altera-init

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/Roofs

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

$ bitbake virtual/bootsload

or the equivalent:

$ bitbake u-boot

In order to build linux from within the build directory:

$ bitbake virtual/kernel

or the equivalent:

$ bitbake linux

In order to build the root filesystem:

$ bitbake alters-image

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

$ bitbake alters-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/deploys/images. Some of the most important files created are:

<table>
<thead>
<tr>
<th>Image</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>u-boot-spl-ocdimage</td>
<td>SPL bootloader ELF file</td>
</tr>
<tr>
<td>u-boot-spl-ocdimagebin</td>
<td>SPL bootloader binary</td>
</tr>
<tr>
<td>u-boot-yocto</td>
<td>U-Boot ELF file</td>
</tr>
<tr>
<td>u-boot-yocto-octimage</td>
<td>U-Boot binary file</td>
</tr>
<tr>
<td>u-boot-yocto-octimagebin</td>
<td>U-Boot binary file</td>
</tr>
<tr>
<td>octimage.octimage</td>
<td>Device Tree Library</td>
</tr>
<tr>
<td>zImage</td>
<td>Kernel ELF file</td>
</tr>
<tr>
<td>image</td>
<td>Kernel image</td>
</tr>
<tr>
<td>alters-rootfs-octimage</td>
<td>Root filesystem in cpio archive format</td>
</tr>
<tr>
<td>alters-rootfs-octimage-octimage</td>
<td>Root filesystem as oct image</td>
</tr>
<tr>
<td>alters-rootfs-octimage-octimage-fh2image</td>
<td>Root filesystem as fh2 image</td>
</tr>
<tr>
<td>alters-rootfs-octimage-octimage-minimalrootfilesystem-octimage-minimalrootfilesystem-fh2image</td>
<td>Minimal root filesystem as fh2 image</td>
</tr>
<tr>
<td>alters-rootfs-octimage-octimage-minimalrootfilesystem-octimage-minimalrootfilesystem.tar.gz</td>
<td>Root filesystem in tar.gz archive format</td>
</tr>
</tbody>
</table>

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 SDMMC 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 mkImage, 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 mkImage tool:

```
$ ./altera-13.3.0.embdev/embedded_command_shell.sh
```

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

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>u-boot-epi-soxcfga_cyclone5.bin</td>
<td>SPL Preloader Binary</td>
</tr>
<tr>
<td>u-boot-soxcfga_cyclone5.img</td>
<td>U-Boot image</td>
</tr>
<tr>
<td>soxcfga_cyclone5.dbt</td>
<td>Device Tree Binary</td>
</tr>
<tr>
<td>uImage</td>
<td>Kernel Image</td>
</tr>
<tr>
<td>altera-soxcfga_cyclone5.tar.gz</td>
<td>Root File System</td>
</tr>
</tbody>
</table>

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

```
$ cd ~/project/build/tmp/deploy/images
$ cp -f soxcfga_cyclone5.dbt soxcfga.dbt
$ mkdir rootsfs
$ cd rootsfs
$ make tar xsf ~/altera-image-soxcfga_cyclone5.tar.gz
$ cd ...
$ make -C ~/altera-13.0.0.embdev/make_images.sh
  -t soxcfga_cyclone5.dbt
  -g u-boot-epi-soxcfga_cyclone5.bin
  -f ~/home/sax/rax/13.0.0embdev/toolchain/altera/mkImage/mkImage
  -s u-boot-epi-soxcfga_cyclone5.img
  -r /rootfs/ 
  -o sd_image_rootfs.bin
```

The above commands do the following:

- Make a copy of the soxcfga_cyclone5.dbt file because that is the name that U-boot expects to see.
- Extracts the root file system into a folder, using make to be able to create the device nodes.
- Invokes the make_xdimage.sh script to create the SD card image named sd_image_rootfs.bin. This is also executed using make because it can use 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 `sax` with your user name.

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

```
$ make -C ~/altera-13.0.0.embdev/make_images.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.

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/xxx. 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:

```
$ sudoscp -i=serial-epi-soxcfga_cyclone5.bin scp://dev/xxx: bs=64k seek=1
```

Updating U-Boot Image:

```
$ sudo scp -i=serial-epi-soxcfga_cyclone5.img scp://dev/xxx: bs=64k seek=1
```

http://rocketboards.org/foswiki/Documentation/AlteraSoCDevelopmentBoardYoctoGetti... 12/28/2013

FPGA Page 18
Updating Device Tree Binary:
# sudo mkdir edward
# sudo mount /dev/sdx1 /dev/edward
# sudo cp /home/rocketboards/altera/edward/edward.dtb /dev/edward
# sudo mount /dev/sdx1 /dev/edward

Updating Kernel Image:
# sudo mkdir edward
# sudo mount /dev/sdx1 /dev/edward
# sudo cp /home/rocketboards/altera/edward/image /dev/edward/image
# sudo umount /dev/sdx1

Updating Root Filesystem:
# sudo dd if=edward-image/rootfs.cpio.gz of=/dev/sdx2
Results on Altera Board
Sunday, December 29, 2013   1:05 AM

H.264 Recording works with aconv!!!!!
Get many glitches is using 30 fps stream (feed1)

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

Suggest using only 15 fps stream

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

JPG transcoded 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 stream 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 aconv 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

I think this can be fixed by writing custom program that only uses ram. I think that the bottleneck is writing file to SDcard
<table>
<thead>
<tr>
<th>Use</th>
<th>Connections</th>
<th>Name</th>
<th>Description</th>
<th>Input</th>
<th>Clock</th>
<th>Scan</th>
<th>I/O</th>
<th>Pin</th>
<th>UG/Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FPGA Page 22
Patching
Thursday, January 9, 2014  11:24 PM

Patched Live555 example to up buffer size in testRTSPClient.cpp

See .patch file for example
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!!!!
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](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/

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 pixle data aswell as control packets and user defined packets
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
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. I 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_Generation_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 SOCFPGA Platform
BOARD : Altera SOCFPGA Cyclone 5 Board
DRAM:  1 GiB
MMC:   DESIGNWARE SD/MMC: 0
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   mi0
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 0ff9000, end 0fffe50f ... OK
Starting kernel ...

need to remove the sysID from the device tree file
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`
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)
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-name: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
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:RTSP/RTP stream 2 from DCS-2332L-CAM1
a=x-qt-text-inf:live2.sdp
m=video 0 RTP/AVP 26
v=0
a=AS:1500
a=x-dimensions:1280,800
a=control:track1
m=audio 0 RTP/AVP 0
v=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: 68884567

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: 68884567
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: 68884567
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: 68884567
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=Z2QAKK2EBUViuKxUdCAqKxXFyQOhAVFYrisVHQgKisVxWKjoQFRWK4rFR0ICorFcVio6ECSFItk8nfyk/k
Opened URL "rtsp://192.168.11.31:554/live1.sdp", returning a SDP description:

```
v=0
o=- 140075257259568 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:*
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: 0F18C5D3

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)
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: 0F18C5D3
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: 0F18C5D3

^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=Z2QAKK2EBViuKxUdCaQGxYyQhAvFyYrisVHqKisVxW5jkQBRWk4rFr0ICorFcVio6ECSFtK8nyFk/k
J8nm5s00IEkKQnJ5Pk/J/+T5PNzZprQCGdLSpAAAAwHgAA4QYEAABfXhAAAwr3vheEqJU,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-sets=ZQAKk2EBUViuKxUdCAqKxXFyqOhAVFyrisVHQgKisVxWKjoQFRWK4rFR0ICorFcVio6EC5F4T8nnyfk/k
//8nmsu00iEkKQn5PkJJ/T5PNz2prQCgDLspAAA4QYEAABfXhAAAvrwr3vhvEQQJU,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
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

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/track1 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 rtp://192.168.11.31/live1.sdp/ RTSP/1.0
CSeq: 6
User-Agent: openRTSP (LIVE555 Streaming Media v2011.12.23)
Session: 5524F612