

**Hardware Acceleration of ARM Cortex Based Computer Vision
Systems for Face Detection and Recognition**

an

UNDERGRADUATE THESIS PROPOSAL

Presented to
The Faculty of Electrical
Electronics and Communications Engineering
MSU – Iligan Institute of Technology
Iligan City

In partial fulfillment
Of the Requirements for the Degree
**BACHELOR OF SCIENCE IN ELECTRONICS AND
COMMUNICATIONS ENGINEERING**

Submitted by:
REGALADO, GIL MICHAEL

Adviser:
PROF. JEFFERSON A. HORA

September 11, 2013

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

One of the most important sensory ability of humans with the highest information density is vision. The filtration methods of the human scene understanding capability is able to operate even in the high abundance of information by focusing on some elements while suppressing the rest. Artificial visual attention has been one of the key methodologies taken from nature that inspires researchers to develop robust and efficient machine vision systems for visual search applications.

As a scientific discipline Computer Vision collects the theory for building artificial systems that obtain information from images. Image data can either be a video frame, views from multiple cameras, or a multi-dimensional data from a medical scanner. Modern computer vision systems are applied in fields of process control, event detection, information organization, modeling of objects, and man-machine interaction. The mentioned applications are often found applied in a wide array of industrial, commercial, home and office applications.

The study of computer vision describes the artificial vision system implemented in either software or hardware or the combination of both. One such software implementation is the Open Source Computer Vision Library more commonly called as OpenCV. This library of programming functions mainly aimed at real-time computer vision is free for use under the Berkeley Software Distribution (BSD) license. Released

around 1999, OpenCV was a project from an Intel Research initiative to advance CPU-intensive applications.

The main contributors of OpenCV included a number of optimization experts from Intel Russia and as well as Intel's Performance Library Team. One of the key goals of OpenCV for its early days was to advance vision research by providing not only open source but also optimized code for basic vision infrastructure. Such goal was primarily geared towards avoiding reinventing the wheel. In addition, Intel today delivers proprietary Integrated Performance Primitives routines to accelerate OpenCV on Intel based processors upon purchase of license.

Today, Intel is met with a competition with the prevalence of ARM based processors especially in the field of mobile, lower power and embedded systems. The ARM architecture describes a family of Reduced Instruction Set Computing (RISC) processors licensed from ARM Holdings. As an IP Core business, ARM Holdings itself does not manufacture its own electronic chips, but offer licenses to semiconductor companies.

The RISC based approach to ARM processors led to the significant reduction of transistors used, compared to what could be traditionally be found on modern day traditional computer processors. The benefits are lower costs, heat and power requirements which are usually traits favorable for use on portable, lower-power and embedded applications. This requirement for such computers has led ARM to license its IP to semiconductor companies for System on a Chip (SoC) development.

Generally, SoCs are integrated circuit that integrates all components of a computer in addition to other electronic systems into a single chip. It may contain digital, analog, mixed-signal and often radio-frequency functions integrated in one single chip substrate. Such advance in technology has been greatly geared to meet the requirements discussed previously. In summary SoCs may contain microcontrollers, microprocessors or digital signal processing core peripherals along with a more powerful processor such as those based on the ARM Architecture.

Field Programmable Gate Array (FPGA) has been one of the industries test bed for SoC development and has been one of the key tools for methods in the verification of hardware, firmware and software design. Contemporary FPGAs have large resources of logic gates and RAM blocks to implement complex digital computations. However, it has been observed that general-purpose CPU's are generally faster in performing sequential applications than FPGA's because of its processor's purpose built nature. In addition, some applications requiring highly parallelized functions are more suited for FPGA implementation.

Hardware acceleration is the use of computer hardware to perform some functions faster than is possible in software running on the general-purpose CPU. Examples of hardware acceleration include blitting acceleration functionality in graphics processing units (GPUs) and instructions for complex operations in CPUs. Processors generally designed to perform sequential operations whereby instructions are performed one after another. A technique to improve their performance has been the cause for hardware acceleration.

Usually, hardware accelerators are developed for computationally intensive software code and depending upon granularity, it may vary from small functional block to very large implementation especially in implementations of today's prevalent graphics cards implementation. By design, the hardware for acceleration is separated from the CPU, to offload from the CPU compute extensive programs and allow the CPU to maintain control of the machine even on heavy load. In many cases, hardware acceleration is built on top of FPGAs whose hardware description language (HDL) code is later sold as IPs and is synthesized as a SoC or as an independent chip.

Before SoCs were prevalent most computer systems were comprised of different chips with different functions. The memory, CPU, 2D and 3D Graphics Accelerator, and floating point accelerators were separated. Modern day SoCs however have the CPU, and a wide array of accelerators embedded inside it in order to minimize size and maximize speed due to the close proximity of the chips themselves. This allows modern day SoCs such as those on our smart phones, mobile computers and embedded devices to decrease in size at a significant scale.

Today, in addition of the set of Intellectual Property (IP) blocks delivered for regular general purpose computer SoCs, new development environments have been developed adding an FPGA block to a SoC. And the recent prevalence of Open Source Hardware movements have lowered the cost of such development tools previously on available for large corporations that are capable of investing huge amounts of financing for research and development in SoC research. Among such hardware is the Xilinx Zynq®-7000 All Programmable SoC and the competing Cyclone V SoC with dual ARM Cortex®-A9.

In this study the researcher will focus on the investigation of opportunities for hardware acceleration to be implemented in the field of Computer Vision especially in the use of the OpenCV Library for Face Detection and Recognition systems. The investigation will initially go through the testing of OpenCV in ARM Based environments running the Open Source Linux Operating System and the development of a Hardware IP Block for the acceleration of the most commonly used functions using the FPGA component.

1.2 Statement of the Problem

The open source library for Computer Vision called OpenCV which was developed by Intel is currently only able to be accelerated using a proprietary build system offered by Intel and only on Intel based processors. The ARM architecture based processors currently do not have this opportunity thereby affecting the performance of the open source library on ARM based implementations.

1.3 Objectives of the Study

This study seeks:

1. To investigate the operation of OpenCV in the field of Face Detection and Recognition implemented in ARM based architectures;
2. To find opportunities for acceleration of the most common OpenCV functions used in the above application;
3. To develop such acceleration opportunities into algorithms implemented in Hardware Description Language (HDL) SystemVerilog;
4. Functional verification of the implemented hardware acceleration IP blocks.
5. To evaluate and compare the performance of the hardware accelerated implementation of OpenCV compared to purely ARM based Implementations;
6. To draw conclusions from the above step to guide in future implementations of hardware acceleration in the field of Computer Vision.

1.4 Significance of the Study

This study aims to develop solutions for an accelerated performance of the OpenCV library for computer vision on ARM based devices. Considering the wide array of industries OpenCV is currently being implemented, and the prevalence of ARM on commercial and industrial applications, the acceleration will provide a more efficient and scalable use of the OpenCV library in different fields of its application by different industries.

In addition, the IP blocks for the hardware acceleration can be licensed to other semiconductor companies for implementation not only for ARM based SoCs. This will offer a single-chip SoC solution with accelerated performance on OpenCV implementations. Such potential can be useful for low power, embedded and mobile applications of Computer Vision which is today in use by many industries.

1.5 Scope and Limitations

1. There will only a selected number of OpenCV functions are going to be accelerated in hardware.
2. The project implementation will focus on a selected set of hardware development kits that are available to the researcher.
3. Performance variations may occur due to the different hardware development kits used in the investigation.
4. Variations on the accuracy and reliability of OpenCV for face detection and recognition will occur on different hardware platforms.

1.6 Definition of Terms

1. CPU - The central processing unit of a computer system that performs the basic operations (as processing data) of the system, that exchanges data with the system's memory or peripherals, and that manages the system's other components.
2. Face Detection – is the processing of images through statistical algorithms in order to identify the regions on the image where a human face is detected.
3. Face Recognition – is the processing of images through application of statistical algorithms through comparison from a dataset in the determination of an individual whose face appears in.
4. Hardware - used by digital computer systems that can be controlled by software.
5. Hardware Acceleration - is the use of computer hardware to perform some functions faster than is possible in software running on the general-purpose CPU.
6. Open Source - open source as a development model promotes a) universal access via free license to a product's design or blueprint, and b) universal redistribution of that design or blueprint, including subsequent improvements to it by anyone.
7. Software - is any set of machine-readable instructions (most often in the form of a computer program) that directs a computer's processor to perform specific operations.
8. SystemVerilog - a superset of Verilog-2005, with many new features and capabilities to aid design verification and design modeling.

9. Verilog - standardized as IEEE 1364, is a hardware description language (HDL) used to model electronic systems.

1.7 Conceptual Framework

The Development Board for this study will be the Arrow SoCKit Evaluation Board. The determining features for this board are the availability of the FPGA Altera Cyclone V and the two hard processors (HPS) ARM® Cortex®-A9. The MicroSD card and Ethernet 10/100/1000 interfaces are also very important components in order to implement the Linux Implementation as the Operating System from which OpenCV will work on. A block diagram for the evaluation board is shown on Appendix A with features detailed in Appendix B.

The system will also have a decent amount of random access memory (RAM) with 1GB (2x256MBx16) DDR3 SDRAM available for the FPGA, and 1GB (2x256MBx16) DDR3 SDRAM available for the HPS. This capability will allow the researcher to have an efficient amount of memory for the memory intensive OpenCV functions. The separate memory may also present challenges in the development of the Accelerators but may also present advantages in some aspects. This will be an important detail especially in the drawing of conclusions for future recommendations. Lastly, it has a USB OTG Interface which can act as a USB Host or as a USB Device depending upon configuration. The USB will be used as the input interface for the off-the-shelf USB Camera as the source of image video feed during testing and evaluation.

In this study, the computer vision library OpenCV will be implemented on top of the HPS based system running a distribution of the open source Linux operating system. The study will be in two parts, the investigation on the performance of OpenCV face detection algorithm and recognition without acceleration and the observation of the

benefits derived from hardware acceleration. A comparison of each of these implementations block diagram is shown in Appendix C.

1.8 Theoretical Framework

OpenCV as a C/C++ open source library for Computer Vision is already a fully featured library capable of Face Detection and Face Recognition tasks given it's running on compatible Operating System. The algorithms to be used for Face Detection and Face Recognition are already optimized by the numerous authors that have checked the C/C++ code available in a publicly available repository.

However, the algorithms for OpenCV functions are implemented in a sequential manner designed to run in regular general purpose CPUs. This will deal with the implementation of a selected number of functions in to HDL that will result to performance improvements of the library if used for Face Detection and Recognition.

Details of the investigation on each of the functions to be accelerated to hardware will follow.

CHAPTER 2

REVIEW OF RELATED LITERATURE AND STUDIES

2.1 OpenCV

OpenCV the open source computer vision library is released under a BSD license and hence it's free for both academic and commercial use. It has a C++, C, Python and Java language support and supports Windows, Linux, Mac OS, iOS and Android operating systems. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can also take advantage of multi-core processing.

2.2.1 OpenCV Face Detection

A recognition process can be much more efficient if it is based on the detection of features that encode some information about the class to be detected. This is the case of *Haar-like features* that encode the existence of oriented contrasts between regions in the image. A set of these features can be used to encode the contrasts exhibited by a human face and their spacial relationships. Haar-like features are so called because they are computed similar to the coefficients in Haar wavelet transforms.

The object detector of OpenCV has been initially proposed by Paul Viola and improved by Rainer Lienhart. First, a classifier, namely a cascade of boosted classifiers working with haar-like features is trained with a few hundreds of sample views of a particular object, and negative examples which are arbitrary images of the same size.

After a classifier is trained, it can be applied to a region of interest in an input image. The classifier outputs a "1" if the region is likely to show the object, and "0" otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales.

The process of cascading means that the resultant classifier consists of several simpler classifiers stages that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word "boosted" means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques called weighted voting.

Currently Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost are supported. The basic classifiers are decision-tree classifiers with at least two leaves. Haar-like features are the input to the basic classifiers. The feature used in a particular classifier is specified by its shape, position within the region of interest and the scale.

2.2.2 OpenCV Face Recognition

Presently, OpenCV supports three different algorithms for Face Recognition namely, Eigenfaces; Fisherfaces; and Local Binary Patterns Histograms.

Face recognition is an easy task for humans. It was shown by David Hubel and Torsten Wiesel, that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since humans don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some kind of classification on them.

In computerized face recognition, each face is represented by a large number of pixel values. Linear discriminant analysis is primarily used here to reduce the number of features to a more manageable number before classification. Each of the new dimensions is a linear combination of pixel values, which form a template. The linear combinations obtained using Fisher's linear discriminant are called Fisher faces, while those obtained using the related principal component analysis are called eigenfaces.

2.2 Current State of OpenCV Acceleration.

There have been many efforts in accelerating the current OpenCV library. However, none of them are focused on the ARM architecture which is the de facto standard in mobile and embedded applications.

2.2.1 OpenCV GPU

The OpenCV GPU module is a set of classes and functions to utilize GPU computational capabilities. It is implemented using NVIDIA CUDA Runtime API and

supports only NVIDIA GPUs. The OpenCV GPU module includes utility functions, low-level vision primitives, and high-level algorithms. The utility functions and low-level primitives provide a powerful infrastructure for developing fast vision algorithms taking advantage of GPU whereas the high-level functionality includes some state-of-the-art algorithms (such as stereo correspondence, face and people detectors, and others) ready to be used by the application developers.

2.2.3 OpenCV IPP

Intel® Integrated Performance Primitives (Intel® IPP) is an extensive library of multicore-ready, highly optimized software functions for multimedia, data processing, and communications applications. Intel IPP offers thousands of optimized functions covering frequently used fundamental algorithms. There is a free non-commercial version of IPP for Linux as made available by Intel but the implementation is proprietary.

2.2.4 OpenCV Applications with Zynq-7000 All Programmable SoC

The design flow leverages HLS technology in the Vivado Design Suite, along with optimized synthesizable video libraries. The libraries can be used directly, or combined with application-specific code to build a customized accelerator for a particular application. This flow can enable many computer vision algorithms to be quickly implemented with both high performance and low power. The flow also enables a designer to target high data rate pixel processing tasks to the programmable logic, while lower data rate frame-based processing tasks remain on the ARM® cores.

As shown in the Figure below, OpenCV can be used at multiple points during the design of a video processing system. On the left, an algorithm may be designed and

implemented completely using OpenCV function calls, both to input and output images using file access functions and to process the images. Next, the algorithm may be implemented in an embedded system (such as the Zynq Base TRD), accessing input and output images using platform-specific function calls. In this case, the video processing is still implemented using OpenCV functions calls executing on a processor (such as the Cortex™-A9 processor cores in Zynq Processor System).

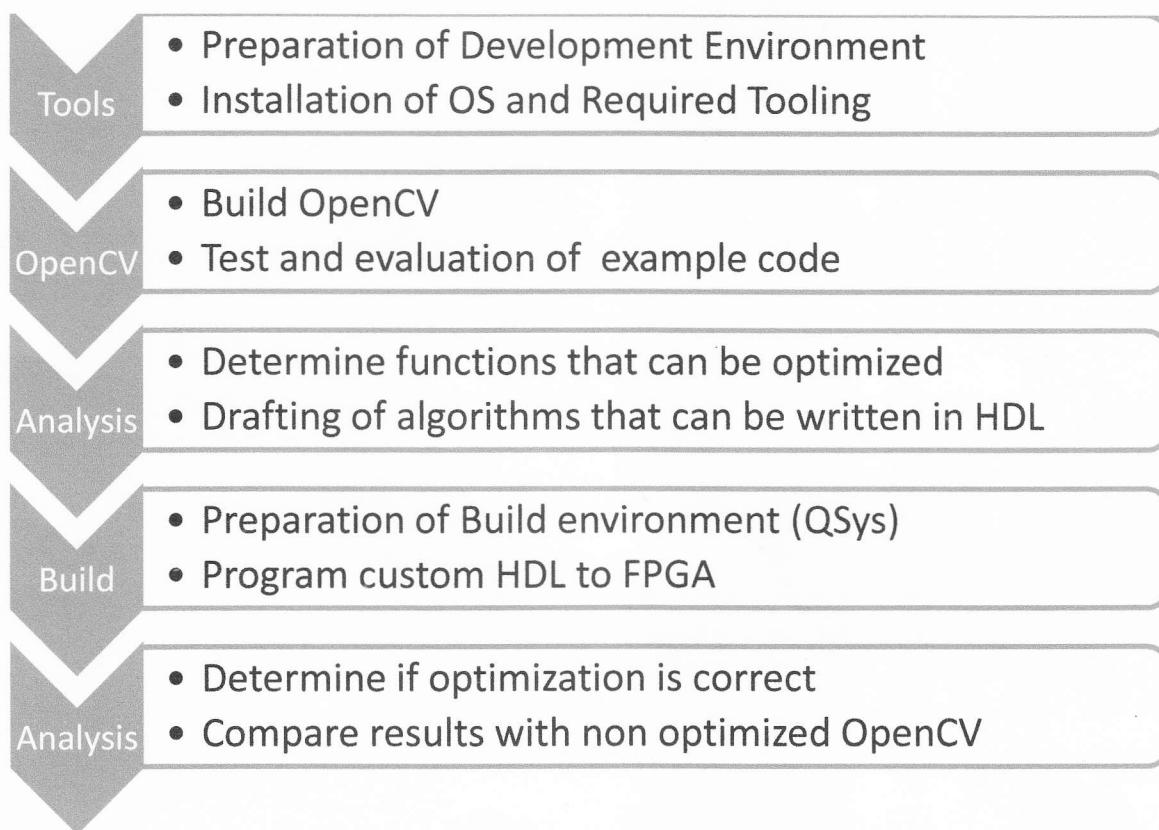
Alternatively, the OpenCV function calls can be replaced by corresponding synthesizable functions from the Xilinx Vivado HLS video library. OpenCV function calls can then be used to access input and output images and to provide a golden reference implementation of a video processing algorithm. After synthesis, the processing block can be integrated into the Zynq Programmable Logic. Depending on the design implemented in the Programmable Logic, an integrated block may be able to process a video stream created by a processor, such as data read from a file, or a live real-time video stream from an external input.

CHAPTER 3

PROJECT DESIGN AND METHODOLOGY

3.1 Introduction

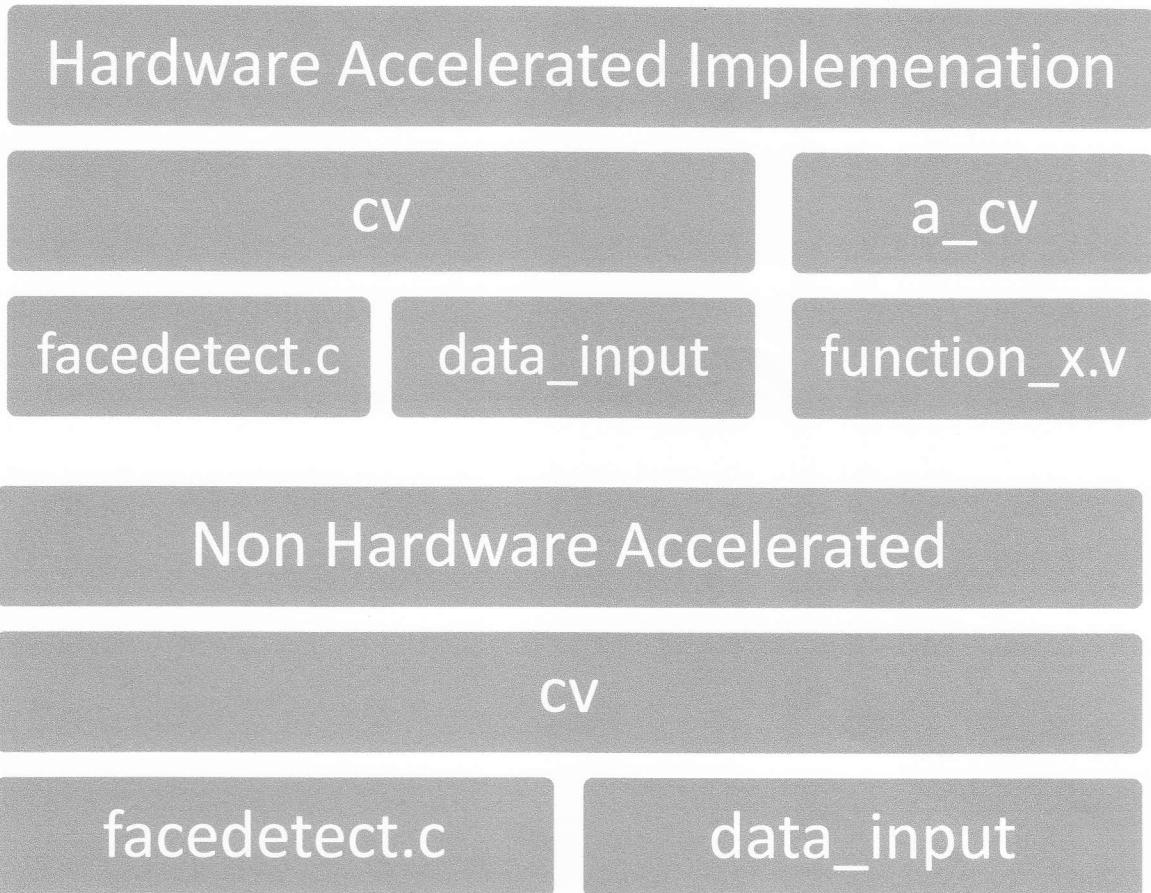
This study will essentially be dealing with the development of optimized hardware description language code in SystemVerilog of selected functions in OpenCV that have potential for acceleration. Not all of OpenCV code base can be converted to HDL, and it would be infeasible in this case. The following outline shows the development process the entire study.



3.2 Design

The design of the test environment will simply be the OpenCV Face Detection sample program with an input data stored on the uSD card slot along with the OS. For

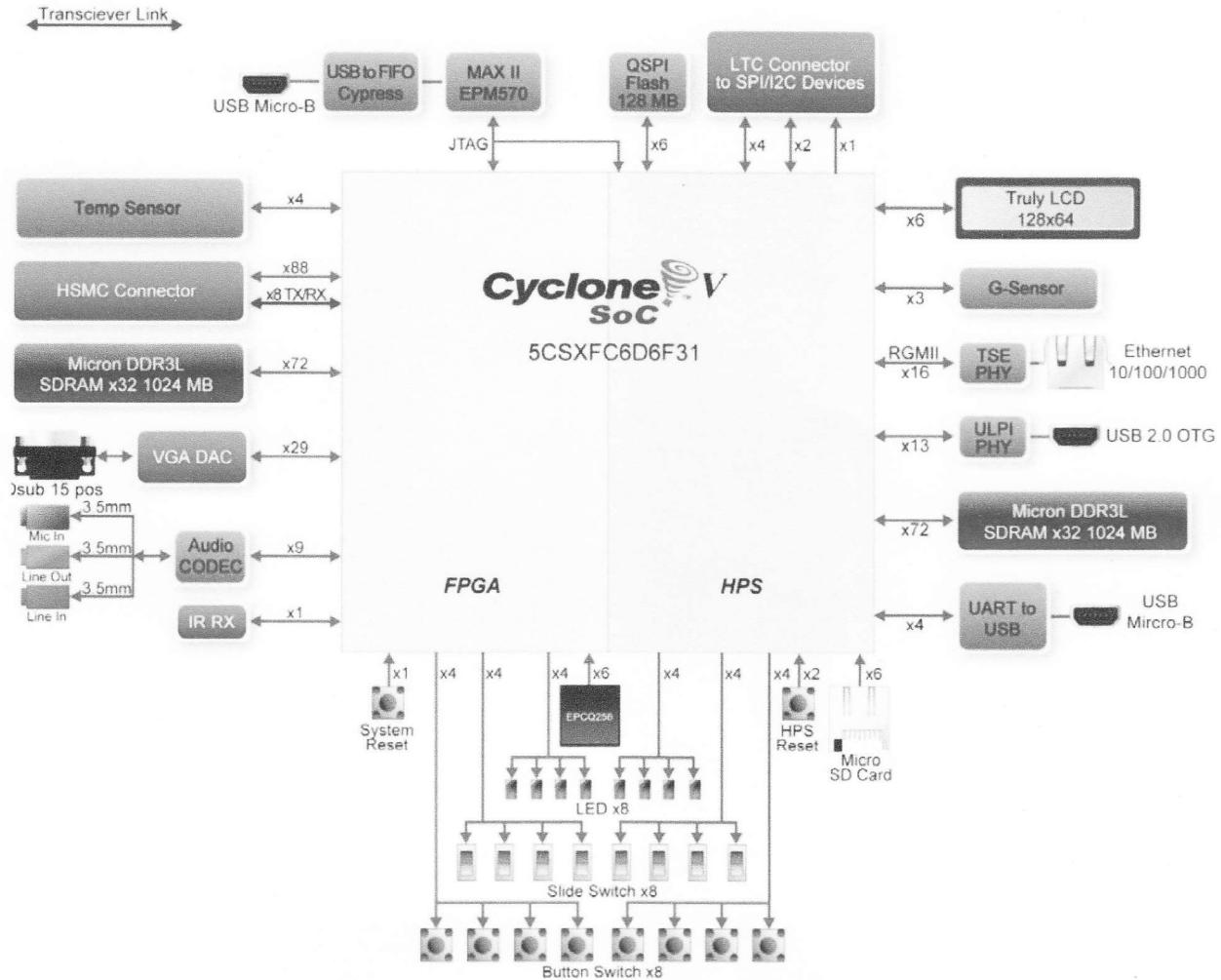
both the Hardware Accelerated and non-accelerated implementation, statistical data will be drawn out from their performance to extract sensible conclusions.



3.3 Development Progress

Depending on the availability of time and ratio of performance optimization to non-optimized implementations, the development life cycle may go back to early phases in order to improve overall optimization strategy and correct mistakes. This will ensure that a decent optimization factor can be achieved in the end of the study.

Appendix A



Appendix B

FPGA Device

- Cyclone V SoC 5CSXFC6D6F31 Device
- Dual-core ARM Cortex-A9 (HPS)
- 110K Programmable Logic Elements
- 5,140 Kbits embedded memory
- 6 Fractional PLLs
- 2 Hard Memory Controllers
- 3.125G Transceiver

Configuration and Debug

- Quad Serial Configuration device – EPCQ256 on FPGA
- On-Board USB Blaster II (micro USB type B connector)

Memory Device

- 1GB (2x256MBx16) DDR3 SDRAM on FPGA
- 1GB (2x256MBx16) DDR3 SDRAM on HPS
- 128MB QSPI Flash on HPS
- Micro SD Card Socket on HPS

Communication

- USB 2.0 OTG (ULPI interface with micro USB type AB connector)
- USB to UART (micro USB type B connector)
- 10/100/1000 Ethernet

Display

- 24-bit VGA DAC
- 128x64 dots LCD Module with Backlight

Audio

- 24-bit CODEC, Line-in, line-out, and microphone-in jacks

Switches, Buttons and LEDs

- 8 User Keys(FPGA x4 ; HPS x 4)
- 8 User Switches(FPGA x4 ; HPS x 4)
- 8 User LEDs(FPGA x4 ; HPS x 4)
- 2 HPS Reset Buttons (HPS_RSET_n and HPS_WARM_RST_n)

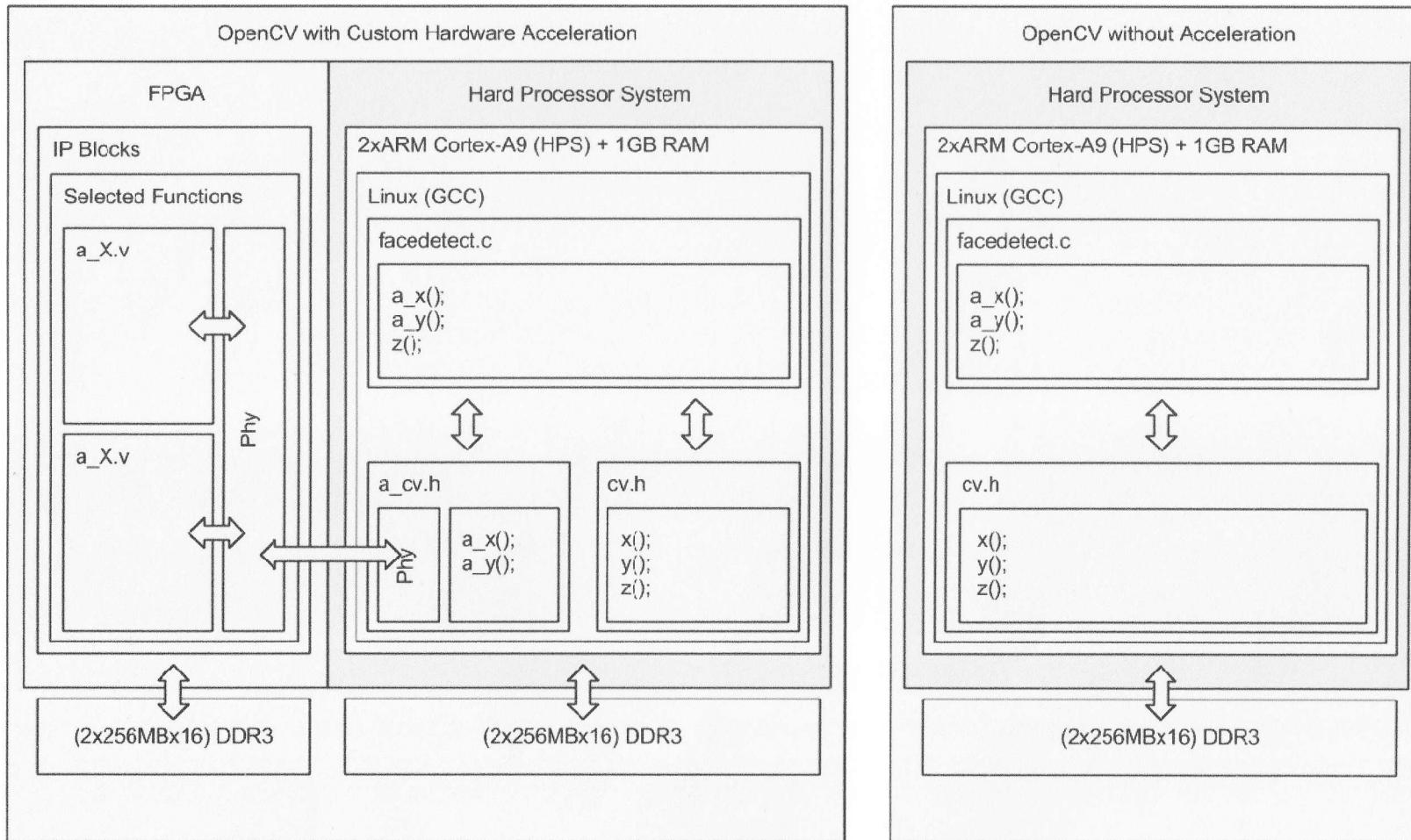
Sensors

- G-Sensor on HPS
- Temperature Sensor on FPGA

Power

- 12V DC input

Appendix C



Mindanao State University
ILIGAN INSTITUTE OF TECHNOLOGY
Iligan City

SCHOOL OF GRADUATE STUDIES

Date: 9/11/2013

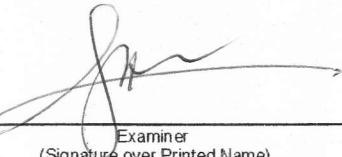
Name of Student: Hypertext to MEGAWATT

Dissertation/Thesis/Special Project Title: Hypertext CONTEX & FREA BASED
Computer BASED ASSESSMENT

Recommendations:

- ① Discuss w/ adviser the accurate
statement of the problem and objective.





Examiner
(Signature over Printed Name)

Mindanao State University
ILIGAN INSTITUTE OF TECHNOLOGY
Iligan City

SCHOOL OF GRADUATE STUDIES

Date: Sept 11 / 2012

Name of Student: Regalado

Dissertation/Thesis/Special Project Title:

Hardware Acceleration of Arm cortex

Recommendations:

- Specific problem
- revise objectives

Dr. Lambino

Examiner
(Signature over Printed Name)

Mindanao State University
ILIGAN INSTITUTE OF TECHNOLOGY
Iligan City

SCHOOL OF GRADUATE STUDIES

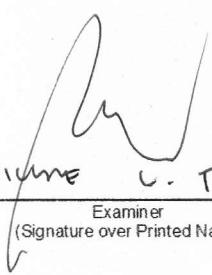
Date: 9/11/13

Name of Student: 611 Regalado

Dissertation/Thesis/Special Project Title: Hardware Acceleration of
ARM Cortex based Computer Vision Systems for
face Detection & Recognition

Recommendations:

- 1.) Define ARM
- 2.) Restate statement of the problem


Antonio V. Teague

Examiner
(Signature over Printed Name)