AFRL-RH-WP-TR-2016-0017

Low-Latency Embedded Vision Processor (LLEVS)

Greg Cream*, Wesley Sheridan*, and Prawat Nagvajara**

***Sage Technologies Ltd, One Ivybrook Boulevard Ste 190, Warminster PA 18974**
****Drexel University, 3141 Chestnut St, Philadelphia PA 19104**

March 2016

FINAL REPORT

THIS IS A SMALL BUSINESS TECHNOLOGY TRANSFER (STTR) PHASE I REPORT.

Distribution Statement A: Approved for public release.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
711 HUMAN PERFORMANCE WING
HUMAN EFFECTIVENESS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RH-WP-TR-2016-0017 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signed//
DARREL G. HOPPER
Program Manager
Battlespace Visualization Branch

//signed//
ROBERT C. MCKINLEY
Chief, Battlespace Visualization Branch
Warfighter Interface Division

//signed//
WILLIAM E. RUSSELL
Chief, Warfighter Interface Division
Human Effectiveness Directorate
711 Human Performance Wing

SAGE
Sage
Technologies,
Ltd.

MEMORANDUM FOR RECORD

FROM: Sage Technologies Ltd, One Ivybrook Boulevard Suite 190, Warminster PA 18974

SUBJECT: Agreement and Approval for Public Release of SBIR Data Rights (FA8650-15-M-6659)

1. Sage Technologies Ltd. waives its SBIR Data Rights for the attached technical report entitled "Low-Latency Embedded Vision Processor (LLEVS)" for SBIR Phase I contract FA8650-15-M-6659. The Government is granted an unlimited nonexclusive license to use, modify, reproduce, release, perform, display or disclose this chart set and the data contained therein.
2. The attached has been reviewed and is approved for public release, distribution unlimited. The following request is published in the interest of scientific and technical information exchange and does not constitute Government approval or disapproval of the ideas or findings.

Julius Olita
Julius Olita, President
Sage Technologies, Ltd.

3-11-16
(Date)

HOPPER.DARREL.G.1231147264
Digitally signed by HOPPER.DARREL.G.1231147264
DN: c=US, o=U.S. Government, ou=DoD, ou=PKI, ou=USAF, cn=HOPPER.DARREL.G.1231147264
Date: 2016.03.11 10:29:19 -05'00'

Darrel G. Hopper
Principal Electronics Engineer
711 HPW/RHCV

(Date)

MCKINLEY.ROBERT.C.1230130538
Digitally signed by MCKINLEY.ROBERT.C.1230130538
DN: c=US, o=U.S. Government, ou=DoD, ou=PKI, ou=USAF, cn=MCKINLEY.ROBERT.C.1230130538
Date: 2016.03.17 09:03:58 -04'00'

(Branch Chief Title/Signature)

(Date)

DENIO.DONALD.1017930490
Digitally signed by DENIO.DONALD.1017930490
DN: c=US, o=U.S. Government, ou=DoD, ou=PKI, ou=USAF, cn=DENIO.DONALD.1017930490
Date: 2016.03.16 15:13:49 -04'00'

(Wing STINFO Title/PA Submitter /Signature)

(Date)

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE ***(DD-MM-YY)*** 09-03-16	**2. REPORT TYPE** Final	**3. DATES COVERED** ***(From - To)*** 30 July 2015 – 1 Mar 2016

4. TITLE AND SUBTITLE

Low-Latency Embedded Vision Processor (LLEVS)

5a. CONTRACT NUMBER
FA8650-15-M-6659

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
65502F

6. AUTHOR(S)

Greg Cream*, Wesley Sheridan*, and Prawat Nagvajara**

5d. PROJECT NUMBER
3005

5e. TASK NUMBER
CV

5f. WORK UNIT NUMBER
H0LW (3005CV85)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
*Sage Technologies Ltd, One Ivybrook Boulevard Ste 190, Warminster PA 18974
**Drexel University, 3141 Chestnut St, Philadelphia PA 19104

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Material Command
Air Force Research Laboratory
711 Human Performance Wing, Human Effectiveness Directorate
Warfighter Interface Division, Battlespace Visualization Branch
Wright-Patterson Air Force Base OH 45433-7022

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
USAF AFMC 711 HPW/RHCV

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RH-WP-TR-2016-0017

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A: Approved for public release.

13. SUPPLEMENTARY NOTES
This is a Small Business Technology Transfer (STTR) Phase I report developed for a purchase order awarded under STTR topic AF15A-T13 Low-Latency Embedded Vision Processor (LLEVS). Contains Patentable Information (patent being pursued but not approved). 88ABW Cleared 04/13/2016; 88ABW-2016-1882.

14. ABSTRACT

This report describes the efforts of a Small Business Technology Transfer (STTR) project to develop and analyze a Low Latency Enhanced Vision System (LLEVS). The LLEVS is an image processing engine capable of supporting image processing functions in helmet mounted and related mobile applications with less than 1 frame latency and at power levels less than 6 watts. The LLEVS must support a range of requirements the most difficult of which is comprised of two cameras and two displays of 5 megapixel resolution operating at 96 frames per second. The baseline LLEVS was predicated on using an imaging processing firmware suite that was developed to support helmet mounted systems. Empirical and estimated parametrics were derived from this suite and used to project the resources, performance and power projections for intermediate and ultimately objective levels for an LLEVS implementation. The data supporting the analyses has been acquired from physical measurements, development tools, simulations and estimates derived from the projection of measured parameters. The results have been used to define the path for an LLEVS capable of the objective performance. This report addresses the technology roadmap to assess the prevailing and projected technologies that will enable the design and implementation of the target processor capabilities.

15. SUBJECT TERMS vision processor, algorithms, low-latency video processing, embedded image processor, wearable electronics, helmet-mounted systems, alternative night/day imaging technologies, multi-source image fusion, battlespace visualization, dismounted operations

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Darrel G. Hopper
Unclassified	Unclassified	Unclassified	SAR	62	**19b. TELEPHONE NUMBER** *(Include Area Code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

FOREWORD

The PE65502F $149,972.00 STTR Phase I purchase order FA8650-15-M-6659 Air Force Research Laboratory (AFRL) Battlespace Visualization Branch (RHCV) Work unit H0LW (3005CV85), was awarded to Sage Technologies Ltd on 30 Jul 2015 with basic effort scheduled to end 1 May 2016.

This effort was awarded under the STTR Topic "AF15A-T13 Low-Latency Embedded Vision Processor (LLEVS)" program. The OBJECTIVE and DESCRIPTION of this topic are as follows.

OBJECTIVE:
Develop architectures for an embedded processor capable of implementing the image processing algorithms required for a digital helmet-mounted display for dismounted soldiers.

DESCRIPTION:
High-performance, low-power, and low-latency processing is needed to perform image processing algorithms in next-generation aircraft helmet systems. New architectures and technologies are needed to respond to issues arising due to continued shrinking of semiconductor fabrication process geometries. Existing approaches have not satisfied end-user needs, such as multi-channel I/O, low-latency, large image sizes, and high frame rates. Novel architectures are needed, and alternatives promising improved power efficiencies of the processor clock tree, logic, memory, and chip I/O must be investigated. Familiarity with the important algorithms, such as distortion correction, multi-spectral/multi-modal fusion, and head-tracking, is required to ensure the solution can meet the challenging performance requirements. Consideration must also be given to the robustness of the processor, as a warfighter's life may depend on its reliability in a challenging electromagnetic radiating environment. Finally, consideration must be given to a solution that can not only be applied to the digital binocular helmet-mounted display, but also to a wider set of applications that can take advantage of high-performance, low-power, low-latency image processing. The processor requirements for the vision processor ASIC developed under the DARPA Multispectral Adaptive Networked Tactical Imaging System (MANTIS) program (2003-2010) is a good example. It was originally conceived to fuse inputs from five helmet-mounted electro-optical sensors operating in the visible-near infrared (VNIR x 2), short wave infrared (SWIR x 2), and long wave infrared (LWIR) bands and generate two synchronized SXGA video outputs at 60 Hz to a pair of microdisplays. However, it resulted in a processor that ingested three sensors (one each VNIR, SWIR, LWIR) and generated just one video output at 30 Hz due to the technical approach (e.g., architecture, microelectronic technologies) and processor geometry (90 nm) used at the time [1]. Under this program, a vision processor for helmet systems (VPHS) is required to enable the design and fabrication of a digital binocular helmet-mounted display (HMD) having all source image fusion with two video outputs. Binocular systems needed by warfighters require threshold (objective) performance comprising two synchronized video outputs, each at 60 Hz x 1.3 Mpx/frame x 8b/px = 0.624 Gbps (5Mpx x 8b x 96Hz = 3.84 Gbps), and must be capable of ingesting matching resolution video (in Mbps) from multiple sources (on-helmet or on-aircraft) comprising various mixtures of live video from sensors, synthetic imagery, and overlay symbology. Monocular systems with similar processing requirements are also of interest. To understand the power and latency impacts of a total solution, it is necessary to both demonstrate a representative set of algorithms on the proposed processor, and to measure the system level performance, including required peripherals, such as external memory. Demonstrated success against a metric, such as GOPS/W or GFLOPS/W, is not sufficient, as it only provides a partial picture of a solution, potentially pushing off the power requirements and demanding physical capabilities to other parts of the system.

1.0 SUMMARY

This report describes the efforts of a Small Business Technology Transfer project to develop and analyze a Low Latency Embedded Vision Processor (LLEVS). The LLEVS is an advanced image processing engine capable of supporting complex image processing functions in helmet mounted and related mobile applications with minimal latency (less than 1 frame) and at extremely low power levels. The LLEVS must support a range of performance requirements the most difficult of which comprises two cameras and two displays of 5 megapixel resolution operating at 96 frames per second. The entire helmet system suite should require approximately ten watts and weigh less than two pounds.

This project has been conducted by Sage Technologies along with support from our academic partner, Drexel University. The effort has been pursued in two paths identified as "low risk" and "high risk" approaches to distinguish both the difficulties in achieving practical solutions to an implementation and the potential of realizing a successful conclusion. Drexel University pursued a high risk approach by focusing on the development and analysis of advanced imaging algorithms not previously available to LLEVS based types of applications. Drexel's experience in high volume data processing and transfer is fundamental to the subject imaging initiative. Sage pursued the low risk approach in order to build on its expertise in helmet mounted system applications and the implementation of practical solutions to imaging system challenges. Sage built its approach on existing imaging system hardware and firmware to establish real baseline performance metrics, and extrapolated the results and architecture to achieve the goals and target performance characteristics of the objective requirements.

The baseline LLEVS was predicated on the Acadia II image processor and an imaging processing firmware suite that was specifically developed to support helmet mounted systems and is presently employed in several active applications. The basic threshold levels of LLEVS performance are just achievable with this Acadia II baseline configuration. Empirical and estimated parametrics were derived from this suite and used to scale up and project the resources, performance and power projections for intermediate and ultimately objective levels for an LLEVS implementation. Data required to conduct the performance analyses have been acquired from physical measurements, development tools that afford simulation results and estimates derived from the projection of measured parameters. The results have been used, along with previous processor technology assessments, to define the path for an LLEVS capable of the objective performance.

The results of this effort are compelling in demonstrating that an LLEVS can be implemented that will achieve the most demanding processor performance requirements. Objective performance goals will require the most advanced Xilinx UltraScale+ MPSoC technologies, but these FPGA devices are becoming available commercially along with their development support tools. In addition preliminary assessment of the high risk advanced algorithms and methods suggests that their adaptation to this FPGA technology can also be accommodated which could provide operating margin in a target application. This could facilitate the use of a lower cost UltraScale+ family member without sacrificing processor performance. The LLEVS target performance seems assured, but should be validated with a detailed design, simulation and analysis with accompanying stress testing of a breadboard or prototype system.

This effort entails the development and documentation of the Vision Processor for Helmet System (VPHS) Requirements and functionality in partnership with the customer and key technology subject matter experts (SMEs). The requirements drive the unit performance specification and assess technology options to support the development of the VPHS development plan and design. This effort evaluates the feasibility of various candidate imaging processor technologies and signal processing techniques that offer the greatest impact for implementation in both low and high risk approaches. Subsystem components are identified and assessed for performance and integration with the VPHS design. Processor technologies, architecture configurations, VPHS hardware and processing techniques, and system interface requirements needed to design the Prototype device and requiring development during Phase II are identified. A work plan is developed that presents a rapid path to designing, simulating and validation of a Prototype VPHS during Phase II. Table 1 lists all of the LLEVS tasks in the Statement of Work.

Throughout this report the acronyms VPHS and LLEVS are used interchangeable to reflect the embedded processor for digital helmet mounted systems.

Table 1. LLEVS Statement of Work (SoW) Tasks

Task	Description
1	Define the VPHS requirements and performance specification. Working with the Government TPOS, determine the preferred technical approaches versus sensor performance requirements that might be best served by the VPHS. Once the overall requirements are determined, detailed design and requirements are generated to specify the functional parameters for each of the approaches and configuration candidates.
2	Low Risk VPHS. Analyze the newest COTS processor alternatives that support the transfer of Acadia II firmware as is currently hosted on the DEVS and BMAIS systems, and conduct a preliminary design that will execute the algorithms identified in Task 1. Host the relevant firmware elements on simulation or other development tools available with that processor suite, and monitor the firmware under execution. Generate and record the performance results for comparison with the goals, and forecast simulation/design options for a Phase II detail design in order to achieve threshold and objective performance goals.
3	High Risk VPHS. This task conducts an assessment of the potential processor technologies and devices, identifies candidate characteristics and project candidate configurations and innovative architecture approaches that can achieve the target performance goals. Where possible model the process and predict performance characteristics. Alternatively, predict performance through analysis and estimation based on similar or comparable devices and configurations. This task results in an advanced technology matrix containing the technology candidates most qualified to achieve the performance identified in the solicitation and specified in the requirements from Task 1. SWaP characteristics are identified for tradeoff assessment, along with performance tradeoffs for potential candidates. A tradeoff table is generated to guide the Phase II and Phase III prototype design and fabrication
4	Candidate configuration and operational issues. This task assesses the constraints and factors related to the implementation options of the VPHS regarding installation configurations, power and power management analyses, thermal management and distributed architecture potential. The Task 4 effort is also concerned with the identification, specification and interfacing of the external subsystems and data sources with the device. The establishment of data interfaces in terms of data transfer rates, formats and types are formulated and codified. Alternatives and tradeoffs for implementation are contrasted along with the preferred methods for the Phase II development plan. Particular attention is focused on the means to establish data and processing throughput while minimizing SWaP and latency impact.
5	Prepare Final Report and Phase II Development Pan. This task provides direct feedback to the development activities from the beginning of the project (effectively from the Kickoff Meeting) by way of bimonthly status reports through the end of Month 6, and a Final Report describing the Phase I effort. The Development Plan provides the planning detail for Phase II effort to design and simulate a prototype VPHS system

2.0 INTRODUCTION

This report is submitted in compliance with contract FA8650-15-M-6659, a Phase I Small Business Technology Transfer (STTR) project, tasked with the development of two approaches to design and analyze an image processor for a Helmet Mounted System (HMS). The first effort was a low risk technical approach that established a baseline capability using existing image processor technology hardware and firmware to acquire initial performance metrics, and serve as a foundation for technology advances and performance upgrades. This baseline capability is already in existing technology and helmet mounted systems, and is able to meet the processor threshold performance requirements. The second effort was the high risk approach where advanced algorithms and processor architectures for processing images were investigated. Sage lead the low risk approach and Drexel University lead the high risk approach. The risk in the names of the two efforts refers to the difficulties in achieving successful outcomes of the efforts and in the potential for yielding a usable image processor solution capable of meeting the objective performance requirements. When the two design concepts were completed, an estimate of the power, frame latency, weight, and size of each of the processors and any peripheral devices was developed. At the end of this report the risks and benefits associated with each approach have been identified.

2.1 Background

There exists an ever growing need for advances in the technologies that support the warfighter in carrying out his missions and affording the highest probability of success. The supporting technologies include a diverse array of components from sensors to processors and to displays. They comprise a suite of electronic subsystems that evolve along separate paths at different rates, but which must all be integrated into a cohesive unit that becomes the warfighter's sensor system. The capability that facilitates the integration of these subsystems and affords the enhancement of their capabilities through processing is the processor and its firmware. The particular focus of this STTR was to evaluate candidate image processors required for helmet mounted systems where performance requirements are demanding, and where other issues related to Size, Weight and Power (SWaP) pose significant challenges.

Advances in digital sensors and digital displays now require the development of improved digital processing capacity that can be integrated within helmet space and mass limitations. The total head-born weight for helmet systems must be less than 5 lbs including the shell and any embedded electronics components (e.g. HMD system). The weight budget allocation to the helmet-mounted components of the HMD system is less than 2 lb, including sensors, processors, micro displays, optics, batteries, and cables. Also, the total power dissipation for the in-helmet components, which is dominated by the in-helmet processor and sensors, must be less than than 10 W to avoid the need for active in-helmet cooling. Prior solutions to the in-helmet processor required for digital all-source imaging have yet to meet these mass and power requirements. However, efforts to date have been based on older levels of microelectronics fabrication technology that are no longer state-of-the art: e.g. 90-nm design rule Application Specific Integrated Circuit (ASIC) technology, or fifth-generation Field Programmable Gate Array (FPGA) devices.

The path to realization of a VPHS is both supported and driven by the continuing advances of integrated circuit fabrication technology in achieving not only denser electronic packaging, but also complex logic configurations that have evolved over the past decade of System on a Chip (SoC) device developments. While the advances on these two fronts form the bases for a VPHS creation, the target capability must be considered in the context of the intended application(s) as well as the legacy represented by prior VPHS type devices, systems and applications. During the VPHS Phase I project and other Sage supported work, the effort examined those issues and features that represent the relevant legacy and evolving technologies pertaining to a VPHS and developed a path that will yield a "best fit" solution within the context of ever advancing technologies and increased performance and application requirements. Particular attention has been devoted to SWaP as they impact helmet mounted systems and the performance requirements to achieve the resolution and frame rate goals as they drive the throughput and power demands of the target technologies.

2.1.1 VPHS Phase I.

Research preliminary to this LLEVS Phase I effort was accomplished by Sage in the VPHS Phase I project [1]. At the end of the Phase I VPHS effort it was concluded that both rapid and substantial changes are occurring in the technology areas supporting and affecting the VPHS processor selection criteria. Of special interest are consumer based influences as they impact the mobile device candidates. In particular are those issues related to power consumption and the battery/recharge requirements and size/weight issues as they relate to the convenience and mobility of the devices. Similar issues exist for the markets beyond the consumer base, but the motivations and requirements serve to drive the performance characteristics to higher levels to achieve application needs and maintain processing capabilities commensurate with peripheral device processor requirements. Regardless of the market demands, the developers and providers of the target technologies find themselves in an ongoing competition to deliver the processor technologies with the lowest size, weight and power, with the required performance and at a competitive price.

The requirements for LLEVS Phase I evolve from the development spiral done for the VPHS Phase I project leading to a design concept and architecture. In the VPHS Phase I multiple types of processor technologies were investigated to determine which processor technology best suited HMD systems. In LLEVS Phase I the candidate technology was used to develop a concept and conduct simulations to better determine how well the candidate technology will be able to meet the requirements. The requirements established by the government in the solicitation for the VPHS Phase I project were refined for the LLEVS Phase I effort. The processor requirements are listed below.

Support an HMD system of ≤ 2lbs
Operating power of the system ≤ 10W
Support a binocular imaging system – (2) synchronized video inputs/outputs
Threshold level: 1.3 Mpix/frame x 14b x60Hz/1.3 Mpix x 8b/pix x 60Hz
Objective level: 5 Mpix/frame x 14b x 96Hz/5 Mpix x 8b/pix x 96Hz
Support ingesting of matching resolutions video from multiple sources
Function with < 1 frame latency

As a result of the investigation and analysis during the VPHS Phase I effort and the evaluation of a large number of candidates across a variety of architectures, two candidates stand out as being particularly well suited to serve as the VPHS processor. A FPGA candidate is deemed especially effective image processor because the firmware developed for one generation of FPGA can be ported to the next generation. With this firmware portability, the continuing technology evolution can be used to advance the processor design and allow the objective performance levels to be achievable in the near term (1-2 years).

2.1.2 LLEVS Phase I Approach.

The LLEVS Phase I project consisted of two concurrent efforts. One effort was a low risk technical approach that established a baseline capability using existing image processor technology hardware and firmware to acquire initial performance metrics, and serve as a foundation for technology advances and performance upgrades. The second effort was a high risk approach where different algorithms and processor methods for processing images were investigated. Sage lead the low risk approach and Drexel University lead the high risk approach.

In this report the two approaches will be discussed separately. In places where the investigations intersect the two topics will be compared. Subheadings will indicate which approach is being discussed.

2.2 Low-Risk Approach - Introduction

The object of the low risk approach is to consider current or near term technologies for the development of an image processing system that will satisfy all or most of the target performance thresholds and objectives and use algorithms previously developed and hosted in the Acadia II. The Acadia II and its image processing firmware provide a solid basis on which to conduct data analysis for performance projections. The data acquired through measurements, simulation and estimation provide the requisite platform for architecture design and performance projections. The processor resource requirements and performance metrics were explored in the VPHS Phase I effort where it was determined that advanced FPGA and SoC technologies were the most suitable candidates for the LLEVS in lieu of an ASIC development such as the Acadia II. Several FPGA sources exist for consideration as the LLEVS processor, but the preferred candidate for this development and analysis is the Xlinx product family of devices. This determination is supported by: (1) The Acadia II firmware has been ported to the Xilinx Zynq 7 devices; (2) The Xilinx products are supported by an extensive array of development and simulation tools; (3) The Xilinx UltraScale+ MPSoC family of advanced technology processors are entering the market with performance levels that virtually achieve the objective hardware performance goals.

2.3 High-Risk Approach - Introduction

A primary objective of the high-risk approach is to develop and analyze advanced image processing algorithms that could be hosted in FPGA – based embedded vision processors. The aim is to develop a faster more efficient image processing progression.

The high risk approach is also concerned with the development of an LLEVS processor that considers the latest and anticipated future devices and system architectures that could be designed to achieve the performance goals for the objective processor implementation. The high risk approach is based on hardware accelerators using IP (Intellectual Property) cores to improve performance, latency and power efficiency. IP cores are Hardware Description Language (HDL) codes that are developed and occasionally available through commercial sources. Example IP cores are available in the library of Integrated Design Environment (IDE) tools. Designers can select and instantiate the IP core components in HDL codes and/or schematic captures. The high risk approach builds on an advanced technology base and augments its performance with architecture, algorithms and technology enhancements.

The Field Programmable Gate Array (FPGA) is a system of reconfigurable hardware peripherals and high-performance processors that can provide flexible design development at a lower dollar cost and alternative technology to Application Specific Integrated Circuit (ASIC). In an FPGA-based embedded vision processor, the peripherals include reconfigurable hardware cores for video streaming, pixel/color correction and Digital Signal Processing (DSP). It is expected that a pipeline of hardware cores implemented on FPGA can provide the required low latency in processing the input video from the sensors to the output display. It is also expected that FPGA-based embedded vision processor will meet the Size, Weight and Power (SWaP) requirements.

The computation cores considered for the proposed FPGA platform are based on researched algorithms and on those used in the SRI Acadia II [2]. The SRI Acadia II is an Application Specific Integrated Circuit (ASIC) vision processor for real-time multi-sensor video fusion, video stabilization and video tracking.

The LLEVS high-risk approach tasks include:

1. Identification of common DSP algorithms used in vision systems, in particular, video fusion, stabilization and moving-object tracking in which hardware cores can be developed to speed up the performance thus reducing the latency
2. Development of architecture models for video stream-processing using the hardware cores
3. Obtaining, analyzing and reporting the projected performance compared to software and the Graphic Processing Unit (GPU)

3.0 METHODS, ASSUMPTIONS AND PROCEDURES

The methods assumptions and procedures for the Low Risk and High Risk approach are very different. Because of that difference the approaches will be discussed separately in this section.

3.1 Low Risk Technical Approach

3.1.1 Summary.

Using a known baseline image processor and image processing firmware, it was possible to establish realistic and verifiable performance metrics to initiate the low risk development. The image processor currently used by Sage is the Acadia II, the processor supporting the DEVS and BMAIS helmet mounted imaging systems. The Acadia II processor is capable of achieving the performance goals of the threshold requirements, which offers additional motivation for its use as a baseline for comparison. However, the Acadia II platform has reached its processing limits with the demands for higher resolution and frame rate sensors and with additional digital inputs needed for new integrated HMD designs. The Acadia II was used as the basis for comparison to the newer technology MPSoC platforms. A block diagram in Figure 2 depicts the high level architecture of the Acadia II. As an ASIC the hardware design of the Acadia II is specific to that of an image processor and cannot be reconfigured. The Acadia II is an extremely capable SoC (system on a chip), Application Specific Integrated Circuit (ASIC), that is currently available as a production product. Unfortunately the development cost and time scales for an ASIC are considerable and discourage further advances along that integrated approach.

The approach was started with the assessment of processor resources required to host the baseline processor and firmware, and subsequently extrapolated to meet the VPHS objective level requirements. Latency and power consumption were the focus issues as the baseline processor configuration was expanded to the objective performance goals. Throughout this process the latency and power were monitored through a combination of measured, simulated and estimated parameters. This effort then supported a preliminary design based on algorithms to be used for the estimation of power consumption and latency. Once the preliminary design was completed, several candidate MPSoCs were chosen. The design was then simulated and tests were conducted using the Acadia II baseline data for comparison.

3.1.2 Low Risk Technical Methods.

To be able to develop estimates for power, frame latency, weight and size, a family of processors needed to be selected. Using the experience with porting Acadia II algorithms in FPGAs, and assuming the application of the current Xilinx Zynq Series 7000 devices, the FPGA resource estimates for the VPHS application were established and are listed in the Table 2

Table 2. Estimated FPGA Resources Required for VPHS

	FF	LUTs	DSP48	BRAM16	MHz
Threshold	177,579	189,587	634	994	100
Objective	459,901	492,815	1,902	2,886	200

Power estimates have been generated for the Xilinx devices using available estimation tools from Xilinx and including the power required by peripheral devices to support the device operation.

Xilinx FPGA Power Estimates are represented in Table 3. The chip I/O power is included in the chip power. The DDR memory is added separately.

Table 3. Estimated Power for VPHS Using MPSoCs

Application	FPGA	#chips	Chip [W]	Mem [W]	ARM [W]	Total [W]	Availability
Threshold	Zynq 7100	1	5.3	0.6	-	6	Now
Objective	Virtex 7 960T	1	16	1.7	1	18.7	Now
Objective	Zynq UltraScale	1	10.4	1.7	-	10.3	2015
Objective	Zynq UltraScale+	1	7.1	1.7	-	6.47	2016

Based on the power estimates, it was possible to zero in on candidate components to use for the estimations. The choices were all members of the Xilinx Zynq Series 7000 and UltraScale+ families of MPSoCs. Because of power concerns it was decided to use the Xilinx family member that would provide sufficient processing capacity, minimal frame latency and limited chip power requirements. Analysis of power estimates and individual MPSoC specifications enabled the selection of the MPSoC to be used for the simulations.

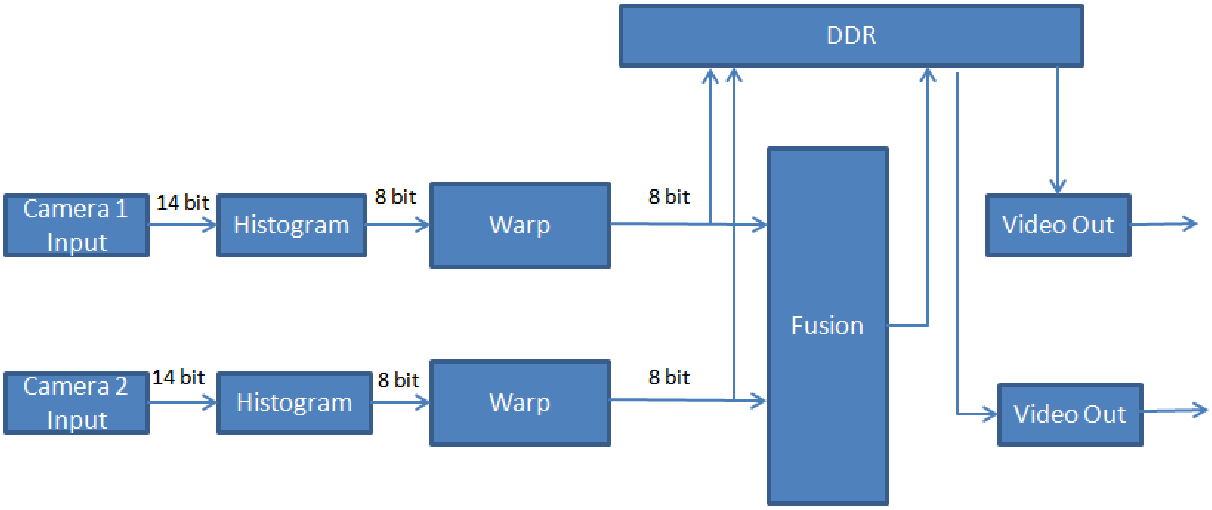
The following list of development tools was used to generate the estimates in Table 3 and in the tables in section 4.

- Xilinx Early Power Estimator Tool (Series7_XPE_2015_3) for Zynq estimates
- Xilinx Early Power Estimator Tool (Series7_XPE_2015_4) for Zynq Ultrascale + MPSoC
- Micron DDR3_Power_Calc (v0.93), Micron DDR3L_Power_Calc (v0.93)
- Micron DDR4_Power_Calc (v1.0)

3.1.3 Preliminary Design.

A preliminary design was needed before the development of the estimation could proceed. The design was based on the selection of SRI algorithms to be used during the estimation and analysis. SRI had previously ascertained latency measurements in the development of the Binocular Multispectral Adaptive Imaging System (BMAIS) and Digital Enhanced Vision System (DEVS) Helmet Mounted Display (HMD) systems. From these measurements it was determined the two extremes for frame latency were the pass through mode and the multi-spectral, multi-source fusion mode. The pass through mode had the least amount of latency and the fusion mode contained the greatest amount. These two algorithms provide the extremes for the analysis.

With the algorithms determined the preliminary design proceeded. Inputs were provided for each of two cameras, and outputs were provided for each of two displays. The functional blocks in Figure 1 below illustrate the pixel flow through the image processing function. Each camera input is processed through a Histogram and Warp process. In the pass through mode the data is then directed to the DDR memory ready for display or in the fusion mode, through the fusion algorithm and then to the DDR memory ready to be displayed.



Note: Parallel paths needed to split video into multiple paths when pixel clock exceeds 150 MHz in the FPGA are not shown in this drawing.

Figure 1. FPGA/MPSoC Processing Functional Block Diagram

3.1.4 Acadia II.

The Acadia II hardware design is very ridged. The ASIC design does not permit the restructuring of inputs, outputs or internal logic paths. The paths through the Acadia II only allow minimum change. The Acadia is dedicated to image processing for the sensors and displays available 10 years ago when it was designed. The interfaces into and out of the Acadia were designed for data rates that were lower than the current and near term future data rates. The data rates are being driven by increases in the sensor and display resolution and frame rates. Figure 2 [3]shows the high level functional architecture of the Acadia II.

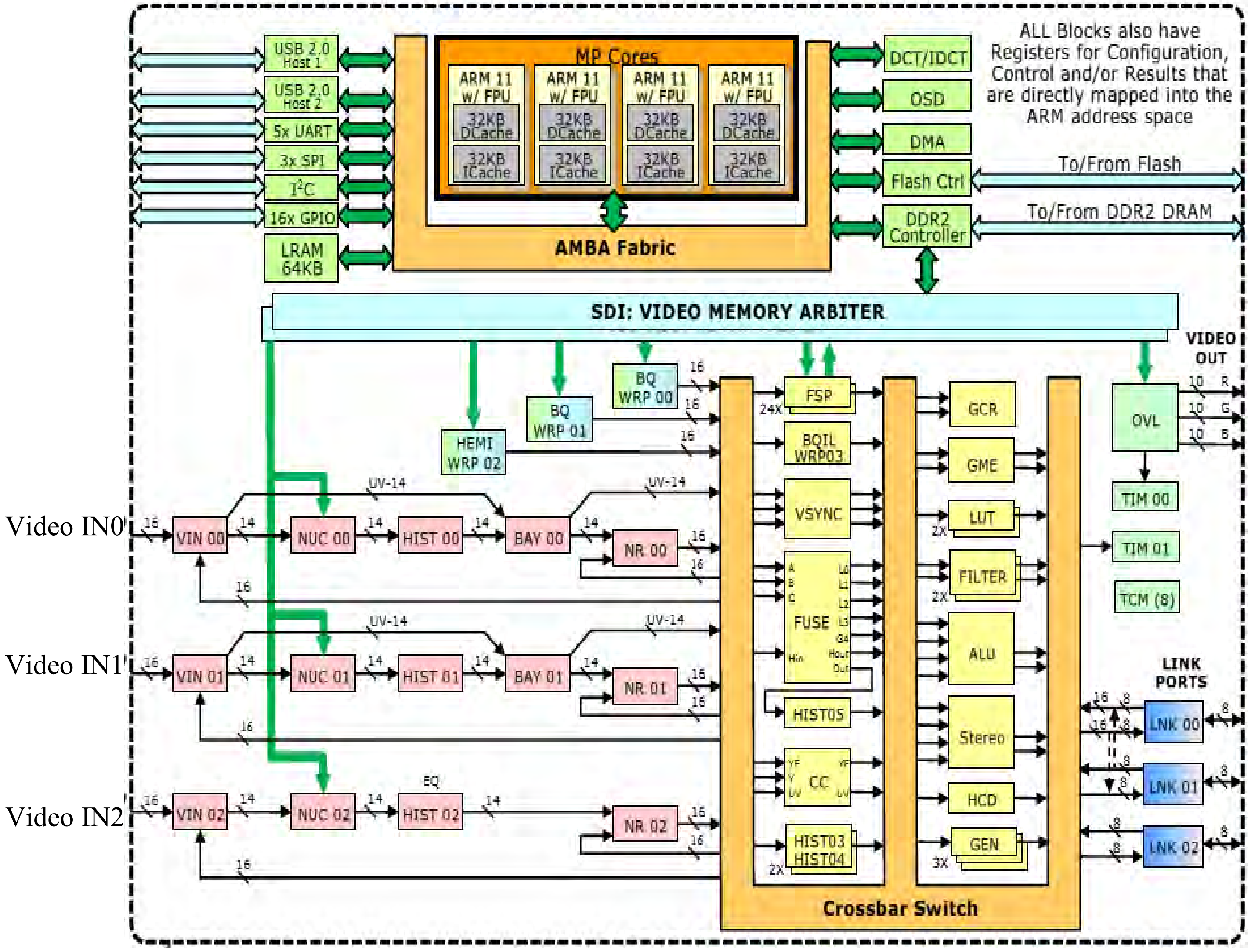


Figure 2. Acadia II Top Level Block Diagram

3.1.5 Xilinx Multiple Processor System on Chip (MPSoC).

Several processors were chosen from the Xilinx family for the estimation tests. The candidate processor platforms chosen for estimation were the following:

- New Platforms
    - The Xilinx, Zynq 7Z020, 7Z030, 7Z045, 7Z100
    - The Xilinx, Zynq, UltraScale+ ZU9EG
- Current Platform
    - Acadia II

Because the Acadia II is an ASIC, its specifications cannot be directly compared with those of an MPSoC.

The Xilinx Zynq 7 Series MPSoCs all have dual ARM Core processors with a maximum processor frequency of 1 GHz. The Zynq UltraScale+ MPSoCs all have quad-core Cortex-A53 MPCore processors with a maximum processor frequency of 1.5 GHz and Dual-core Cortex-R5 MPCore processors with a maximum processor frequency of 600 MHz. All of the Xilinx Zynq MPSoCs have a host of different Input and Output (I/O) capabilities.

The rest of the high level specifications for the Xilinx MPSoCs being simulated are contained in Table 3 below. The major differences between the MPSoCs are the number of logic devices and the amount of onboard memory they contain, the selection of interfaces to the outside world and the chip power usage. The interface capabilities of the Xilinx chips are described in the peripheral section of Table 4 [4][5].

Table 4. High Level Characteristics for the Xilinx Devices Simulated for the Analysis

Description	Specs. - Zynq 7Z020	Specs. Zynq 7Z030	Specs. Zynq 7Z045	Specs. Zynq 7Z0100	Specs. ZU9EG
Processor Core	Dual ARM Core Processors	Dual ARM Core Processors	Dual ARM Core Processors	Dual ARM Core Processors	a. Quad-Core ARM Processors, b. Dual-Core ARM
Max. Processor Frequency	1 GHz	1 GHz	1 GHz	1 GHz	a. 1.5 GHz b. 600 MHz
L1 Cache	32 KB Instruction & 32 KB Data per processor	32 KB Instruction & 32 KB Data per processor	32 KB Instruction & 32 KB Data per processor	32 KB Instruction & 32 KB Data per processor	32KB per core
L2 Cach	512 KB	512 KB	512 KB	512 KB	1 MB
On-Chip Memory	256 KB	256 KB	256 KB	256 KB	a. 256 KB b. 128 KB
External Memory Support	DDR3, DDR3L, DDR2, LPDDR2	DDR3, DDR3L, DDR2, LPDDR2	DDR3, DDR3L, DDR2, LPDDR2	DDR3, DDR3L, DDR2, LPDDR2	X32/x64 DDR3, DDR3L, DDR2, LPDDR2
DMA Channels	8	8	8	8	NA
Peripherals	2-UART, 2-CAN, 2-I2C, 2-SPI, 4-32b GPIO	2-UART, 2-CAN, 2-I2C, 2-SPI, 4-32b GPIO	2-UART, 2-CAN, 2-I2C, 2-SPI, 4-32b GPIO	2-UART, 2-CAN, 2-I2C, 2-SPI, 4-32b GPIO	PCLe Gen2 x4, 2x USB3.0, SATA 3.1 Display Port, 4x Tri-mode Gigabit Ethernet, 2xUSB 2.0
Logic Cells	85K	125K	350K	444K	600K
Look-Up Tables	53K	79K	219K	277K	274K
Flip Flops	106K	157K	437K	554K	548K
Total Block Ram	4.9 Mb	9.3 Mb	19.1 Mb	26.5 Mb	32.1 Mb
Programmable DSP Slices	220	400	900	2020	2,520

Figure 3 shows a high level block diagram of the Zynq Series 7000 architecture. The Zynq Series 7000 products integrate a feature-rich dual-core ARM® Cortex™-A9 based processing system (PS) and 28 nm Xilinx programmable logic (PL) in a single device. The ARM Cortex-A9 CPUs are the heart of the PS and also include on-chip memory, external memory interfaces, and a set of peripheral connectivity interfaces. The Zynq-7000 architecture enables implementation of custom logic in the PL and custom software in the PS. It allows for the realization of unique and differentiated system functions. The block diagram of the Zynq 7000 series in Figure 3 [4], with the characteristics in Table 4 above, show the real versatility of this chip.

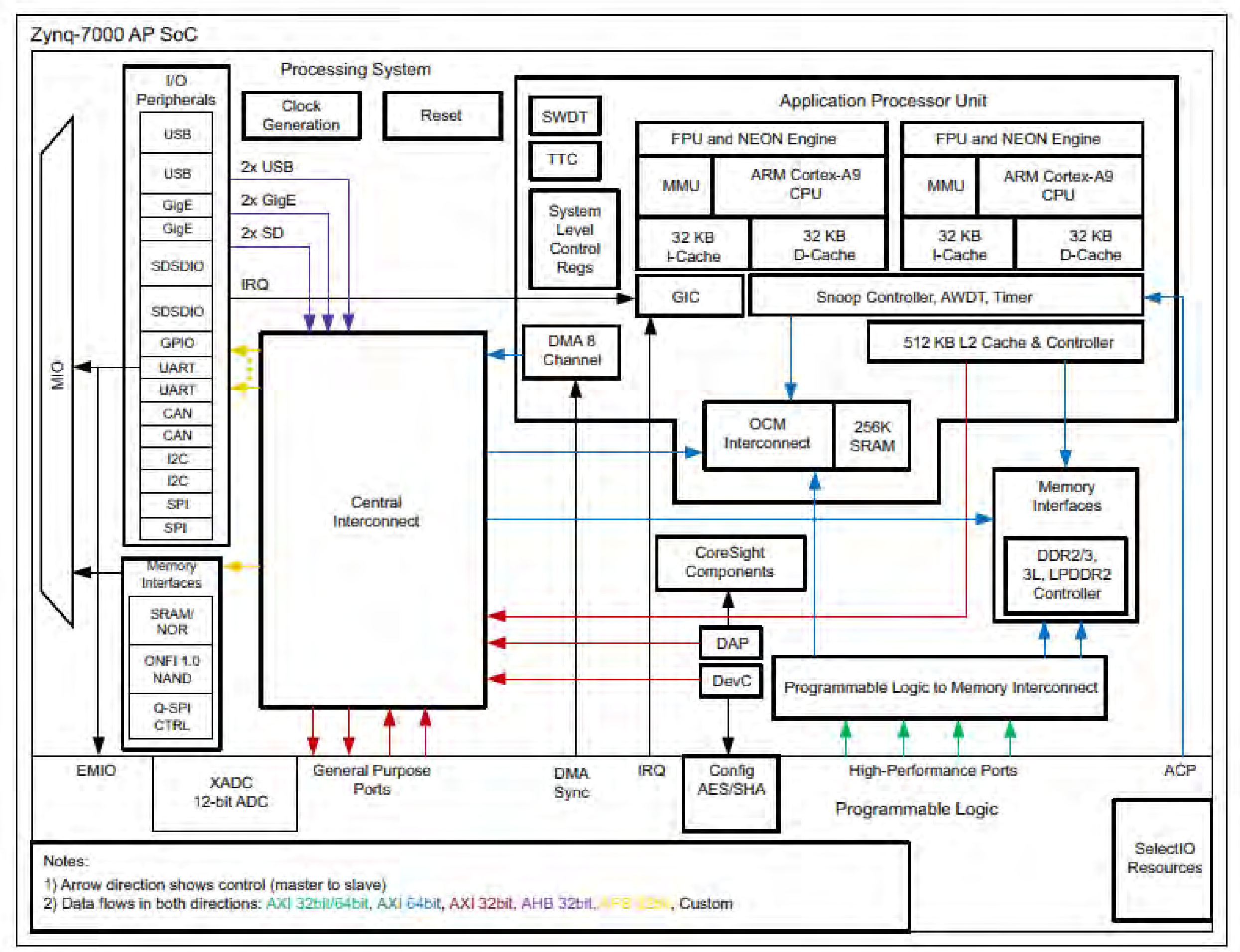


Figure 3. Basic Block Diagrams for Zynq Series 7000

Figure 4 [6] shows a high level block diagram of the UltraScale+ architecture.
Zynq UltraScale+ MPSoC is built upon the next-generation 16nm FinFET process node and contains a scalable 32- or 64-bit multiprocessor CPU. The UltraScale+ combines the Quad ARM® v8-based Cortex®-A53 high-performance energy-efficient 64-bit application processor with the ARM Cortex-R5 real-time processor and the UltraScale architecture to create an all programmable MPSoCs with a wide range of interconnect options, DSP blocks, and programmable logic choices. The UltraScale+ has capabilities beyond the Series 7000. The UltraScale+ has more external interfaces, additional processors, more logic devices, faster clock speeds and lower power consumption requirements.

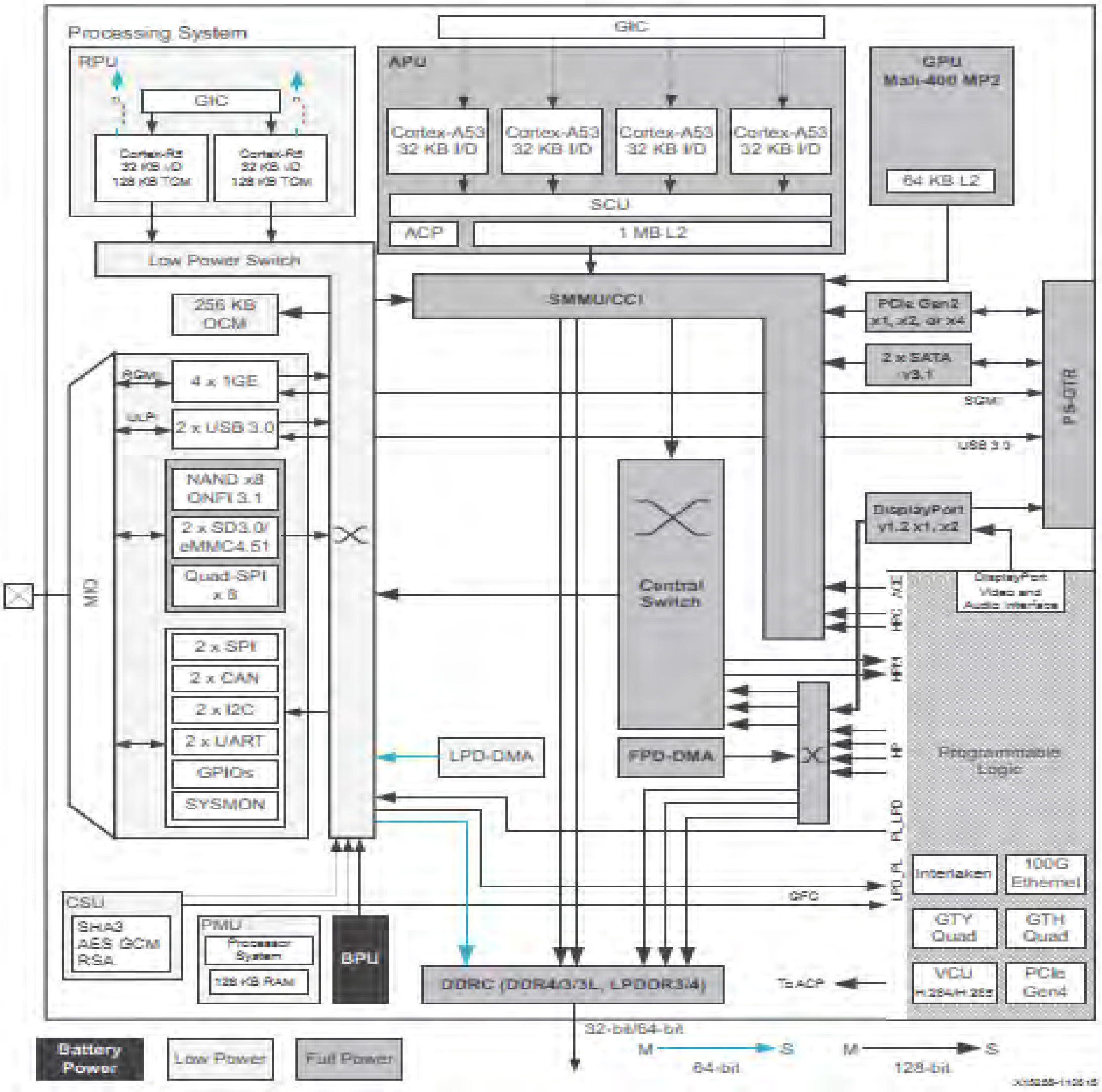


Figure 4. Basic Block diagram for Zynq UltraScale+

The block diagrams for the Acadia II ASIC and the two MPSoCs illustrate the difference in the two types of devices. The Acadia II ASIC has logic and I/O specifically designed for image processing. As such only the software in the ASIC CPU can be changed.

The two MPSoCs are not designed for any specific functions. The MPSoCs can then be reconfigured for application changes. This makes the MPSoCs very flexible. This flexibility comes with a cost. In general most applications will not require all of the MPSoC's circuitry to be used. The unused circuitry consumes power. Circuitry designed for the MPSoCs may not be as efficient as the dedicated circuitry in an ASIC. One result of this analysis will determine if the FPGA technology in the Zynq Series 7000 and UltraScale+ respectively, will be able to overcome the advantages of the ASIC in power and efficiency. Preliminary findings are found in Section 4.0.

The UltraScale+ having lower power needs and faster clock rates processes more image data at faster rates. When higher image resolutions and faster frame rates are used the processing clock rate is important to meet the objective requirement. The UltraScale+ has the most flexibility of the Zynq family of chips. With six internal processors, and reconfigurable I/O, the UltraScale+ can be configured to meet several different applications.

3.1.6 Analysis.

A plan for analysis was developed to determine what data would be collected and the method that would be used to do the collection. The measurements collected during each scenario are as follows:

- Power Consumption (Watts)
    - Dynamic
    - Static
- Number of CPUs used (Number)
- RAM (MB)
- Flip Flops Used (Number)

The preliminary design was used with the algorithms to form a basis for the latency analysis and estimates. The power was estimated using the a Xilinx tool for the MPSoC and a Micron power estimating tool for the external memory power usage.

3.1.7 Scenarios

To provide a sufficient cross section of results, several different scenarios were developed and simulated. These scenarios included four sensor resolutions, two frame rates, two different image processing algorithms and one or two sensor inputs. Results were gathered for each scenario for both a Zynq 7 family MPSoC and a Zynq UltraScale+ MPSoC. The ZU9EG was the only UltraScale+ MPSoC characterized in the Vivado simulator at this time.
The different scenarios are listed in Table 5 below.

The design shown in the block diagram in Figure 1 above was loaded into a simulation and the data values were estimated. The data collected for the Acadia II scenarios were from measurements taken using actual Acadia II hardware and from estimations where exact measurements were not accessible.

Table 5. Estimation Scenarios

Sensor Resolution	Frame Rate (Hz)	Number Of Sensors	Algorithm	Image Processor
640 X 480 by 14 bits	60	1	Pass Thru	Acadia II
640 X 480 by 14 bits	60	2	Pass Thru	Acadia II
640 X 480 by 14 bits	60	2	Fusion	Acadia II
1280 X 1024 by 14 bits	60	1	Pass Thru	Acadia II
1280 X 1024 by 14 bits	60	2	Pass Thru	Acadia II
1280 X 1024 by 14 bits	60	2	Fusion	Acadia II
640 X 480 by 14 bits	60	1	Pass Thru	Zynq 7Z020
640 X 480 by 14 bits	60	2	Pass Thru	Zynq 7Z020
640 X 480 by 14 bits	60	2	Fusion	Zynq 7Z030
640 X 480 by 14 bits	60	1	Pass Thru	Zynq UltraScale + ZU9EG
640 X 480 by 14 bits	60	2	Pass Thru	Zynq UltraScale + ZU9EG
640 X 480 by 14 bits	60	2	Fusion	Zynq UltraScale + ZU9EG
1280 X 1024 by 14 bits	60	1	Pass Thru	Zynq 7Z030
1280 X 1024 by 14 bits	60	2	Pass Thru	Zynq 7Z030
1280 X 1024 by 14 bits	60	2	Fusion	Zynq 7Z030
1280 X 1024 by 14 bits	60	1	Pass Thru	Zynq UltraScale + ZU9EG
1280 X 1024 by 14 bits	60	2	Pass Thru	Zynq UltraScale + ZU9EG
1280 X 1024 by 14 bits	60	2	Fusion	Zynq UltraScale + ZU9EG
2560 X 2048 By 14 bits	60	1	Pass Thru	Zynq 7Z045
2560 X 2048 By 14 bits	60	2	Pass Thru	Zynq 7Z045
2560 X 2048 By 14 bits	60	2	Fusion	Zynq 7Z100
2560 X 2048 By 14 bits	60	1	Pass Thru	Zynq UltraScale + ZU9EG
2560 X 2048 By 14 bits	60	2	Pass Thru	Zynq UltraScale + ZU9EG
2560 X 2048 By 14 bits	60	2	Fusion	Zynq UltraScale + ZU9EG
2560 X 2048 By 14 bits	96	1	Pass Thru	Zynq 7Z045
2560 X 2048 By 14 bits	96	2	Pass Thru	Zynq 7Z045
2560 X 2048 By 14 bits	96	2	Fusion	Zynq 7Z100
2560 X 2048 By 14 bits	96	1	Pass Thru	Zynq UltraScale + ZU9EG
2560 X 2048 By 14 bits	96	2	Pass Thru	Zynq UltraScale + ZU9EG
2560 X 2048 By 14 bits	96	2	Fusion	Zynq UltraScale + ZU9EG

3.1.8 Premises and Assumptions

During the analysis some premises and assumptions were made. These are:

1. The power dissipated by system RAM (DDR2, DDR3L or DDR4) was included in all power estimates.
2. For Acadia II power measurements, DDR2 memory was used. For Zynq power estimates, DDR3L memory was used. For all Zynq UltraScale+ MPSoC power estimates, DDR4 memory was used.
3. Display resolution matches camera resolution.
4. Cameras can be externally triggered and synchronized.
5. Losses due to power regulation are not included.
6. Ambient temp in enclosure is 40C.
7. Pass through application includes histogram and lens distortion correction.
8. Fusion algorithm used for estimates is SRI's multi resolution, adaptive fusion algorithm
9. For video interfaces, a parallel interface was used for the two lower resolution threshold requirements while a high speed serial interface was used for the object requirements.
10. All UltraScale+ power estimates are based on the ZU9EG Zynq UltraScale+ MPSoC.

The Low Risk analysis results and discussions are contained in Section 4.0 after the High Risk methods discussion.

3.2 High-Risk Approach

The high risk approach attacks the performance and latency problem based on the fact that most video processing algorithms are suitable for pipeline hardware structure that can be implemented as IP cores. Video processing algorithms are typically dense matrix computations with regular data access patterns, in which blocks of pixels are efficiently processed using pipeline structure. Blocks of pixel data are streamed into the hardware where different computations of the algorithm are calculated on different blocks concurrently. For instance, the convolution filter algorithms compute a new pixel value as a weighted sum of the old pixel value and the surrounding pixel values. A block size is typically 3x3 or 5x5. Moreover, the new values can be computed in parallel. Latency can be further improved by parallelism using, for instance, dual or quad IP cores. The parallelism is limited to coarse-grain due to the bottleneck on the VDMA transfers between the frame buffers and the cores.

The high risk approach will restrict the computation on the IP core to fixed-point data calculations. This will give an advantage in the power consumption compared to calculations that require Floating-Point Units (FPU) because parts of the algorithms that require floating-point calculations are carried out by the processor and its FPUs. The strategy is for the processor to delegate the computation which can be done in pipeline (stream) processing using IP cores. With less computational load on the processor, the system power consumption is improved.

This section identifies the image processing algorithms to be processed and the architecture models to support the performance analysis. Analyses are then conducted on the models, including potential commercial candidates, to determine the efficacy and utility of the respective methods.

3.2.1 Task 1- Algorithms

As established by a literature review of video fusion [7-11], stabilization [12-14] and moving object tracking [15-16] algorithms used in these applications are based on multiresolution analysis (MRA), statistical analysis and filtering. The computations in these algorithms lend themselves to pipeline processing where hardware computing cores can be arranged as pipeline architecture. The computation stages for the fusion, the stabilization and moving-object tracking are as follows:

3.2.1.1 Fusion Processing. Fusion of two or three video signals from Near-visible Infrared (NVIR) sensor, Shortwave IR (SWIR) and Longwave IR (LWIR) is the main processing for Night Vision (NI) system. The system also includes pattern selective fusion where saliences are analyzed for true-color video image and enhancement. The stages include:

1. Preprocessing - Non-uniformity correction (pixel gain and offset), pixel correction, BAYER (VNIR to YUV color), noise reduction (3x3 median filter and temporal IIR), dynamic range reduction (look up table histogram)
2. Warp and resample
3. Contrast normalized
4. Fusion: Multiresolution Analysis (MRA) techniques; Laplacians pyramid or the Discrete Wavelet Transform (DWT) and the inverse transforms
5. Pattern selective fusion: decomposition of images into pattern elements, estimate salience, align and resample

3.2.1.2 Stabilization Processing. The real-time processing that stabilizes video from camera jitters is commonly based on motion estimation. The stages include:

1. Motion estimation feature-based or global intensity alignment algorithms (image registration problem). This stage uses the DWT.
2. Motion stabilization and image warping/motion smoothing to remove high-frequency fluctuation. This involves filter algorithms.
3. Image synthesis creating new frames to reduce abrupt motion between two consecutive frames.

3.2.1.3 Moving Object Tracking. The real-time processing stages for tracking moving objects include:

1. Moving region segmentation
2. Object detection by analyzing consecutive frames
3. Object tracking

3.2.1.4 Algorithm Computations on Hardware and Software Computations are distributed on hardware and software. The stages of the fusion, salience sensitive fusion, stabilization and moving object tracking algorithms are well suited for pipeline processing that uses custom hardware cores. The hardware cores can reduce the latency from the input video stream to the output video stream. Symbology, graphic overlay and pattern selective analysis are generated in software programs.

3.2.2 Task 2 – Architecture Models

The proposed FPGA implementation for fusion and stabilization is an architecture in which video data are streamed through custom hardware intellectual property (IP) cores via bus interconnections and Video Direct Memory Access (VDMA) hardware. The FPGA architecture consists of high-performance processors and their peripherals which are reconfigurable hardware IP cores. The processors run on an embedded operating system (e.g., Linux). The main program first initializes the bus interconnection/ VDMA and the peripherals. The main program then starts the peripherals and the data transfer. The processing (computation) is done by a pipeline of the IP cores. The main program also starts other software applications such as the graphic overlay and salience analysis.

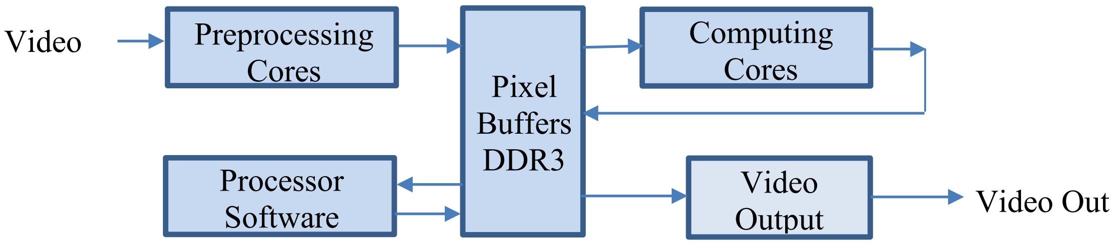


Figure 5. Video Stream Processing Data Flow

Figure 5 shows a data flow diagram for FPGA video processing architecture. The architecture utilizes pixel buffering for the pipeline signal processing algorithms. The architecture is similar to a pass-through architecture where a pipeline of cores process the pixels as they stream through the pipeline, however, the proposed architecture uses buffers for the processing stages.

The preprocessing cores in Fig. 2.1 convert the input video frame into stream of pixels. The preprocessing includes cores which perform non-uniformity correction (pixel gain and offset), pixel correction, BAYER (VNIR to YUV color), noise reduction (3x3 median filter and temporal IIR) and dynamic range reduction (look up table histogram).

The architecture buffers the pixels in fast access-time memory such as the DDR3 (Double Data Rate dynamic RAM) and transfers them to the computing cores. The processed pixels are buffered for the application software which further manipulates the frames with graphic overlay and salience analysis. The processed pixels are transferred to the video output controller generating the video output signal for display.

3.2.2.1 Latency Analysis. The goal is to process the information from the sensors to the display with minimal delay (latency). With information packet as frames, one-frame latency between video in and video out is equal to the reciprocal of the frame rate. For instance, the processing time for the 60Hz rate is to be less than 16.66ms, and for the 94Hz rate less than 10.4ms. The latency associated with the dataflow block diagram Figure 6 is the amount of time (in seconds) a video signal frame at the video-in port is transferred through preprocessing, the computing cores, the software application, the video output controller, and output at the video-out port.

Estimating the latency requires the number of clock cycles for transferring and processing, multiplied by the corresponding clock frequencies. This is specific to the architecture and implementation platform since different functional blocks have different clock frequencies. However, a rough estimate of the latency T is the pixel clock frequency Fvclk timed the number of pixels per frame N_{frame} plus the latency in the pipeline $T_{pipeline}$ (Equation 1 below).

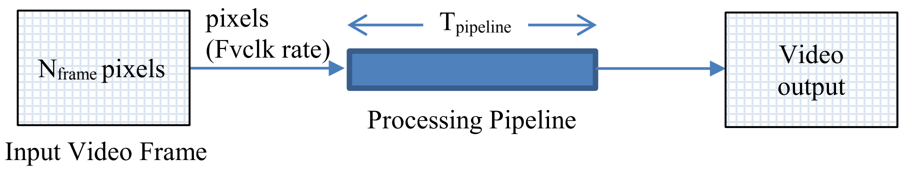


Figure 6. Rough Latency Estimate

$$T = Fvclk \times N_{frame} + T_{pipeline} \qquad (1)$$

For example, consider Fvclk = 148.5M pixel/second, (pixel clock frequency for 1080p/60Hz input video) and 1280 x 1024 pixel/frame, the latency T = 8.83ms + $T_{pipeline}$. For the 60Hz frame rate, the one-frame latency is 16.66ms, which implies that the $T_{pipeline}$ is to be less than 7.83ms. With the processing hardware clock rate Faclk = 150MHz, the 7.83ms is equal to 1,174,500 clock cycles.

Equation 1 shows that the latency depends largely on the pixel clock rate Fvclk. Therefore, matching the pixel rate Fvclk to the maximal processing hardware clock rate Faclk is essential for minimizing the latency. An Fvclk of 150MHz is currently the norm for Xilinx Zynq 7000 FPGA. However, an optimized design can attain 300MHz with Zynq 7000 and 500 MHz or higher with the latest FPGA technology such as Xilinx Ultrascale+ [17-20].

To minimize $T_{pipeline}$ the design approach is to minimize the wait times (pipeline stalls) between processing stages. The types of algorithms and calculations impact minimal pipeline stalls. For instance, the image sensor processing algorithms (pixel, color and gamma correction, and RGB-YUV conversion) can be straightforwardly done in a pipeline fashion with minimal or no pipeline stalls.

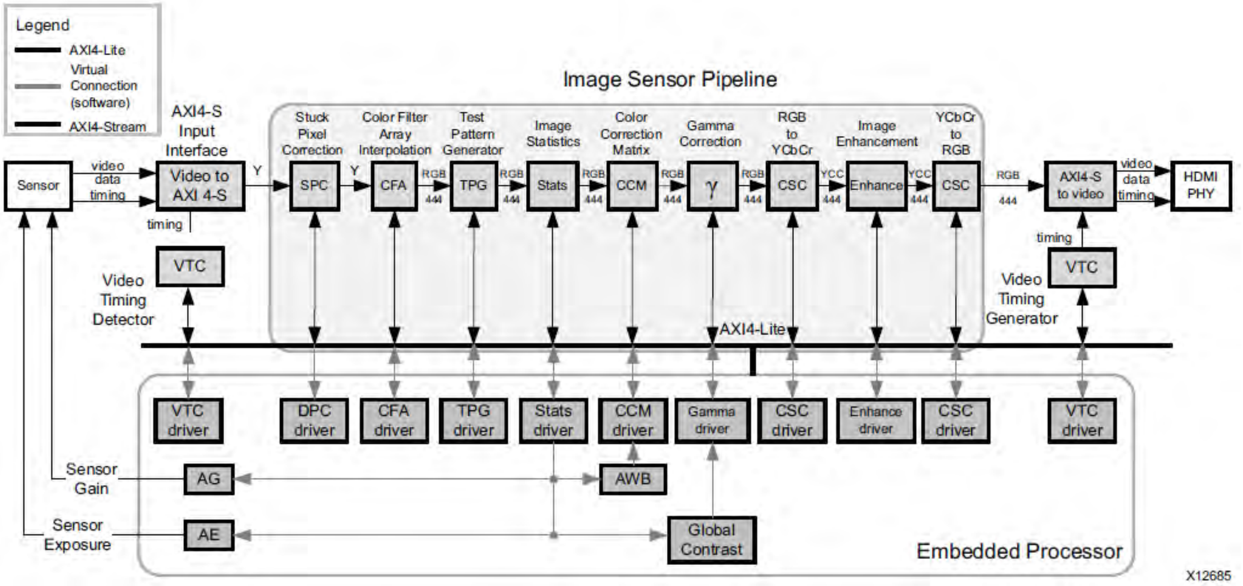


Figure 7. Example of Pipeline Architecture without Buffering [29]

Figure 7 shows an example of a pipeline architecture (from Xilinx Product Guide PG103 [29]) where pixels of video frames are streamed through a pipeline of IP cores. The architecture does not use buffering since it is not necessary for the processes. The input video data and timing transferred at the Fvclk rate from the sensor are converted to a stream Advance eXtensible Interface (AXI) bus signal using the Video to AXI4-S IP core [22]. The Video Timing Controller (VTC) IP core [23] detects the video timing signals which include the horizontal/vertical blanks and syncs, and the active video and active chroma signals. At the output of the pipeline the AXI4-S to Video core and VTC core converts and generates, respectively, the video output signal. In this design the embedded processor initializes and configures the hardware cores via an AXI-Lite bus for writing and reading of the core registers. The block diagram (Fig. 2.3) shows the software drivers for the cores and software applications.

3.2.2.2 Block Pipeline. A general architecture based on pixel buffering of a current frame must process and output the video within the allowed one-frame latency time. This might not be met even with a maximally possible pixel clock rate. An alternative approach is to increase the frame rate and utilizes block pipeline architecture. With a higher frame rate, the architecture can buffer and process the current frame while processing and outputting the previous frame, overlapping the processing of consecutive frames. The advantage is that with larger block of data in the buffers, the processing hardware can be optimized in terms of the clock frequency and the pipeline stalls. The architecture will incur more than one frame delay, for example, 1.5-frame latency however, with higher frame rate the latency in seconds is minimized.

3.2.2.3 Implementation Example. A reference implementation of FPGA video processor such as the Zynq 7000 AP SoC ZC702 Base Targeted Reference Design [24] can be used for better understanding the contributing factors to the latency (Eq. 1). A detailed description of the base reference design is as follows.

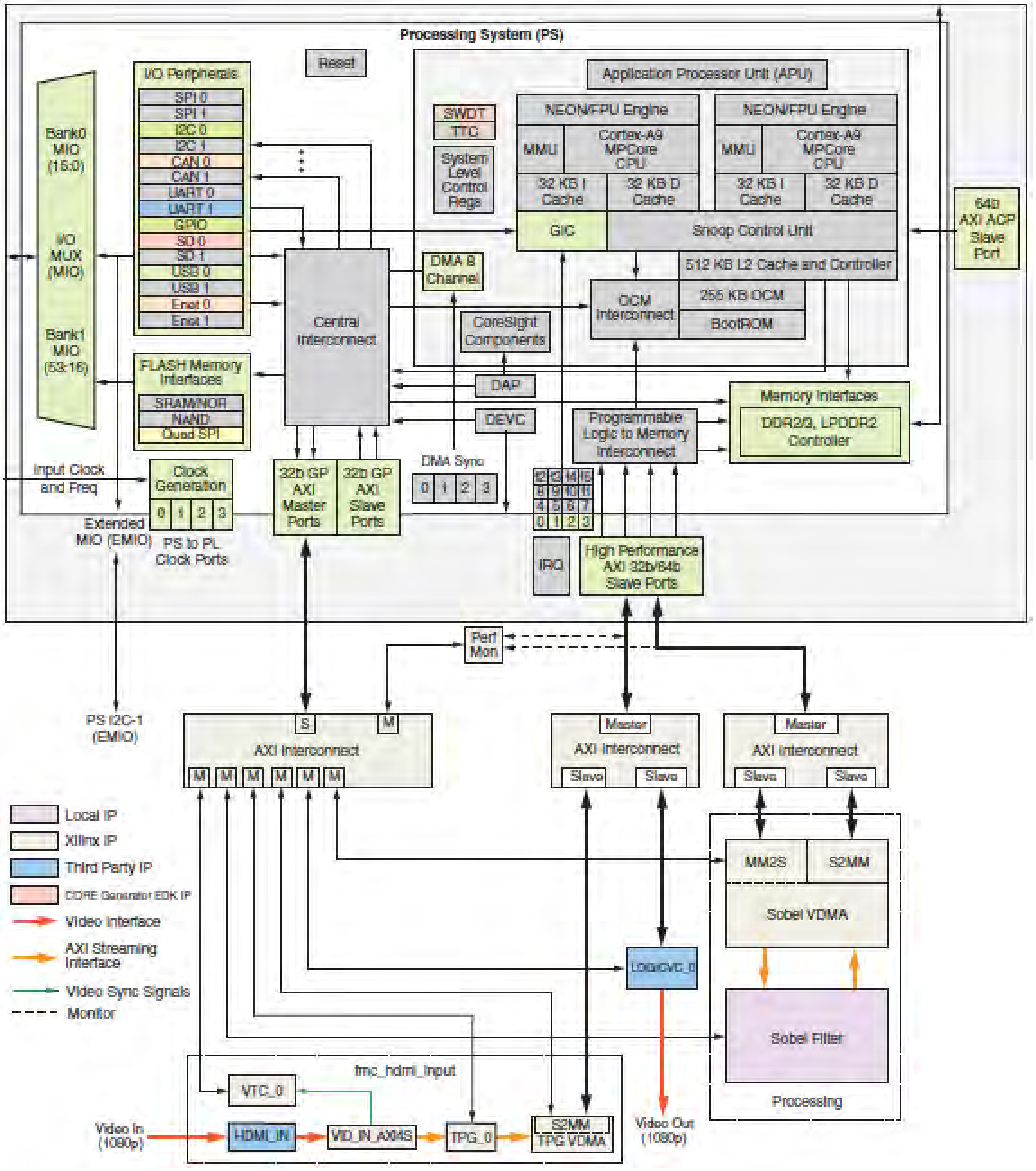


Figure 8. FPGA Video Stream Processing Architecture [24]

Figure 8 shows the architecture of the Zynq 7000 FPGA, which consists of the Processing System (PS) and the Programmable Logic (PL). The PS consists of computer system components:

i) Multiplex Input/output (MIO) and I/O Peripherals
ii) Application Processor Unit (APU) with two ARM Cortex-A9 CPUs

iii) Memory Interfaces
iv) Interconnect
v) Advance eXtensible Interface (AXI) bus Ports

The Programmable Logic (PL) is used for implementing the peripherals to the PS which include:

i) fmc_hdmi_input (the preprocessing cores in Fig. 2.1)
ii) processing (the computing cores in Fig. 2.1)
iii) LOGICVC_0 (Video Output Controller in Fig. 2.1)

The peripherals are connected to the PS via the AXI bus, which is represented in the design block diagram in Figure 8 by the three AXI Interconnect blocks. They are connected by the bold face and regular double-ended arrow lines. The application software running on the processors (CPUs) uses the AXI-Lite Interconnect (the AXI-interconnect block on the left hand side) for peripheral initialization and configuration. The middle AXI Interconnect is used for transferring the input frames from the fmc_hdmi_input to DDR3 memory, (the DDR3 is not shown in Figure)8 via High performance AXI slave ports and Memory Interfaces of the Processing System. The DDR3 memory is used for the pixel buffers. The middle AXI Interconnect is also used for transferring the processed pixels to the video output controller (LOGICVC_0). The right hand side AXI Interconnect is used for transferring the pixels to the processing cores. In this base reference design the core performs Sobel (edge detection) filtering.

The High-Performance AXI slave ports (bottom of the PS block) provide FIFO interfaces between the Faclk = 150 MHz, PL peripherals clock domain and the Fmclk = 533 MHz, DDR3 clock domain. The memory interfaces block in the PS provides the interfaces to the DDR memory. The PS General Interrupt Controller (GIC) handles the interrupt requests from the four master peripheral connections to the Fcpu = 667 MHz, ARM Cortex-A9 processors. The interrupts include the AXI_VDMA of the fmc_hdmi_input IPs block, AXI_VDMA of the processing IP block, the video display controller (logicvc block), the Sobel filter block, the HDMI_ IN block and the performance monitor unit.

The software application initializes the peripherals via the AXI master ports. The connections are the AXI-Lite interconnects indicated by regular size data paths. The AXI Streaming Interfaces are the orange boldface paths and the video frame data interfaces are the black boldface paths.

The ZC702 Base Targeted Reference Design development board has an FPGA Mezzanine Card - FMC not shown in Figure 8 providing HDMI input source. The fmc_hdmi_input block at the bottom of Figure 8 of the Programmable Logic (PL) comprises the HDMI_IN which converts HDMI to AXI stream data (orange path output of VID2AXI4S block). The Video Sync (green data interconnect) signal is an input to the Video Timing Control (VTC). The Test Pattern Generator (TPG) block generates internal AXI stream test video or passes through the external video. The Stream-to-Memory-Map (S2MM) VDMA [25] interfaces the stream data to memory-map peripheral of the PS. The application software configures the VDMA cores via the AXI-Lite. A performance monitor (Perf Mon) core is connected to AXI bus paths, and it is a slave peripheral to the processor.

In the proposed FPGA vision system processor, the processing block will consist of hardware cores computing algorithms such as the discrete wavelet transforms used in video fusion.

However, the bus interconnection architecture will be the same as the base reference design. The Stream-to-Memory-Map (S2MM) and Memory-Map-to-Stream (MM2S) of the VDMA are used for transferring data from the buffer to the computing cores via MM2S and streamed back to the buffer via S2MM.

The actual numbers on the bus transfer latency and rate, the memory access time, the latency associated with the preprocessing and computing cores and latency associated with the software application (e.g., graphic overlay) depend on the implementation of the algorithms. The latency associated with the hardware cores architecture does not have as significant an impact on the total latency as does the latency associated with the pixel clock rate and the pipeline stalls. The pipeline stall latency depends on the choice of the computation techniques.

3.2.2.4 Detailed Design. Estimating the latency between the input video frame and the processed frame output to the display requires a good understanding of practical FPGA architecture. The number of the register stages in the block is the latency associated with the block. Therefore understanding of register-level architecture of the cores is important. The clock frequency of the block and the number of register stages can be used to calculate the block latency in seconds. Different clock domains include the video pixel clock at Fvclk Hz, the Programmable Logic (PL) hardware at Faclk Hz, the DDR3 at Fmclk Hz and the Processing System (PS) at Fcpu Hz. For example in the base design the clock frequencies are 148.5MHz, 150Mhz, 533MHz and 667MHz, respectively.

The following discussion covers an example of FPGA architecture – Zynq 7000 AP SoC ZC702 base targeted reference design using the Xilnx Vivado Integrated Design Environment (IDE). The example will demonstrate that the latencies associated with different blocks in the architecture are negligible when compared to the pixel clock rate and the pipeline stalls.

The block diagram of the top level consists of the Processing System (PS) and the Programmable Logic (PL) (Fig. 2.5). The IDE synthesizes the block diagram of the system into a hardware description and generates the FPGA hardware configuration file called "bitstream" file. The bitstream is included the Linux boot file. The boot file includes the Linux kernel, the applications and the FPGA bitstream. The development board can be booted from an SD card storing the boot.bin.

Figure 9 below shows the Vivado IDE design block diagram consists of the PL components and the PS – the ZYNQ7 Processing System core.

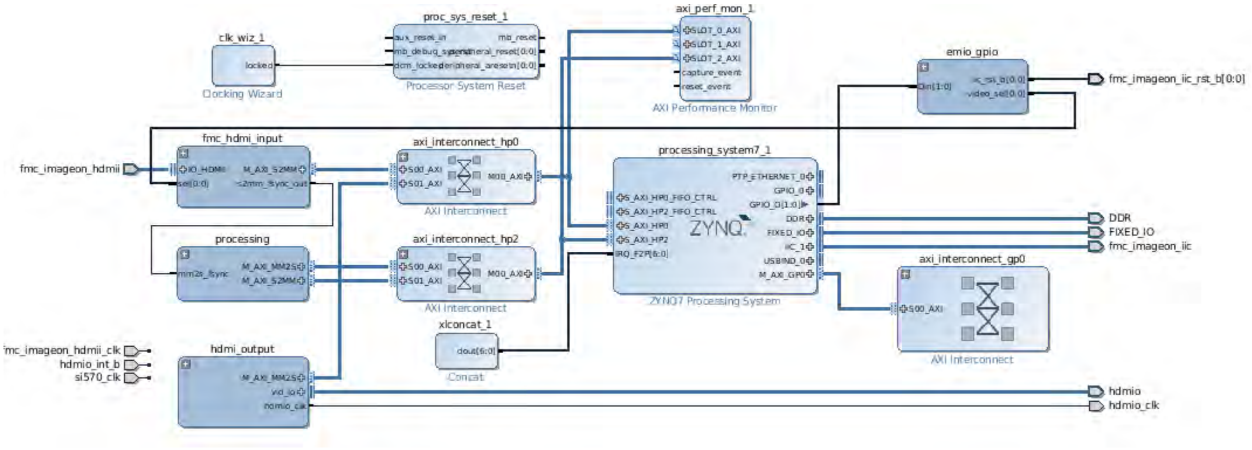


Figure 9. Vision Processing System Block Diagram

The video HDMI input block takes in the HDMI video stream at the fmc_imageon_hdmii port (left hand side of the diagram) and the HDMI out signal is at the hdmio port (right hand side of the diagram). The fmc_hdmi_input converts the HDMI format to AXI stream pixel data and buffers the pixels in the DDR3 via the axi_interconnect_hp0. The high-performance port HP0 and HP2 of ZYNQ7 interfaces the PL clock domain and the DDR3 clock domain. The pixels get processed by the processing block. The axi_interconnect_hp2 block transfers the pixels from DDR3 to the processing block and transfers the processed pixel back to the DDR3. Last, the axi_interconnect_hp0 transfers the processed pixels to the hdmi_output block (bottom left corner) which generates the HDMI output signal.

Figure 10 shows fmc_hdmi_input block. The HDMI input pixels are transferred at the pixel clock rate. Figure 11 shows a block diagram of the Video In to AXI4-Stream (Xilinx Product Guide). The diagram shows approximately 6 flip-flop stages latency.

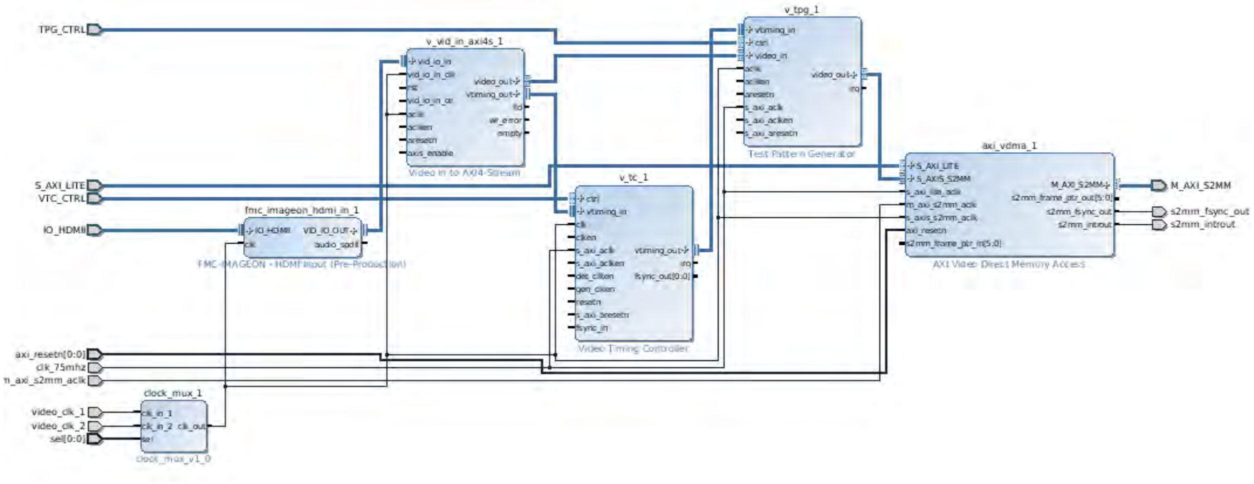


Figure 10. fmc_hdmi_input Block

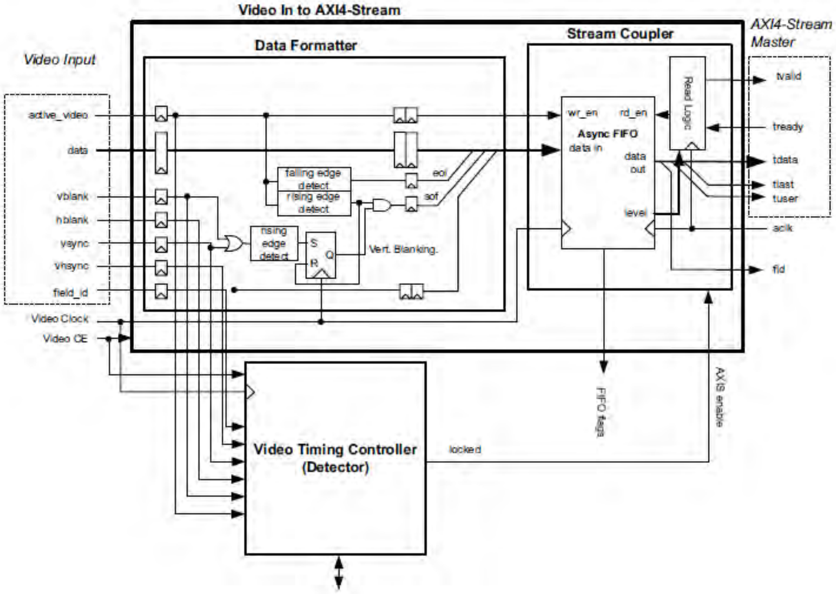


Figure 11. Video In to AXI Stream Block and Video Timing Controller [26]

The fmc_hdmi_input also includes the video Test Pattern Generator (v_tpg_1 block) which can generate internal test video or bypass the external pixels stream to the AXI Video Direct Memory Access core. The axi_vdma_1 core converts the AXI stream peripheral to the AXI4 (memory-mapped) peripheral. The AXI4 data (output of VDMA) are transferred to the DDR3 buffers via the AXI interconnect HP0 connects to the high performance port 0 of the processing system. The number of register stages in the fmc_hdmi_input is on the order of 100 cycles or less.

Figure 12 shows the axi_interconnect_hp0 which contains a crossbar switch and AXI Infrastructure cores (couplers) which includes data buffer and converters (clock, data width, protocol). The block connects the AXI4 peripheral input data through the s00_couplers block and routes to the HP0 port of the PS. The axi_interconnect_hp0 also connects the processed pixels from the DDR3 buffer via the PS HP0 port through the s01_couplers block to the hdmi_output core (bottom left).

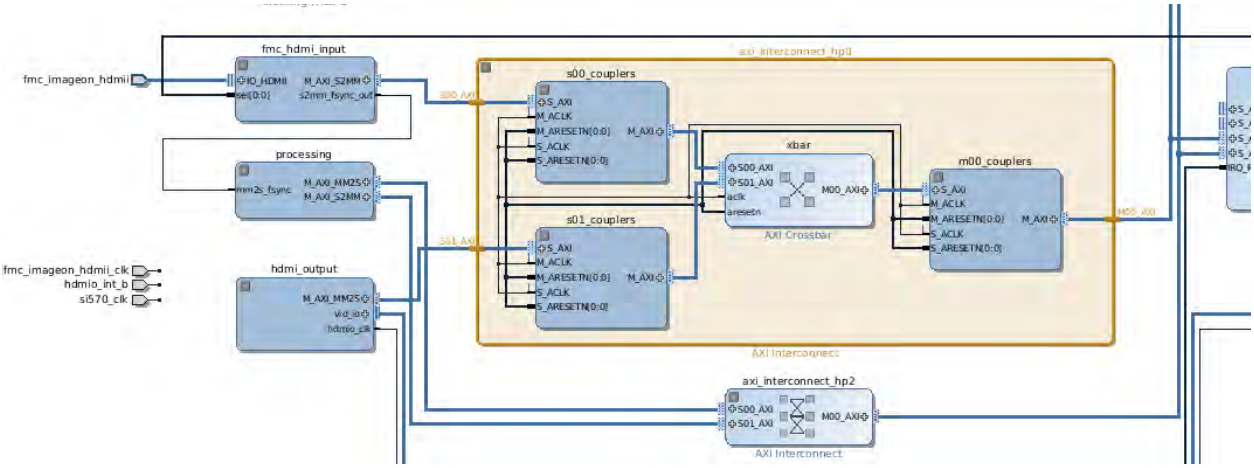


Figure 12. axi_interconnect_hp0 Block Diagram

Figure 13 shows the AXI Interconnect connecting the DDR3 buffer to the processing block. The latency associated with the AXI Interconnect cores is also on the order of less than 100 cycles.

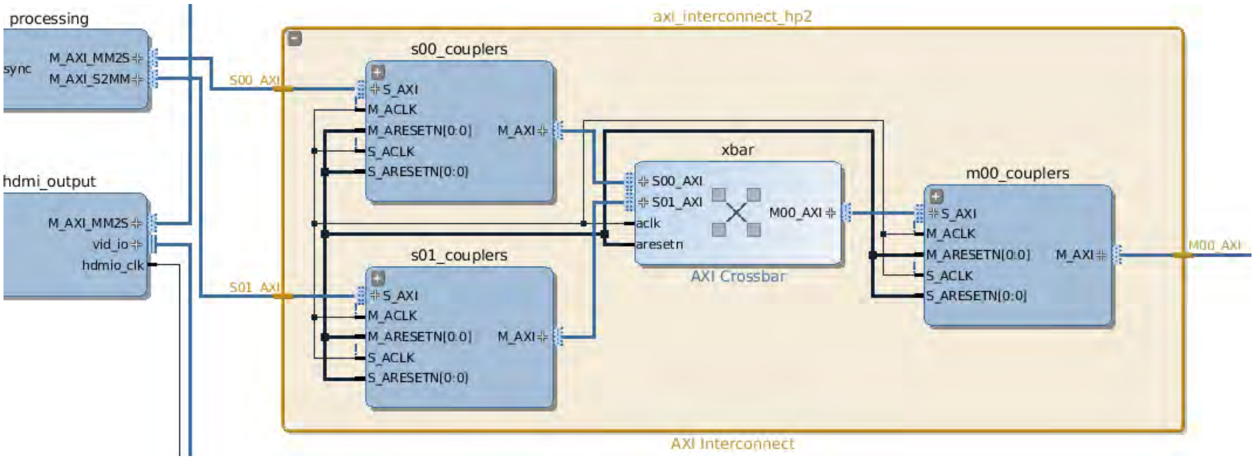


Figure 13. AXI Interconnect HP_2 Block Diagram

Figure 14 shows the processing block comprising the image_filter (Sobel edge detection), AXI4-Stream Register Slices and the AXI VDMA. The registers serve as the input and output registers to the filter. The VDMA converts the AXI4 input data to AXI stream and the AXI stream filter output data to AXI4 data.

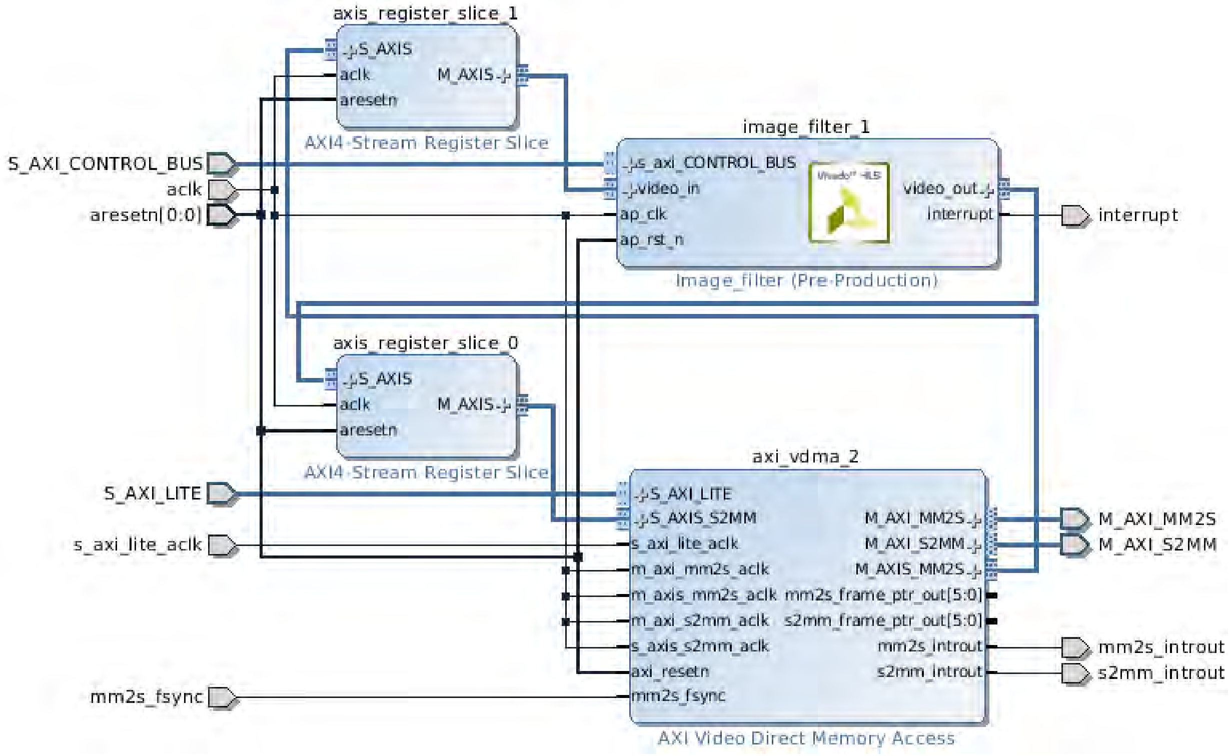


Figure 14. Processing Core Block Diagram

The latency associated with the processing core in the fusion and stabilization will be significantly higher.

Figure 15 shows the video output block which is the Multilayer Video Controller that generates the HDMI video output.

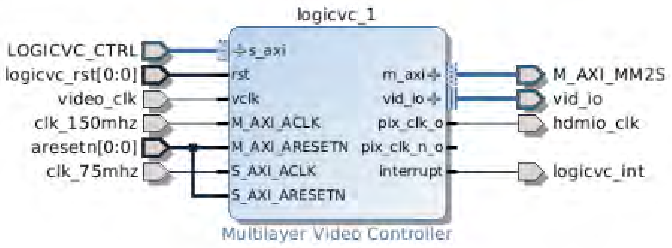


Figure 15. HDMI Output (Video Display Controller)

In the above detailed example of FPGA architecture, the latency due to the register stages of the cores is negligible compared to the latency due to the pipeline stall that can occur in the video fusion and stabilization. The choices of algorithms and computation can minimize pipeline stall.

The feasibility of designing hardware cores with minimal pipeline stall was demonstrated by the SRI Acadia II on which video fusion, stabilization and moving object tracking was performed.

In summary, Task 2 covered latency analysis of the proposed FPGA-based vision processor and showed that pipeline architecture comprising custom Intellectual Property (IP) cores can meet the latency requirement. The IP cores compute the algorithms in a pipeline structure where the latency is equal to the time (number of clock cycles) it takes a video frame to pass through plus the latency due to the register stages and the pipeline stalls specific to the algorithm data dependency. The pass through latency depends on the pixel transfer clock rate. The latency due to the register stages in the architecture components is negligible.

3.2.3 Task 3 Projected Performance Analysis of FPGA-based Vision Processor

In Task 3 the analysis now turns to the potential pipeline stall latency on the algorithms.

3.2.3.1 Algorithms Latency Analysis. Standard algorithms in vision processing can be used as benchmarks in projecting the latency. Vision processing algorithms are typically dense linear algebra calculations. These calculations are well suited for pipeline structure. The analysis for this project will be based on the Gaussian pyramid and Laplacian pyramid [27, 28] calculation in applications such as video fusion [7-11]. For instance, in night vision applications, fusion of three video sources – LWIR (Longwave Infrared), SWIR (Shortwave Infrared) and VNIR (Visible Near Infrared) can be employed using the Laplacian pyramids for the three sources [7, 9]. One of the pixel-based fusion-rules involves selecting the highest values among the three Laplacians to form a fused Laplacian pyramid, which is used in the construction of the video frame.

3.2.3.1.1 Gaussian and Laplacian Pyramids Latency Analysis. The Gaussian pyramid is an iteration of down sampling using a convolution filter. The filter process involves the convolution of the frame and the kernel (shown below).

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

The down sampling involves deleting even rows and columns (kernel image source opencv pyrDown() [21]). The hardware core for calculating the filter must provide an optimal latency and rate. The frame pixels are transferred via AXI bus in a serial fashion to AXI-stream convolution filter hardware.

3.2.3.1.2 Low-latency Convolution Filter.

The convolutional filter calculates its output, P(i, j), i = 1, …, h and j = 1, … w (h is the frame height and w is the width) as the weighted sum of the kernel elements and the pixels in the window centered at i, j. In the case of the opencv pyrDown() the 5x5 kernel, new P(i, j) ← 36P(i,j) + P(i-2, j-2) + 4P(i-2, j-1) + … + P(i+2, j+2).

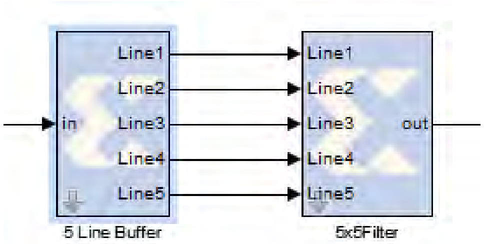


Figure 16. Convolution Filter with 5-line Buffer

A version of hardware IP core for the convolution filter has as its input 5 rows of the frame pixels where a filtered pixel is processed every clock cycle (optimal pipeline rate). Figure 16 shows a schematic of a 5x5 filter, which receives the inputs from the 5-line buffer. The schematic is a MATLAB/Simulink design model of the Xilinx System Generator DSP tool. The design serves as an analysis of an optimal rate and latency convolution filter design. The 5x5 filter on Xilinx FPGA takes image pixels from MATLAB tool and the filtered image is displayed with MATLAB.

Figure 17 shows the 5-line buffer where the width, w, of the frame (the number of columns, e.g., 1280 of the 1280x1024 resolution) is equal to 90 pixels in this example.

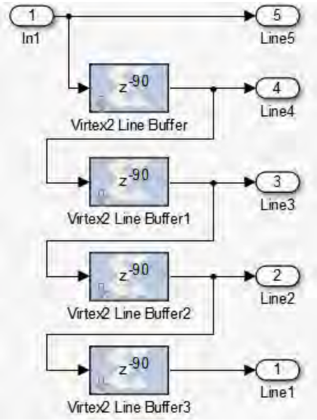


Figure 17. Line Buffer – 5 lines Buffer with 90-pixel Depth

Figure 18 shows a block diagram of the 5x5 filter. It uses five 5-tap MAC FIR (Multiply and Accumulate Finite Impulse Response) units to calculate the row-wise weighted sums of the kernel and adders to sum the MAC FIR outputs.

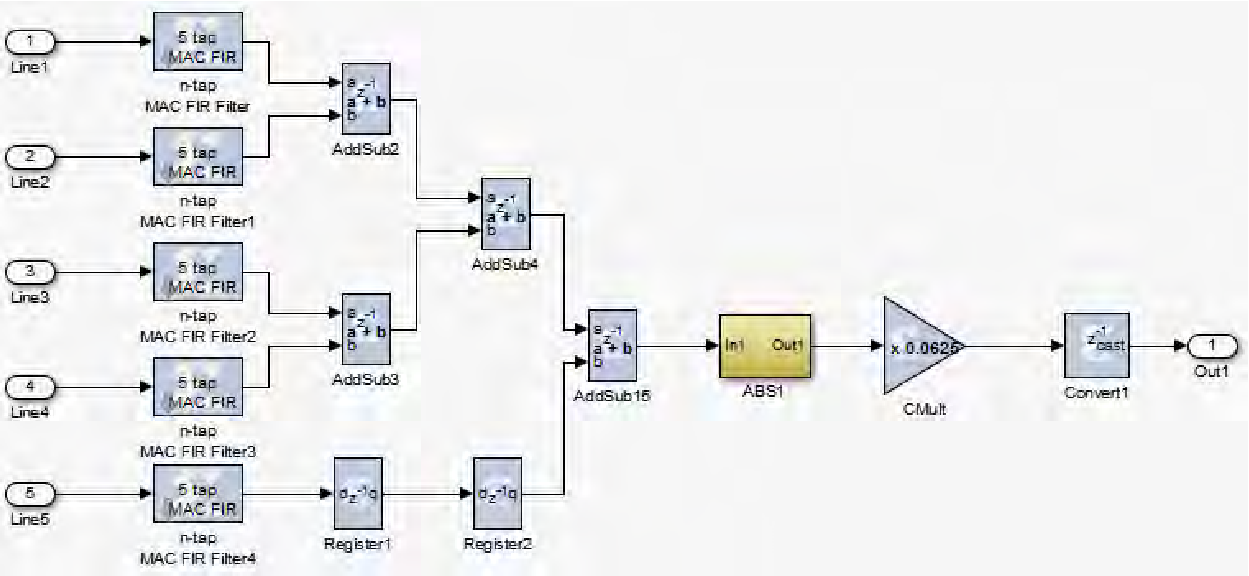


Figure 18. 5x5 Convolution Filter

The latency associated with the filter is on the order of 4w (width). For 1280x1024 frames the latency is 5,129 clock cycles. The additional 9 cycles are due to the MAC FIR filters and the adders in Figure 18.

3.2.3.1.3 Gaussian Pyramid. The series of frames

$$G_{l+1} = \text{pyrDown}(G_l),\ l = 0, \ldots, n; \tag{3.1}$$

where the pyrDown (opencv operation for pyramid down sample) is the down sample of G_{l-1} and l is called a level. For example with n = 5, the Gaussian pyramid has the input frame G_0 is 1280x1024, G_1 is 640x512, G_2 is 320x256, G_3 is 160x128, G_4 is 80x64 and G_5 is 40x32. The down sample, pyrDown operation performs the convolution filter on a frame and rejects even rows and even columns. The pipeline hardware core that performs the Gaussian pyramid can be a cascade of the line-buffers and convolution filters in Figure 16 and control logic for rejecting the even rows and columns.

3.2.3.1.4 Laplacian Pyramid. An up sample pyrUp(Gl+1) is done by injecting Gl+1 with zero even rows and columns expanding it to the same size as Gl, and performing the convolution filter with the same kernel as pyrDown but multiplied by 4. The errors between the pixels of Gl and the interpolated Gl+1, pyrUp(Gl+1) is the Laplacian at the level l. A set {Ll, l = 0, …, n-1} is the Laplacian pyramid defined by

$$L_l = G_l - \text{pyrUp}(G_{l+1}),\ l = 0, \ldots, n\text{-}1; \tag{3.2}$$

A pipeline hardware core can consist of a cascade of the line-buffer and convolution filter pairs (Fig 3.1), control logic for injecting even zero rows and columns and addition/subtraction units to obtain the pixels of L_l, l = 0, …, n-1.

Figure 19 shows a pipeline Laplacian pyramid with n = 4. The input frame is G_0. The pyrDown blocks and pyrUp blocks are both the line-buffer/5x5 FIR-filter hardware cores.

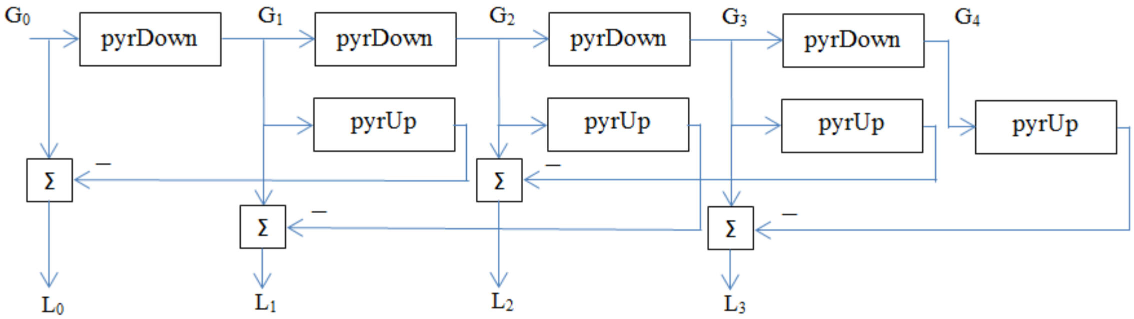


Figure 19. Pipeline Laplacian Block Diagram

The pipeline latency associated with the Laplacian pyramid hardware $\{L_l , l = 0, 1, .., n\text{-}1\}$ is on the order of 4w (frame width w and 5x5 kernel). This is due to the pipeline stall in the first stage line-buffer. The optimal pipelining rate is one pixel per cycle.

3.2.3.2 Fusion. Figure 20 shows the Laplacian pyramid fusion block diagram. The hardware is a pipeline structure where the input pixels from three video sources are serially transferred from the input ports on the AXI-stream buses. The output of the Laplacian pyramid blocks are pipelined into the fusion block. The three Laplacian pyramid sources are fused into a single Laplacian pyramid $\{L_0, L_1, \dots , L_n\}$. The fusion rule is pixel based, for example, by selecting the largest pixel values from the three pyramid sources.

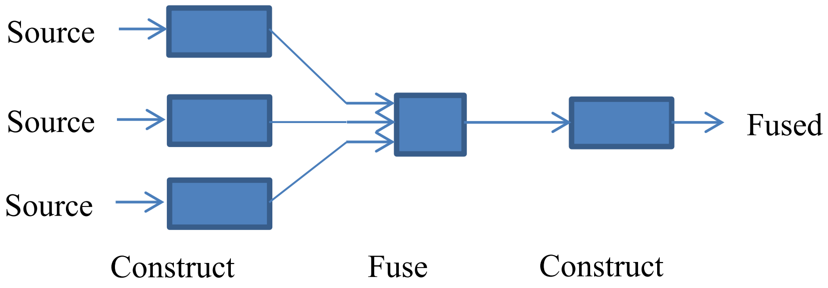


Figure 20. Multiple Sources Fusion

To construct the frame, the process iterates in the inverse fashion using pryUp operation,

$$G_l = \text{pyrUp}(G_{l+1}) + L_l, l = n\text{-}1, \dots, 0; \qquad (3.3)$$

where $G_n = L_n$ and G_0 is the constructed frame.

A pipeline stall occurs at this stage since the iteration begins l = n-1 down to 0. The first stage line-buffer for the pixels of G_n (L_n) will incur $4w/2^n$ cycles before pyrUp(G_n) starts.

3.2.3.3 Wavelet Transform. The Discrete Wavelet Transform (DWT) is another ubiquitous multiresolution technique for fusion, stabilization and motion analysis [7, 10, 11]. It consists of low-pass filtering, high-pass filtering and down sampling the rows and columns of (M x N) frame (M rows and N columns). The two-dimensional DWT low-pass filters and high-pass filters and then down samples the rows into two (M/2 x N) frames – I_L and I_H. Then it filters and down samples the columns of I_L into two (M/2 x N/2) frames – I_{LL}^1 and I_{LH}^1, and similarly, the columns of I_H into two (M/2 x N/2) frames – I_{HL}^1 and I_{HH}^1, where the super script 1 indicates the level 1. For the level 2 (k = 2) the DWT repeats the process on I_{LL}^1 which generates four more sets of (M/4 x N/4) coefficients I_{LL}^2, I_{LH}^2, I_{HL}^2, I_{HH}^2, and from the level 1 – I_{LH}^1, I_{HL}^1, I_{HH}^1. The DWT with k decomposition levels has 3k + 1 sets of coefficients.

The wavelet-based fusion uses different fusion rules for different decomposition levels to combine multiple DWT coefficients from multiple sources. The fused image construction process inverse transforms (IDWT) the combined coefficients.

Pipeline hardware cores for wavelet-base fusion will have a similar structure to the Laplacian pyramid fusion. The latency associated with the pipeline is minimal since the low-pass filter and the high-pass filter are well-suited for pipeline structure.

3.2.4 Graphic Processing Unit (GPU) v. Field Programmable Gate Array Custom Hardware for Real-Time Multiresolution Analysis.

This section is a narrative of the comparison between using Graphic Processing Unit (GPU) versus FPGA for the multiresolution algorithms such the Laplacian pyramid and the discrete wavelet transform. The discussion includes the technologies for embedded vision processors and the performance, programmability and ease in development of them.

3.2.4.1 GPU in Embedded and Mobile Platform A Graphic Processing Unit (GPU) for embedded and mobile platforms is a massively parallel processor, which comprises Floating-Point Unit (FPU) cores, distributed local memory and caches. For example, the NIVDIA Tegra K1 [30] mobile processor consists of the Kepler SMX (Stream Multiprocessor) architecture (1GHz) GPU [31] and quad-core ARM A15 processor (2.3GHz) CPU. Figure 21 (image source from Kepler Architecture whitepaper [31]) shows the architecture with 129 CUDA cores (each with single-precision FPU and fixed-point arithmetical unit), 64 Double Precision (DP) units, 32 Special Function Units (SFU), 32 Load/Store units, instruction and data caches, register file and the hardware texture units.

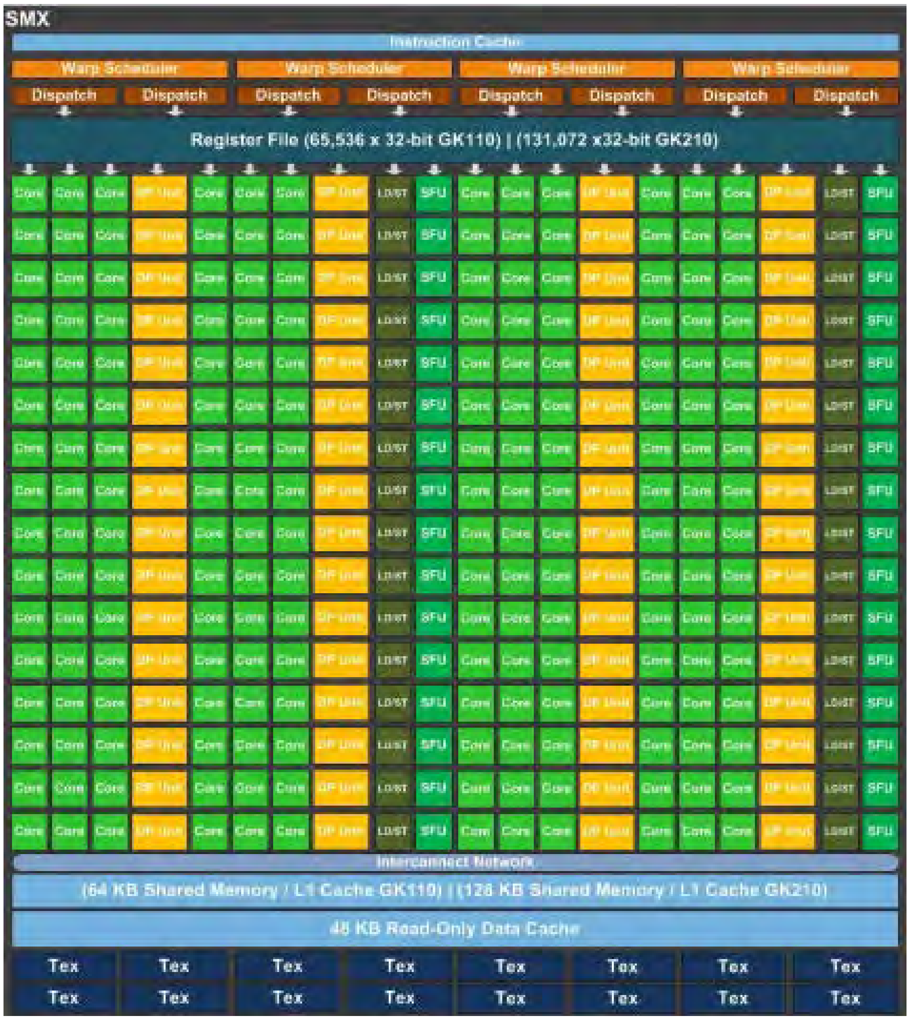


Figure 21. Kepler Architecture [31]

The GPU/CPU embedded processors such as the Tegra K1 are specific to graphic computations on mobile devices and are a suitable platform technology for the vision embedded processor.

High-performance scientific computing research projects have leveraged GPU massively parallel floating-point unit cores architecture to achieve desktop supercomputing performance. In 2009, Nagvajara, et al., compared the performance of the GPU, PlayStation 3 (PS3 Cell processor) and CPU to compute the Fast Fourier Transform (FFT) for the phase reconstruction algorithm, which requires iteration of FFT and the inverse FFT [32]. The results showed that the performance (computation time) gained from the parallel FPUs in GPU and PS3 Cell processor was not as expected due to data transfer bottleneck between the cores, especially when the Intel Math library FFT on CPU could provide equal performance. However, since 2009, GPU performance for FFT calculations has improved. Most recently, deep neural networks (deep learning) research has used GPU to perform accelerated two-dimensional FFT [33]. For instance, driver-assisted automobiles use the Tegra X1 GPU [34].

In terms of application development, the GPU has the OpenCV (Open source Computer Vision) library support. The OpenCV GPU module is a set of classes and functions for accelerated processing with the NVDIA GPUs [35]. There are functions for image processing and image filtering for example the Laplacian pyramid.

Image fusion using GPU has been reported. Strengert et. al. [36] used GPU to calculate the pyramid methods used in biquadratic B-spline filtering zooming, blurring, and scattered pixel data interpolation. The authors reported a data transfer bottleneck between the main memory and the GPU CUDA cores. Lu, et.al. [37] used GPU to perform wavelet-based fusion of remote sensing images. They reported using the render function to texture (hardware texture units) technology to speed up transform calculations over CPU for large image sizes 5-fold for1024x1024 and 10-fold for 2048x2048.

3.2.4.2 FPGA Advantages over GPU. Using the GPU stream processors and texture units available in GPU architecture to achieve low-latency performance is far from straightforward as GPU vendors suggest. GPU is customized for accelerating graphic calculations and offers advantages for floating-point data algorithms. However, the video processing algorithms considered in this project calculate fixed-point pixel data. The floating-point units in GPU will have higher power consumption than the FPGA fixed-point hardware.

3.2.4.2.1 Preprocessing Hardware Cores (in FPGA but Not in GPU). With regard to the video preprocessing frontend of the vision processor, which includes – non-uniformity correction (pixel gain and offset), pixel correction, BAYER (VNIR to YUV color), noise reduction and dynamic range reduction, it is best to use FPGA. The preprocessing would be a challenge using GPU since it would have to be done with software.

3.2.4.2.2 OpenCV Available for High-Level Synthesis (HLS) in FPGA Custom Hardware Design.

FPGA hardware cores make use of the OpenCV image and signal processing and mathematics modules [38]. Xilinx High Level Synthesis (HLS) offers an effective methodology for developing the hardware IP cores for signal processing computation. High-level language such as C efficiently describes signal processing algorithms. There are libraries and open source codes (e.g., the open source computer vision – the OpenCV project) for developers to reuse in coding the algorithms. In the HLS design paradigm, the hardware algorithms are described in C, C++ or System C codes. The HLS design tool synthesizes algorithms to optimized IP cores described as Hardware Description Language (HDL) codes. The optimization is typically for minimal latency and throughput or minimal hardware resources (silicon area) or minimal power consumption. The HLS method optimizes design using techniques such as loop unrolling, array data structure synthesized to memory (RAM) and First-in First-out (FIFO) queues and pipeline hardware by means of specialized HLS directives in high-level language descriptions.

The HLS tools can aid rapid delivery of prototypes, particularly, the advanced signal processing algorithms used in video fusion, stabilization and object tracking. The HLS method can provide design specification, description, library codes, verification and test bench.

3.2.4.2.3 Xilinx UltraScale+. The FPGA technology is continuing to improve with next generation chips such as the UltraScale+ devices [17-20]. These FPGAs can provide the required low-latency vision processing and low-power consumption (performance per watt). The proposed LLEVS performs real-time video processing that demands pipeline hardware and efficient memory hierarchy to deliver low latency performance.

4.0 RESULTS AND DISCUSSION

In this section the results of the analysis for both the Low Risk and High Risk Approaches are presented and discussed.

4.1 Low Risk Approach

The Acadia II test results are located in Table 6. The data were collected using an Acadia II development board. The Acadia II is not capable of supporting 2560 x 2048 sensors, so those resolutions have not been included. To support two independent outputs with a resolution of 1280 x 1024 and a frame rate of 60 Hz requires two Acadia chips and correspondingly higher power. The number of flip flops and logic cells used does not directly apply to the Acadia as it is an ASIC.

The results in Table 6 show that the Acadia is capable of supporting a HMD system with a 640 x 480 by 14 bits @ 60 Hz camera with a single 1280 by 1024 8 bit @ 60 Hz display output within the required power and latency requirements. The Acadia II will also support a 1280 x 1024 by 14 bits @ 60 Hz camera with a single 1280 by 1024 8 bit @ 60 Hz display output within the required power and latency requirements. If two separate display out puts at 1280 by 1024 8 bit @ 60 Hz are required, two Acadias would be needed.

Table 6. Acadia II Test Results

CAMERA RESOLUTION & FRAME RATE	ACADIA II		
	1 Camera Pass Thru	2 Camera Pass Thru	2 Camera Fusion
640 x 480 by 14 bits @ 60 Hz			
Latency (In Frames/msec)	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec
Power Consumption (Watts)	1.9	2.1	2.2
• Dynamic	1	1.2	1.3
• Static	0.9	0.9	0.9
CPU Usage (# of CPUs used)	1	1	1
RAM (MB)	1.8	3.5	5.3
Flip Flops (#)	N/A	N/A	N/A
Logic Cells (#)	N/A	N/A	N/A
1280 x 1024 by 14 bits @ 60 Hz			
Latency (In Frames/msec)	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec
Power Consumption (Watts)	2.3	2.6	3.5
• Dynamic	1.4	1.7	2.6
• Static	0.9	0.9	0.9
CPU Usage (# of CPUs used)	1	1	1
RAM (MB)	2.9	5.8	8.0
Flip Flops (#)	N/A	N/A	N/A

The results from the analysis for the Xilinx MPSoCs are located in Table 7. The results show the Zynq 7000, 7Z045, 7Z100 and ZU9EG have sufficient processing power and flip flops to meet the threshold and objective requirements of less than one frame latency. The real issue is when there are two 2560 x 2048 by 14 bits @ 60 Hz cameras. At this point only the UltraScale+ ZU9EG or other family member can meet the power requirement of 6 watts or less. The analysis that determined this power consumption figure of 6 watts is at the end of this section.

For two 2560 x 2048 by 14 bits @ 96 Hz cameras neither the Zynq Series 7000 7Z100 nor the UltraScale+ ZU9EG can meet the power consumption requirement or have sufficient onboard memory. The objective power may be achieved by adapting the device configuration for the target implementation, controlling the active power planes available on the MPSoC device or fine tuning the algorithms for optimized power consumption, such as minimizing the memory access cycles. Memory function may be enhanced by employing a higher functioning family member of the MPSoC devices or adapting one of the advanced capability algorithms being pursued under the high risk effort of the project.

The two cells in Table 7 highlighted in red indicate that memory resources not in the device will be required in the design. The Zynq 7000 7Z100 would require an additional 23% of memory to meet the objective goal. This is the reason the UltraScale+ device was included as the best current candidate. The 1% memory shortage estimate during the simulation analysis has the potential for being solved by adjusting the method for onboard memory usage or going to UltraScale+ devices with more memory when they become available.

Table 7. Power and Latency Estimates

CAMERA RESOLUTION & FRAME RATE	Xilinx Zynq 7000			Xilinx Zynq UltraScale+		
	1 Camera Pass Thru	1 Camera Pass Thru	2 Camera Fusion	1 Camera Pass Thru	1 Camera Pass Thru	2 Camera Fusion
640 x 480 by 14 bits @ 60 Hz	7Z020		7Z030	ZU9EG		
Latency (In Frames/msec)	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec
Power Consumption (Watts)	1.74	1.83	2.58	2.035	2.135	2.401
• Dynamic	1.479	1.562	2.184	1.167	1.259	1.459
• Static	0.261	0.268	0.396	0.868	0.876	0.942
CPU Usage (# of CPUs used)	1	1	1	1	1	1
RAM (MB/%)	1.8	3.5	5.3/56	1.8	3.5	5.3/16
Flip Flops (#/%)	21000	28000	60000/38	21000	28000	60000/11
1280 x 1024 by 14 bits @ 60 Hz	7Z030		7Z030	ZU9EG		
Latency (In Frames/msec)	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec
Power Consumption (Watts)	2.516	2.71	3.59	2.235	2.335	2.97
• Dynamic	2.199	2.382	3.164	1.364	1.459	2.018
• Static	0.317	0.328	0.426	0.871	0.876	0.952
CPU Usage (# of CPUs used)	1	1	1	1	1	1
RAM (MB/%)	2.9	5.8	8.0/84	2.9	5.8	8.0/24
Flip Flops (#)	21000	28000	60000/38	21000	28000	60000/11

Table 8. Power and Latency Estimates (concluded)

CAMERA RESOLUTION & FRAME RATE	Xilinx Zynq 7000			Xilinx Zynq UltraScale+		
	1 Camera Pass Thru	1 Camera Pass Thru	2 Camera Fusion	1 Camera Pass Thru	1 Camera Pass Thru	2 Camera Fusion
2560 x 2048 by 14 bits @ 60 Hz	7Z045		7Z100	ZU9EG		
Latency (In Frames/msec)	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec	< 1 Frame/ <16 msec
Power Consumption (Watts)	4.073	4.59	6.43	3.67	3.98	5.27
• Dynamic	3.456	3.974	5.674	2.677	2.968	4.228
• Static	0.617	0.616	0.756	0.993	1.012	1.042
CPU Usage (# of CPUs used)	1	1	1	1	1	1
RAM (MB/%)	5.9	11.7	22.0/82	5.9	11.7	22.0/68
Flip Flops (#/%)	41000	55000	127000/23	41000	55000	127000/23
2560 x 2048 by 14 bits @ 96 Hz	7Z045		7Z100	ZU9EG		
Latency (In Frames/msec)	< 1 Frame/ <10 msec	< 1 Frame/ <10 msec	< 1 Frame/ <10 msec	< 1 Frame/ <10 msec	< 1 Frame/ <10 msec	< 1 Frame/ <10 msec
Power Consumption (Watts)	4.562	5.51	8.07	4.47	4.97	6.47
• Dynamic	3.945	4.844	7.214	3.461	3.938	5.388
• Static	0.617	0.666	0.856	1.009	1.032	1.082
CPU Usage (# of CPUs used)	1	1	1	1	1	1
RAM (MB/%)	8.8	17.5	33/123	8.8	17.5	33/101
Flip Flops (#)	69000	90000	185000/34	69000	90000	185000/35

4.1.1 Size & Weight Comparison.

Table 8 shows a comparison of the package sizes of the Acadia and an average Zynq 7000 and UltraScale device. The final measurements will depend on the device selection.

Table 9. Processor Physical Characteristics

	Acadia II	Zynq 7000 Series Example*	UltraScale+ Example*
Length (mm)	29	31	35
Width (mm)	29	31	35
Height (mm)	2.938	3.15	3.45
Weight-incl PCB (gms)	26	27	29

*Note: Size and weight will vary depending on the part selected.

The mounting areas required by the Zynq 7000 and UltraScale+ are larger than the area required by the Acadia II. Unit size will be one of the factors considered when choosing a MPSoC component. With the additional functionality of the MPSoC some of the peripheral circuitry needed to support the Acadia II will no longer be needed providing more room for the chip.

4.1.2 Power Requirements.

Power consumption is a major issue for mobile systems especially with head and helmet mounted systems. From a practical viewpoint the need for batteries to power a system impacts the size and weight of what must be mounted on an operator’s helmet and further contributes to the battery load that the operator must tote on a mission. There is also the added concern of how much heat must be dissipated by the system as it affects user comfort and may contribute to the user’s heat signature on thermal sensors.

The system power budget is set at less than 10W which leaves the processor at an allocation of between five and six watts. For a DEVS type of system the power breakdown would be roughly as follows:

SWIR camera (TECless)	2.1W
LWIR camera	1.2W
Displays	.6W
Support Electronics	.7W
Total Wattage Less Processor	4.6W
Processor Allocation	5.4W

The processor allocation of 5.4W is within striking distance of the 6.47W estimated for the UltraScale+ ZU9EG under objective conditions.

4.2 High-Risk Approach

4.2.1 Task 1.

The results from Task 1 on a literature survey of the applications in video fusion, salience sensitive fusion, stabilization and moving object tracking, show that the algorithms used in these applications are well suited for pipeline processing implemented on custom hardware cores. The hardware cores reduce the latency from the input video stream to the output video stream. FPGA has as its components embedded processors that run on an embedded Linux operating system. These embedded processors are suitable for running symbology, graphic overlay and pattern selective analysis programs.

4.2.2 Task 2.

The latency calculation $T = Fvclk \times N_{frame} + T_{pipeline}$ (Equation 1) in which the transfer time $Fvclk \times N_{frame}$ depends on the pixel clock rate dominates the pipeline latency $T_{pipeline}$. In the detailed example of FPGA architecture covered in Task 2, the latency due to the register stages of the cores is negligible compared to the latency due to the pipeline stall that can occur in the video fusion and stabilization. The choices of algorithms and computation can minimize pipeline stall. The feasibility of designing hardware cores with minimal pipeline stall was demonstrated by the SRI Acadia II on which video fusion, stabilization and tracking was performed. Recall the case Fvclk = 148.5M pixel/second and 1280 x 1024 pixel/frame, the latency $T = 8.83ms + T_{pipeline}$. The one-frame latency constraint is 16.66ms and $T_{pipeline}$ is to be less than 7.83ms. With the processing hardware clock rate equal to 150MHz, the 7.83ms is 1,174,500 clock cycles. The latency due to the register stages in the architecture components is negligible. With the pixel clock frequency equal to the hardware clock frequency at 500MHz, FPGA architecture offers a viable platform for a low-latency embedded vision system.

4.2.3 Task 3.

The latency analysis model is based on the calculation as the number of clock cycles to pass through the frame pixels plus the pipeline latency, which includes the register stages and stalls (buffers), that is, $T = (N_{frame} / Fvclk) + T_{pipeline}$, where the first term is the pass through latency – number of pixels divided by pixel clock rate, and the second term is the pipeline latency. For the Laplacian pyramid hardware, the pipeline needs to buffer 4 rows of the image using the line-buffer before the 5x5 Gaussian filter can start. This results in an approximately 5,120 cycle pipeline stall (for 1280x1024 pixels). The analysis presented in Task 3 is that the pipeline stalls will not present a problem for FPGA as a viable technology.

In Task 3, results from the project literature survey comparing the Graphic Processing Units (GPUs) v. FPGA shows that FPGA is the technology of choice because it offers custom design for the vision processor. As the SRI Acadia II is a custom Application Specific Integrated Circuit (ASIC) for vision processor and the GPU is custom ASIC for graphic processor, FPGA offers a reconfigurable vision processor where upgrading of algorithms can be on both hardware and software cores.

5.0 CONCLUSION

5.1 Low Risk Approach

The purpose of the Low Risk approach is to generate an architecture for currently available image processing devices capable of processing up to a 5 Megapixel video image stream at 96 frames a second with less than one frame of latency for a helmet mounted imaging system.

The analysis completed in the previous section shows the Acadia II, in the fusion mode, can meet the threshold level for two cameras but does not have sufficient output capability to feed separate data to two displays. The Zynq Series 7000 7Z030, in the fusion mode, can support two cameras and two displays at the threshold level. The Zynq Series 7000 7Z100, in the fusion mode, meets the requirements of the 2560 x 2048 by 14 bits @ 60Hz including latency, but does not meet the power requirement needing 20% more power than is in the power budget of 5.5W described in the previous section.
The UltraScale+ ZU9EG, it can meet all of the threshold levels of the requirements and also for the higher resolution of 2560 x 2048 by 14 bits @ 60Hz. The ZU9EG falls short for the same resolution at 96 Hz, the objective level. The onboard memory falls short by 1% and the power is 20% greater than the amount allotted in the power budget but is able to meet the latency level of less than one frame. Neither of these short falls of the objective requirement is sufficiently large to put the UltraScale+ out of contention as the replacement for the Acadia II. At this time, design simulations for only one chip in the UltraScale family were available for analysis allowing only one device to be verified by analysis. Additional UltraScale family members will be coming on line throughout 2016.

The Xilinx Zynq family of MPSoCs can currently meet the LLEVS threshold goals and meet the objective goal resolution at 60Hz. It would take only a small change in the power consumption of the UltraScale+ ZU9EG to meet the 5.5 watt goal. These power savings may be achievable using the power management capabilities provided on the chip. The increased size in the Zynq MPSoCs can be compensated for in the design of the printed circuit card where it will reside and with the possible reduction of supporting hardware.

The UltraScale+ has four power domains for efficient power management. The four UltraScale+ domains are listed below.

- Battery-power domain in the processing system (PS) containing the real-time clock and battery-backed RAM.
- Low-power domain in the PS containing the RPU, general peripherals, on-chip memory (OCM), platform management unit, and configuration security unit.
- Full-power domain in the PS containing the APU, high-speed peripherals, system memory manager, and DDR controller
- And the programmable logic (PL) is contained within its own power domain

Other than the battery-power domain, which is always on, there is a wide range of operating modes and power levels from which to select. Domains that are not needed can be turned off at boot and then intelligently woken up at an interrupt or event.

The low-power and full-power domains also support power islands on individual engines for even finer-grained control over power. Each Cortex-A53 processor in the APU can be power gated, while the two Cortex-R5 processors in the RPU can be power-gated together, and the pixel and geometry processors in the GPU are individually gated. Tightly-coupled memory to the RPU and on-chip memory (OCM) are further broken into banks that can also be individually gated, including the L2 cache in the APU. Many of the general- and high-speed peripherals can also be individually gated as power islands. This affords the opportunity to reduce power consumption with careful attention to power during the design phase. If it is determined to port the MPSoC design to an ASIC, a reduction in power would be an additional benefit.

The 1% memory shortage estimate during the simulation analysis has the potential for being solved by adjusting the method for onboard memory usage or going to UltraScale+ device with more available memory when they become available.

This analysis has demonstrated the capability of an MPSoC device meeting the current needs for image processing and the potential to meet the objective goals for future sensors.

5.2 High Risk Approach

The results from the high-risk approach of the LLEVS Phase I provide a convincing argument that FPGA with its embedded processor and reconfigurable hardware cores can provide a increased speed over software running on high-performance processor platforms. It provides an alternative to the ASIC technology with its drawback in the dollar cost and the inability to support algorithm changes, whereas FPGA technology offers hardware reconfiguration flexibility for algorithm changes and falls well within the latency parameters.

A more aggressive image processing design when compared to the Acadia II firmware could compensate for the current MPSoC device shortcomings.

6.0 RECOMMENDATIONS

6.1 Low Risk Approach

The Sage Phase I LLEVS effort has provided positive data that a MPSoC solution to the image processor is very feasible. By using the Xilinx Series 7000 and UltraScale+ chips, the project thresholds are definitely achievable and the objective levels are in reach. These results point to a more detailed design being done for an image processor using the best MPSoC technology available at that time, a development system being bread boarded and more definitive testing being conducted. At the very least this effort should produce a design capable of meeting and exceeding the threshold levels and a high probability of meeting the objective levels.

6.2 High-Risk Approach

The Phase II of the low-latency embedded vision processor (LLEVS) project should deliver a method for developing optimal latency and power consumption algorithms specific to reconfigurable LLEVS. Reconfigurable hardware technology such as Field Programmable Gate Array (FPGA) is continuing to advance as an alternative to ASIC technology due to its system-on-a-chip (SoC) integration that comprises high-performance multiprocessors, a graphic processor, memory hierarchy and the programmable hardware. The algorithms to be considered are from the multiresolution analysis – the pyramid and wavelet-based fusion, stabilization, and moving-object tracking.

6.3 Phase II Plan – Low & High Risk Approach

The focus of this Phase II effort is concerned with the design of a next generation image processing system using the data and conclusions generated from the analysis of two viable developmental approaches, the low risk and high risk approaches, in the Phase I effort. The intent of this Phase II undertaking is to use the information gathered from the low risk approach in Phase I to drive the design of a state of the art processor solution using primarily Commercial Off The Shelf (COTS)components supported by readily available design and simulation tools . At the same time the high risk approach will continue to develop and simulate improved imaging algorithms to determine their effectiveness in reaching the Phase II objective goals. If needed to meet the requirements, the high risk improved algorithms will be implemented as part of the low risk design. In a Phase III implementation the analysis accomplished in Phase I & II will be used to investigate the use of an ASIC. When the LLEVS is at the stage to be used in a device ready for production, the MPSoC device can be ported to an ASIC design. Porting an MPSoC to an ASIC involves less risk than going directly to an ASIC design. If the MPSoC design is developed with an ASIC being the final target device, cost and time can be greatly reduced in the conversion. When the ASIC is completed there will be a significant reduction in power and cost compared to a MPSoC. These factors will be a benefit in larger production quantities where the cost savings of the device are sufficient to cover the Non-Recurring Engineering and foundry costs.

The primary objectives of the Phase II effort can be delineated as follows:

- In accordance with the Phase I architecture and development plan, generate a processor design based on the Xilinx UltraScale architecture and device families that will support the LLEVS threshold through objective processor performance requirements.

- Host the specified Acadia II image processing algorithms on the LLEVS processor design to establish benchmark functions for performance assessment.

- Conduct simulations of the processing functions using the design and simulation tools in order to measure and estimate the performance parametrics. Specifically identify the performance shortfalls to determine the potential system compromises and areas for design optimization and upgrade.

- Host and benchmark the high risk approaches to determine potential design upgrades that may support realization of the objective performance goals.

- Implement a limited representative system using COTS modules, peripherals and prototyping tools to provide a test platform on which processing and process functions can be executed to validate the performance results achieved in the simulation testing.

6.4 Technical Road Map

The makers of MPSoC technology are constantly striving to improve the speed, power consumption and capabilities of their devices. The constant need to improve is being driven by the portable consumer devices and their manufacturers who always want to be on the leading edge of technology. The consumer goods requirements for these devices will drive the cost down further.

This technology road map will focus on the Xilinx MPSoC devices used in the Phase I analysis. The Xilinx Zynq Series 7000 MPSoC were developed using 28nm technologies. All of this family of devices is available for purchase in quantity.

The Xilinx UltraScale+ family of chips was developed with 16 nm technology. Some of these devices are available now and some will be available in the second half of 2016.

The latest technology in development is 7 nm. This technology is being developed for Xilinx in collaboration with Taiwan Semiconductor Manufacturing Company (TSMC). TSMC also collaborated with Xilinx on the 28 nm, 20 nm and 16 nm technology devices. New 7 nm products are planned to be introduced by Xilinx in 2017. However TSMC's goal is to begin 7nm volume production by 2018. So this leaves the exact release of Xilinx 7 nm parts in question.

TSMC has plans to begin 5 nm production by 2018. There is no current indication if Xilinx will develop new MPSoP devices based on this technology.

7.0 REFERENCES

1. W. Sheridan, G. S. van der Wal, *Vision Processor for Helmet System (VPHS) Final Report*, Sage Technologies Ltd, One Ivybrook Blvd, Warminster, PA, USAF AFMC 711 HPW/RHCV, AFRL-RH-WP-TR-2014, February 28, 2014

2. SRI International Acadia II Product Guide, 2011 https://www.sri.com/sites/default/files/brochures/sri_acadia_product_family.pdf

3. D. Zhang,, S. Chai, G. Van der Wal, *Method of Image Fusion and Enhancement Using Mask Pyramid,* SRI International, July, 2011 https://www.researchgate.net/publication/252027510_Method_of_image_fusion_and_enhancement_using_mask_pyramid

4. Xilinx, *Zynq-7000 All Programmable SoC Architecture Porting Quick Start Guide* UG1181, v1.1.1, October 22, 2015, http://www.xilinx.com/support/documentation/user_guides/ug1181-zynq-7000-architecture-porting.pdf

5. Xilinx, Zynq UltraScale+ MPSoC Product Tables and Product Selection Guide, http://www.xilinx.com/support/documentation/selection-guides/zynq-ultrascale-plus-product-selection-guide.pdf

6. Xilinx, Zynq UltraScale+ MPSoC Technical Manual, UG1085 v1.0, November 24, 2015, http://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf

7. H.B. Mitchell, *Image Fusion Theories, Techniques and Applications* 2010 Springer-Verlag, Berlin Heidelberg

8. J. Shen, Y. Zhao, S. Yan and X. Li, “IEEE Exposure Fusion Using Boosting Laplacian Pyramid,” Jianbing IEEE Transactions on Cybernetics 2013

9. K. Lee, J. Kriesel, and N. Gat, “Night Vision Camera Fusion with Natural Colors Using a Spectral / Texture Based Material Identification Algorithm,” February Opto-Knowledge Systems, Inc. (OKSI) 19805 Hamilton Ave. Torrance CA 90502, 2010

10. P. Hill, N. Canagarajah and D. Bull Image Fusion using Complex Wavelets, BMVC 2002

11. Y. Yang, “Multiresolution Image Fusion Based on Wavelet Transform By Using a Novel Technique for Selection Coefficients,” Journal of Multimedia, vol. 6, no. 1, February 2011

12. M. Drahanský, F. Orság, and P. Hanáček, “Accelerometer Based Digital Video Stabilization for General Security Surveillance Systems,” International Journal of Security and Its Applications Vol. 4, No. 1, January, 2010

13. J. Yang, D. Schonfeld andM. Mohamed, “Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion,” IEEE Transactions on Circuits And Systems For Video Technology, Vol. 19, No. 7, July 2009

14. Y. Matsushita, E. Ofek, Weina Ge, X. Tang and H. Y. Shum, “Full frame video stabilization with motion inpainting”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 7, (2006) July, pp. 1163-1178.

15. I. Cohen G. Medioni, “Detecting and Tracking Moving Objects for Video Surveillance,” IEEE Proc. Computer Vision and Pattern Recognition, 1999

16. H. Cho, Y. Seo, B.V.K. Vijaya Kumar, and R. Rajkumar, “A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments,” IEEE International Conference on Robotics & Automation (ICRA) 2014

17. XCell Journal Issue 90, First Quarter 2015, “Xilinx 16nm UltraScale+ Devices Yield 2-5X Performance/Watt Advantage, http://www.xilinx.com/publications/archives/xcell/Xcell90.pdf

18. Xilinx UltraScale+ Architecture and Product Overview, DS890 (v2.6) Dec. 2015, http://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf

19. N. Mehta, Xilinx UltraScale Architecture for High-Performance Smarter Systems, WP434 (v1.2) Oct. 29. 2015

20. Xilinx Zynq UltraScale+ MPSoC Overview, Advance Product Specification, 2015, Nov. 2015, DS891 (v1.0) http://www.xilinx.com/support/documentation/data_sheets/ds891-zynq-ultrascale-plus-overview.pdf
21. Test Pattern Generator v4.00a www.xilinx.com PG103

22. AXI Interconnect Product Guide v2.0 www.xilinx.com PG059

23. Video Timing Controller v6.1 PG016

24. Zynq-7000 AP SoC ZC702 Base TRD www.xilinx.com UG925 (v8.0)
25. AXI VDMA v6.2 www.xilinx.com PG020

26. Video In to AXI4-Stream v3.0 www.xilinx.com PG044
27. P. Burt and E. Adelson, “The Laplacian Pyramid as a Compact Image Code,” *IEEE transactions on communications*, vol. com-3l, no. 4, April 1983

28. E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt and J. M. Ogden, “Pyramid methods in image processing,” *RCA Engineer* 29-6 Nov/Dec 1984

29. OpenCV Image Filtering, http://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html

30. Tegra K1 Processor, http://www.nvidia.com/object/tegra-k1-processor.htm

31. NVIDIA’s Next Generation CUDA Compute Architecture: Kepler GK110/210, Whitepaper v1.1

32. Anupama Shankar Kurpad, “A Comparative Performance Analysis of the Phase Recovery Algorithm for Microstructure Reconstruction,” Master’s Thesis Drexel University, 2009

33. The NVIDIA CUDA Deep Neural Network library, https://developer.nvidia.com/cudnn

34. Automotive Technology, http://www.nvidia.com/object/drive-automotive-technology.html
35. OpenCV GPU Module, http://docs.opencv.org/2.4/modules/gpu/doc/gpu.html

36. M. Strengert, M. Kraus, and T. Ertl, “Pyramid Methods in GPU-Based Image Processing,” Institut fur Visualisierung und Interaktive Systeme (VIS) Universitatsstraße, Stuttgart, Germany, 2006 http://wwwvis.informatik.uni-stuttgart.de/~kraus/preprints/vmv06_strengert.pdf

37. J. Lu*, B. Zhang, Z. Gong, E. Li and H. Zhengzhou, “The Remote-sensing Image Fusion based on GPU,” International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B7. Beijing 2008.

38. S. Neuendorffer, T. Li, and D. Wang, “OpenCV Applications with Zynq-7000 All Programmable SoC using Vivado HLS” Video Libraries, XAPP1167 (v3.0) June 24, 2015

APPENDIX A - Intellectual Property, Publications, and Personnel

A.1 Intellectual Property

No intellectual property claims have been asserted for this report.

A.2 Publications and Presentations of Results from This Program

There have been no publications or presentations resulting from this project other than those presented and submitted to the government.

A.3 Professional Personnel Associated with This Program

A3.1 Sage Technologies, Ltd.

Greg Cream
Wesley Sheridan

A3.2 Drexel University

Prawat Nagvajara Ph.D.

A3.3 SRI International

Kevin Kaighn

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AFRL	Air Force Research Lab
AGC	Automatic Gain Control
CP P	Commercialization Pilot Program
APU	Application Processor Unit
ARM	Advanced RISC Machine
ASIC	Application Specific Integrated Circuit
AXI	Advance eXtensible Interface
BMAIS	Binocular Multispectral Adaptive Imaging System
CEVS	Coxswain's Enhanced Vision System
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
CSAR	Combat Search and Rescue
DARPA	Defense Advanced Research Project Authority
DEVS	Digitally Enhanced Vision System
DMBS	Digital Multispectral Binocular System
DDR3	Double Data Rate dynamic RAM
DP	Double Precision
DSP	Digital Signal Processor
DU	Drexel University
DVE	Degraded Visual Environment
DWT	Discrete Wavelet Transform
EMI	Electro Magnetic Interference
FFT	Fast Fourier Transform
FIFO	First in, First out
FOV	Field of View
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
Fvclk	Video Frequency Clock
GIC	General Interrupt Controller
Gops	Giga Operations Per Second

GPIO	General Purpose I/O Discretes
GPU	Graphics Processing Unit
HDL	Hardware Description Language
HDMI	High-Definition Multimedia Interface
HLS	High-Level Synthesis
HMD	Helmet Mounted Display
HMU	Helmet-Mounted Unit
IDE	Integrated Development Environment
IDWT	Inverse Discrete Wavelet Transform
I/O	Input/Output
I2	Image Intensifier
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IP	Intellectual Property
IPRO	Intermediate Level Processor
IR	Infrared
IIR	Infinite Impulse Response
J DEVS	Joint Digital Enhanced Vision System
LLEVS	Low Latency Embedded Vision Processor
LWIR	Long Wave Infrared
MAC FIR	Multiply and Accumulate Finite Impulse Response
MATLAB	Matrix Laboratory – numerical computing environment
MIO	Multiplex Input/Output
MRA	Multiresolution Analysis
MPSoC	Multi Processor System On a Chip
NATO	North Atlantic Treaty Organization
NI	Night Vision
NIR	Near Infrared
NUC	Non Uniform Correction
NVEC	Night Vision Equipment Corporation
NVG	Night Vision Goggle

NVIR	Near-Visible Infrared
NVL	Army Night Vision Lab
OEM	Original Equipment Manufacturers
OLED	Organic Light Emitting Diode
PCB	Printed Circuit Board
PJ’s	ParaRescue Ju mpers
PL	Programmable Logic
pyrUp	Pyramid UP
pyr Down	Pyramid Down
RAM	Random Access Memory
SBIR	Small Business Innovative Research
SDK	Software Development Kit
Ser/Des	Serialization / Deserialization
SFU	Special Function Unit
SME	Subject Matter Expert
SoC	System on a Chip
SOW	Statement of Work
SRI	SRI International
STTR	Small Business Technology Transfer
SWaP	Size, Weight and Power
SWIR	Short Wave InfraRed
TPG	Test Pattern Generator
TPOC	Technical Point of Contact
UART	Universal Asynchronous Receiver/Transmitter
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
VDAM	Video Direct Memory Access
VPHS	Video Processor for Helmet Systems
VTC	Video Timing Controller
YUV	Image color Space (luminance-Y, color components-UV)
Zynq	Xilinx FPGA/MPSoC family