

**AN EFFICIENT ENERGY AWARE ADAPTIVE
SYSTEM-ON-CHIP ARCHITECTURE FOR
REAL-TIME VIDEO ANALYTICS**

HISHAM AHMED ALI AHMED

UNIVERSITI SAINS MALAYSIA

2016

**AN EFFICIENT ENERGY AWARE ADAPTIVE
SYSTEM-ON-CHIP ARCHITECTURE FOR REAL-TIME VIDEO
ANALYTICS**

by

HISHAM AHMED ALI AHMED

**Thesis submitted in fulfilment of the
requirements for the degree of
Doctor of Philosophy**

October 2016

ACKNOWLEDGEMENTS

First and above all, I thank the Almighty Allah for his endless blessings.

This thesis would not have been possible without the support of many people over the years of my study. First of all, I would like to present my cordial gratitude to my supervisor Prof. Dr. Othman Sidek for his guidance and support. He always able to push ideas one level further, and shares his rich experience about the research process. His contagious enthusiasm, technical passion, work ethics, and qualities have greatly inspired me.

I am also grateful to the Universiti Sains Malaysia and Prof. Dr. Othman Sidek for supporting my research work through the Research Creativity and Management Office (RCMO) grant (1001/PCEDEC/854003) and the USM Fellowship.

Last but not the least, I would like to thank my parents, Ahmed and Suad, and my wife Arwa, and my two young daughters, Ragad and Razan, for their sincere prayers, unconditional love, and never-ending support over the past years.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iii
LIST OF TABLES	ix
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xvi
ABSTRAK	xxii
ABSTRACT	xxiv
CHAPTER ONE: INTRODUCTION	
1.1 Applications of Video Analytics	1
1.2 Requirements of Embedded Video Analytics	2
1.3 Hardware Architectures for Embedded Vision	3
1.4 Programmable System-on-Chip Trend	6
1.5 Statement of the Problem	8
1.6 Thesis Objectives	11
1.7 Summary of Contributions	13
1.8 Thesis Outline	14
CHAPTER TWO: BACKGROUND AND RELATED WORKS	
2.1 Introduction	17
2.2 Review of Analytical Performance Models	19
2.2.1 Performance Modelling in Embedded Systems	20

2.2.2	Analytical Performance Modelling for Multicore Processors	22
2.2.3	The Roofline Performance Model	23
2.2.4	Performance Modelling for Field Programmable Gate Arrays	27
2.2.5	Lack of Performance Models for Zynq All Programmable System-on-Chip	30
2.3	Harris Corner Detection Algorithm	32
2.4	Review of the Implementations of Harris Corner Detection Algorithm on Field Programmable Gate Arrays	34
2.5	Partial Reconfiguration in Field Programmable Gate Array	38
2.5.1	Partial Reconfiguration Types and Design Flows	41
2.5.2	Configuration Process	42
	2.5.2(a) Full Configuration Bitstream	42
	2.5.2(b) Partial Configuration Bitstream	42
2.5.3	Configuration Modes	43
2.5.4	Partial Reconfiguration Benefits	44
2.5.5	Partial Reconfiguration Limitations	45
2.5.6	Review of Dynamic Partial Reconfiguration Implementations of Video and Image Processing	47
2.6	Conclusion	51

**CHAPTER THREE: APSOC-ROOFLINE: AN ANALYTICAL
PERFORMANCE MODEL FOR ZYNQ ALL PROGRAMMABLE
SYSTEM-ON-CHIP**

3.1	Introduction	53
3.2	Zynq All Programmable System-on-Chip Characterization	54

3.2.1	The Processing System Characterization	55
3.2.2	The Programmable Logic Characterization	58
3.2.3	Processing System and Programmable Logic Interfaces	65
3.3	APSoC-Roofline Performance Model	67
3.3.1	Attainable Performance on the Processing System	67
3.3.2	Attainable Performance on the Programmable Logic	68
3.4	Elaboration and Validation of the APSoC-Roofline Performance Model	69
3.5	Integration of the APSoC-Roofline Model in Hardware/Software Codesign Flow	75
3.6	Remarks on APSoC-Roofline Model	78
3.7	Conclusion	79

CHAPTER FOUR: A SYSTEM-ON-CHIP ARCHITECTURE FOR REAL-TIME HARRIS CORNER DETECTION ALGORITHM

4.1	Introduction	81
4.2	Design of the System-on-Chip using Hardware/Software Codesign	82
4.2.1	System Specifications	82
4.2.2	Hardware/Software Partitioning and Mapping	82
4.3	Implementation of Harris Corner Detection Algorithm on FPGA using High Level Synthesis	87
4.3.1	LineBuffer and WindowBuffer Structure	89
4.3.2	Performing Sub-operations using LineBuffer and WindowBuffer	90
4.4	Implementation of Harris Corner Detection Algorithm on FPGA using Fixed-point Representation	93
4.4.1	Fixed-point Representation	94

4.4.2	Floating-point to Fixed-point Conversion	94
4.5	Overall System-on-Chip Architecture	97
4.5.1	The System-on-Chip Hardware Architecture	97
4.5.1(a)	AXI Interconnect IP Core	97
4.5.1(b)	Test Pattern Generator IP Core	98
4.5.1(c)	FMC Imageon HDMI In IP Core	98
4.5.1(d)	Video In to AXI4-Stream IP Core	99
4.5.1(e)	Video Timing Controller IP Core	99
4.5.1(f)	AXI Video Direct Memory Access IP Core	99
4.5.1(g)	LogiCVC-ML IP Core	100
4.5.2	The System-on-Chip Software Architecture	100
4.5.2(a)	Linux User Space	101
4.5.2(b)	Linux Kernel Space	102
4.6	Results and Discussion	102
4.6.1	Floating-point and Fixed-point Implementations Results	103
4.6.2	Comparison of the Hardware and Software Implementations of Harris Corner Detection Algorithm	105
4.6.3	Comparison of the Harris Corner Detection Algorithm Implementations on FPGA	106
4.6.4	Analysis of the Real-Time Performance of Harris Corner Detection Intellectual Property Cores	109
4.6.5	The Complete Real-Time System-on-Chip for Video Analytics Integration and Prototyping	110
4.7	Conclusion	112

CHAPTER FIVE: AN ENERGY-AWARE ADAPTIVE

SYSTEM-ON-CHIP ARCHITECTURE FOR REAL-TIME VIDEO

ANALYTICS

5.1	Introduction	114
5.2	Implementation of Adaptive System-on-Chip using Dynamic Partial Reconfiguration	115
5.3	Dynamic Partial Reconfiguration using Processor Configuration Access Port	118
5.4	Implementation of the Energy Aware Adaptive System-on-Chip	120
5.5	Power Measurement Setup	125
5.6	Results and Discussion	126
5.6.1	Analysis of the Reconfiguration Overhead Effect on the Performance	127
5.6.2	Analysis of the Impact of the Dynamic Partial Reconfiguration on Power Consumption	129
5.7	Conclusion	133

CHAPTER SIX: CONCLUSION AND FUTURE WORK

6.1	Conclusion	134
6.2	Future Work	136

REFERENCES	138
-------------------	-----

APPENDICES

Appendix A: Single-Precision General Matrix-Matrix Multiplication

(SGEMM) C Code

Appendix B: Stencil Algorithm C Code

Appendix C: Harris Corner Detection Algorithm C Code For High Level

Synthesis

Appendix D: CONTEXT-AWARE CONFIGURATION SCHEDULER

APPLICATION C CODE

LIST OF TABLES

		Page
Table 2.1	Power and energy consumption of parametrized architecture for HCD (Possa, Mahmoudi, Harb, Valderrama, & Manneback, 2014).	36
Table 2.2	Harris corner detection algorithm implementations on FPGA.	37
Table 2.3	Field Programmable Gate Array configuration modes bandwidth (Xilinx, 2015i, 2015l).	44
Table 3.1	Processing system parameters.	56
Table 3.2	DRAM memory parameters.	57
Table 3.3	Adders and Multipliers operators subtypes (Xilinx, 2015j).	60
Table 3.4	Operators utilization and performance.	62
Table 3.5	The Zynq-7020 APSoC (XC7Z020) available resources.	62
Table 3.6	The peak performance of Zynq-7020 APSoC (PL _{peak}).	63
Table 3.7	BRAM Memory Parameters	65
Table 3.8	PS-PL interfaces details and bandwidths (Xilinx, 2015l).	66
Table 3.9	APSoC-Roofline performance model parameters.	70
Table 4.1	Analysis of Harris corner detection algorithm.	86
Table 4.2	Harris corner detection algorithm data types. W denotes the word length, I denotes the integer part, AP_RND denotes rounding to plus infinity in quantization, and AP_SAT denotes saturation if overflow occurs.	96

Table 4.3	Comparison of floating-point and fixed-point implementations of Harris corner detection algorithm on Zynq-7020 APSoC programmable logic.	104
Table 4.4	Comparison of Harris corner detection algorithm implementation in different platforms for HD1080 (i.e., 1920x1080) resolution.	105
Table 4.5	Comparison of the performance of Harris corner detection algorithm implementations on FPGA.	106
Table 4.6	Comparison of the resources usage and performance of Harris corner detection algorithm implementations on Zynq-7020 APSoC.	108
Table 4.7	Comparison of the real-time requirements of three video standards and the performance of the developed Harris corner detection IP cores.	110
Table 4.8	FPGA resources usage of the Harris corner detection algorithm real-time System-on-Chip.	110
Table 5.1	Real-time requirements and reconfiguration time for the HD1080p, HD720p, and VGA video standards.	128
Table 5.2	Power and energy consumption of the reconfigurable modules measured at run-time.	129
Table 5.3	User-defined constraints.	130

Table 5.4	Parametrized IP core for 1920x1080, 1280x720, and 640x480 resolutions power and energy consumption.	131
Table 5.5	Comparison of the operation time of the HCD processing for 1920x1080, 1280x720, and 640x480 video resolutions using DPR and parametrized IP methods.	131
Table 5.6	Achieved energy saving when the HCD IP core changed from higher to lower resolution at run-time using Dynamic Partial Reconfiguration (DPR).	132

LIST OF FIGURES

		Page
Figure 1.1	Comparison of performance and flexibility of several architectures in embedded video processing (Xilinx, 2015c).	4
Figure 1.2	Zynq All Programmable System-on-Chip (Xilinx, 2016e).	7
Figure 1.3	System-on-Chips power consumption, and battery capacity projections (ITRS, 2012).	10
Figure 1.4	Thesis outline.	16
Figure 2.1	Roofline performance model graph depicts two hypothetical algorithms, A1 and A2, running on three different hypothetical machines in (a), (b), and (c).	24
Figure 2.2	Use of Dynamic Partial Reconfiguration (DPR) to time multiplex several reconfigurable modules in a reconfigurable partition.	39
Figure 2.3	Reconfigurable frames base regions for CLB, DSP48, and BRAM in Zynq-7020 APSoC.	40
Figure 2.4	Bitstream files format and configuration process (Xilinx, 2015i).	43
Figure 3.1	Zynq-7020 APSoC architecture. Adapted from (Xilinx, 2015l).	54
Figure 3.2	Abstract SoC model.	56
Figure 3.3	Programmable logic architecture overview. Adapted from (Xilinx, 2014a, 2014b, 2014c).	59
Figure 3.4	APSoC-Roofline performance model graph.	68

Figure 3.5	APSoC-Roofline performance model elaboration using Single-Precision General Matrix-Matrix Multiplication (SGEMM).	72
Figure 3.6	APSoC-Roofline performance model validation for processing system and programmable logic using SGEMM. $SGEMM_{ps}$ and $SGEMM_{pl}$ denote SGEMM running on the PS and the PL, respectively.	74
Figure 3.7	APSoC-Roofline performance model validation for processing system and programmable logic using stencil algorithm. $Stencil_{ps}$ and $Stencil_{pl}$ denote Stencil running on the PS and the PL, respectively.	76
Figure 3.8	Augmenting Hardware/Software codesign flow with APSoC-Roofline model for HW/SW partitioning and mapping steps.	77
Figure 4.1	Harris corner detection algorithm profiling on dual ARM Cortex-A9.	85
Figure 4.2	APSoC-Roofline model graph for Harris corner detection algorithm showing the predicted attainable performance AP_{ps} , $AP_{pl_{ps-pl}}$, and $AP_{pl_{pl-bram}}$ of the algorithm operations.	87
Figure 4.3	Performing the convolution operation using sliding window method.	88
Figure 4.4	Harris corner detection algorithm operations.	89
Figure 4.5	Window-based operations parallelization and pipelining.	91

Figure 4.6	Harris corner detector IP core architecture.	92
Figure 4.7	Fixed-point numbers representation.	95
Figure 4.8	Overall System-on-Chip hardware architecture.	98
Figure 4.9	Overall System-on-Chip software architecture.	100
Figure 4.10	APSoC-Roofline model graph for Harris corner detection algorithm showing the achieved performance $HCD(AP_{pl_{pl-bram}})$ (Not-Optimized) and $HCD(AP_{pl_{pl-bram}})$ (Optimized) after the implementation.	103
Figure 4.11	The results of fixed-point and floating-point implementations of Harris corner detection algorithm tested on 256x256 resolution checkerboard image.	104
Figure 4.12	Complete system setup.	111
Figure 5.1	A design flow to implement the adaptive System-on-Chip using Dynamic Partial Reconfiguration (DPR).	116
Figure 5.2	Device view of the Zynq AP SoC. The static logic is highlighted with the orange colour, and the reconfigurable partition is marked with the purple rectangle.	117
Figure 5.3	Adaptive System-on-Chip partial reconfiguration through the Processor Configuration Access Port (PCAP).	119
Figure 5.4	Energy-aware operation modes selection.	121
Figure 5.5	Context-aware configuration scheduler flowchart.	122
Figure 5.6	Dynamic Partial Reconfiguration subroutine flowchart.	123
Figure 5.7	Power measurement setup.	126

Figure 5.8 Power consumption measurement using Texas Instruments Fusion Digital Power Designer. The graph shows the power in (Watt) on the vertical axis and time on the horizontal axis.

LIST OF ABBREVIATIONS

ACP	Accelerator Coherency Port
ADAS	Advanced Driver Assistance System
AES	Advanced Encryption Standard
AFAS	Automated Fingerprint-based Authentication System
AMBA	ARM Advanced Microcontroller Bus Architecture
AMP	Asymmetrical Multiprocessing
ANPR	Automatic Number Plate Recognition
APSoC	All Programmable System-on-Chip
ASIC	Application Specific Integrated Circuit
ASIP	Application-Specific Instruction Set Processor
ASSP	Application-specific standard parts
ATLAS	Automatically Tuned Linear Algebra Software
AVS	Adaptive Voltage Scaling
AXI	Advanced eXtensible Interface
BFM	Bus Functional Model
BLAS	Basic Linear Algebra Subprograms
BRAM	Block RAM

BSCAN	Boundary-Scan Interface
BUFG	Global Clock Buffer
BUFR	Regional Clock Buffer
CCTV	Closed-Circuit Television
CLB	Configurable Logic Block
CNN	Convolutional Neural Network
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
CPS	Cyber-Physical Systems
CRC	Cyclic Redundancy Check
DCP	Design Checkpoint
DDR3	Double Data Rate Type Three
DMA	Direct Memory Access
DPR	Dynamic Partial Reconfiguration
DRAM	Dynamic Random Access Memory
DSE	Design Space Exploration
DSP	Digital Signal Processor
DVFS	Dynamic Voltage and Frequency Scaling
ECC	Error Checking and Correction

EMIO	Extended Multiplexed Input/Output
FF	Flip Flop
FFT	Fast Fourier Transform
Flops	Floating-point operations
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
FPS	Frame Per Second
GP	General Purpose
GPU	Graphic Processing Unit
GUI	Graphic User Interface
HCD	Harris Corners Detection
HD	High-Definition
HDMI	High-Definition Multimedia Interface
HLS	High Level Synthesis
HMAC	keyed-Hash Message Authentication Code
HP	High Performance Port
HPC	High Performance Computing
HW	Hardware
ICAP	Internal Configuration Access Port

IDCT	Inverse Discrete Cosine Transform
IOB	Input/Output Block
IoT	Internet of Things
IP	Intellectual Property
ISERDES	Input Serializers/Deserializers
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group
LUT	Look Up Table
LZW	Lempel–Ziv–Welch
MCAP	Media Configuration Access Port
MGT	Multi-Gigabit Transceiver
MMCM	Mixed-Mode Clock Manager
MPSoC	Multiprocessor System-on-Chip
NMS	Non-Maximum Suppression
NRE	Non-Recurring Engineering
OI	Operational Intensity
OS	Operating System
OSERDES	Output Serializers/Deserializers
PCAP	Processor Configuration Access Port

PCIe	Peripheral Component Interconnect Express
PL	Programmable Logic
PLL	Phase Locked Loop
PMBus	Power Management Bus
PMM	Power Mode Management
PS	Processing System
QoS	Quality of Service
RISC	Reduced Instruction Set Computing
RM	Reconfigurable Modules
RP	Reconfigurable Partition
RTL	Register Transfer Language
SD	Secure Digital
SGEMM	Single-precision GEneral Matrix-matrix Multiplication
SIMD	Single Instruction Multiple Data
SIR	Sampling Importance Resampling
SLAM	Simultaneous Localization And Mapping
SMP	Symmetrical Multiprocessing
SoC	System-on-Chip
SUSAN	Smallest Univalve Segment Assimilating Nucleus

SW	Software
Tcl	Tool Command Language
UART	Universal Asynchronous Receiver/Transmitter
UAV	Unmanned Aerial Vehicle
UHD	Ultra High Definition
VCA	Video Content Analysis
VGA	Video Graphics Array
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VLIW	Very Long Instruction Word
WMSN	Wireless Multimedia Sensor Network
XADC	Xilinx Analogue-to-Digital Converter
XDC	Xilinx Design Constraint
XSDK	Xilinx Software Development Kit

SENI BINA BOLEH SUAI BAGI SISTEM-ATAS-CIP PEKA-TENAGA CEKAP BAGI ANALITIK VIDEO MASA-NYATA

ABSTRAK

Aplikasi analitik video yang kebanyakannya ada pada peranti terbenam semakin kerap digunakan kini. Pertumbuhan pesat yang ditunjukkan ini menyebabkan perlunya Sistem-atas-Cip (SoC) dibangunkan untuk menjalankan pemprosesan terbaik pada cip tunggal berbanding pada komponen diskret. Penglihatan terbenam tertakluk kepada keperluan yang ketat, iaitu prestasi masa-nyata, tenaga yang terhad, dan kemudahsuaian untuk mendepani evolusi piawaian. Tambahan pula, untuk mereka bentuk SoC yang sedemikian kompleks, khususnya SoC Boleh Atur Cara Semua Zynq, pendekatan reka bentuk yang selari untuk perkakasan/perisian tradisional bergantung pada pemprofilan perisian untuk menjalankan pemetakan perkakasan/perisian tidak mampu lagi menjalankan tugas ini kerana pemprofilannya tidak dapat meramal prestasi aplikasi pada perkakasan. Oleh itu, satu model yang menghubungkan ciri-ciri kepada prestasi platform adalah sangat penting untuk dibangunkan. Untuk menghantar prestasi masa-nyata bagi resolusi video yang pantas berkembang sambil menjaga kelenturan seni binanya pada pemproses, Unit Pemprosesan Grafik, Pemproses Signal Digital, dan Litar Bersepadu Aplikasi-Spesifik, ia tidak dapat dibuat. Selanjutnya, dengan penskalaan teknologi semikonduktor, dijangka bahawa pelepasan kuasa akan meningkat kerana kapasiti bateri dijangka tidak akan meningkat dengan mendadak. Model prestasi bagi Zynq dibangunkan dengan menggunakan kaedah analitis dan digunakan dalam reka bentuk selari bagi perkakasan/perisian adalah untuk membantu pemetaan algoritma bagi perkakasan. Selepas itu, SoC bagi analitik video masa-nyata direalisasikan pada Zynq

dengan menggunakan algoritma pengesanan sudut Harris. Analisis yang teliti terhadap algoritma tersebut dan penggunaan yang cekap pada sumber Zynq menghasilkan seni bina yang bukan sahaja terselari dan tertalipaipan, malah melebihi prestasi algoritma yang paling terkini. Dengan menjalankannya pada SoC peka-kuasa yang boleh ubah serta dibangunkan dengan menggunakan pengkonfigurasiannya semula separuh dinamik, aplikasi penjadual konfigurasi yang peka-konteks akan mengikut konteks operasi dan menukarkan resolusi video dengan penggunaan kuasa bagi menampung masa operasi yang lama ketika menghantar prestasi masa-nyata. Pengesanan sudut masa-nyata pada 79.8, 176.9, dan 504.2 bingkai sesaat tercapai, iaitu masing-masing bagi HD1080, HD720, dan VGA. Ketiga-tiga bingkai sesaat berjaya mengatasi prestasi kajian terdahulu dengan gandaan 31 kali lebih baik bagi HD720 dan 3.5 kali bagi VGA. Penjadual berfungsi pada ketika proses konfigurasi berjalan. Pada ketika itu, perkakasan yang sesuai digunakan di mana ia dapat memenuhi konteks operasi dan halangan yang didefinisikan oleh pengguna; dalam kalangan pemecut yang dibangunkan sebagai contoh piawai video HD1080, HD720, dan VGA menggunakan tenaga yang rendah. Kaedah penyesuaian diri berjaya mencatatkan tempoh masa operasi yang lebih panjang berbanding dengan teras parameter IP untuk kadar kapasiti bagi bateri yang sama iaitu sebanyak 1.77 kali. Di samping itu, lebihan pengkonfigurasiannya tenaga boleh diabaikan bagi kaedah ini. Kesan pada kelewatan masa bagi pengkonfigurasiannya masa separuh diperhatikan, contohnya, hanya dua bingkai video diturunkan bagi HD1080p60 ketika masa pengkonfigurasiannya semula. Pemudahan proses reka bentuk dengan model analisis, dan penggunaan sumber Zynq serta keputusan adaptivity diri dalam peka-tenaga SoC dengan cekap, ini menyediakan prestasi masa-nyata untuk video analitik.