



Cairo University

FRONT-END ENHANCEMENTS FOR CAIRO UNIVERSITY
HARDWARE/SOFTWARE PARTITIONING TOOL
(CUPSHOP)

By

Ahmed Hamed Fatehy Hamed

A THESIS SUBMITTED TO THE
FACULTY OF ENGINEERING AT CAIRO UNIVERSITY
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRONICS AND COMMUNICATIONS ENGINEERING

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2015

FRONT-END ENHANCEMENTS FOR CAIRO UNIVERSITY
HARDWARE/SOFTWARE PARTITIONING TOOL
(CUPSHOP)

By
AHMED HAMED FATEHY HAMED

A THESIS SUBMITTED TO THE
FACULTY OF ENGINEERING AT CAIRO UNIVERSITY
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under the Supervision of

Prof.
Serag Eldin Elsayed Habib

Associate Prof.
Mohamed Bakr Abdelhalim Osman

Professor
Electronics and Communications
Department
Faculty of Engineering
Cairo University

Associate Professor
College of Computing and Information
Technology (CCIT)
Arab Academy of Science and Technology and
Maritime Transport (AASTMT)

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2015

FRONT-END ENHANCEMENTS FOR CAIRO UNIVERSITY
HARDWARE/SOFTWARE PARTITIONING TOOL
(CUPSHOP)

By
AHMED HAMED FATEHY HAMED

A THESIS SUBMITTED TO THE
FACULTY OF ENGINEERING AT CAIRO UNIVERSITY
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRONICS AND COMMUNICATIONS ENGINEERING

Approved by the
Examining Committee

Prof.Elsayed Mostafa Saad (Helwan University)

Prof. Ashraf Alfarghaly Salem (Ain-Shams University)

Prof. Serag Eldin Elsayed Habib, Thesis Main Advisor (Cairo University)

Dr. Mohamed Bakr Abdhalim Osman, Thesis Advisor (AASTMT)

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2015

Engineer's Name: Ahmed Hamed Fatehy Hamed
Date of Birth: 24/12/1985.
Nationality: Egyptian.
E-mail: engineerahmedhamed@gmail.com
Phone: 01287785555
Address: 1st of Abu-Bakr Al-Sediek st.,
Kornish El-Neel, Warak,
Giza, Egypt
Registration Date:/...../.....
Awarding Date:/...../.....
Degree: Master of Science.
Department: ELECTRONICS AND ELECTRICAL
COMMUNICATIONS ENGINEERING



Supervisors: Prof. Serag Eldin Elsayed Habib
Associate Prof. M. B. Abdelhallim (AASTMT)

Examiners: Prof. Elsayed Mostafa Saad (External examiner)
Prof. Ashraf Alfarghaly Salem (External examiner)
Prof. Serag Eldin Elsayed Habib (Thesis main advisor)
Associate Prof. M. B. Abdelhallim (Thesis advisor)

Title of Thesis:
FRONT-END ENHANCEMENTS FOR CAIRO UNIVERSITY
HARDWARE/SOFTWARE PARTITIONING TOOL (CUPSHOP)

Key Words:
HW/SW Co-Design, High Level Synthesis; C Front-End; Embedded Systems; Control-Data Flow Graphs; Intermediate Representations;

Summary:
This thesis introduces a Front-End enhancement for Cairo University Particle Swarm optimization Hardware/software Partitioning (CUPSHOP) tool by enabling this tool to accept input designs written in ANSI C. This Front end can accept hierarchical input C codes of arbitrary number of levels. It extracts a corresponding control and data flow diagram (CDFG) representation of the system, together with all the data base information needed to define the design completely. This CDFG is fully compatible with CUPSHOP, and can also be used with several HLS and HW/SW partitioning tools. The developed C front end is tested successfully against several published HLS benchmarks, including benchmarks from CHStone suite.

Acknowledgments

The work done in this thesis wouldn't have been conducted without continues support and guidance of my advisors, and I would like to seize the opportunity to express my gratitude towards Prof. Dr. Serag El-Din Habib and Dr. Mohammed Bakr for their advice and guidance throughout my work. I would like to express my gratitude to my teammate Eng. Rania Osama, who cooperated with me at several points through my work.

Dedication

To my Father, my Mother and my Wife

“THE ENDLESS SOURCE OF SUPPORT IN MY LIFE”

Table of Contents

Table of Contents

Acknowledgments.....	I
Dedication	II
Table of Contents.....	III
List of Figures	VI
List of Tables	IX
Nomenclature.....	X
Abstract	XI
Chapter 1. Introduction	1
1.1 Introduction to High-level Synthesis HLS	1
1.2 Introduction to HW/SW Co-Design.....	4
1.3 Introduction to CUPSHOP version 1.0	6
1.4 Research Motivation	7
1.5 Thesis Methodology Proposed enhancements to CUPSHOP's Front-End.....	9
1.6 Thesis Organization.....	9
Chapter 2. Literature Review.....	11
2.1 High Level synthesis Literature Overview.....	12
2.1.1 High-level synthesis in the Gasjki-Kuhn Y-chart.....	12
2.1.2 High-Level Synthesis Key concepts	13
2.1.3 HLS and Design Exploration Perspectives	20
2.1.4 High- Level Synthesis Evaluation Criteria	20
2.1.5 Common HLS tools	21
2.2 Introduction to HW/SW partitioning problem	28
2.3 HW/SW Co-Design Tools.....	30
2.3.1 POLIS: A Framework for Hardware-Software Co-Designs	30
2.3.2 COOL: A HARDWARE/SOFTWARE CO-DESIGN TOOL.....	33
2.4 Intermediate representations formats and techniques	34
2.5 State of the art representations (DFG, CFG, and CDFG)	34
2.6 Converting the input sequential description to a CDFG	37
2.7 Compiler Transformations	37
2.7.1 Constant folding.....	37

2.7.2	Redundant operator elimination and Dead Code Elimination	38
2.7.3	Tree height transformation.....	39
2.8	HLS Tools and Intermediate Representations Review	40
2.8.1	SPARK's Intermediate Representation.....	40
2.8.2	CDFG TOOLKIT's Intermediate Representation	41
2.8.3	GAUT's Intermediate Representation	44
2.8.4	LegUp's Intermediate Representation	46
2.9	Scheduling.....	47
2.9.1	ASAP (As Soon As Possible) Scheduling Algorithm.....	47
2.9.2	ALAP (As Late As possible) Scheduling Algorithm.....	48
2.9.3	List Scheduling Algorithm.....	48
2.10	CUPSHOP's Intermediate Representation.....	49
Chapter 3.	CUPSHOP's C Front-End.....	50
3.1	SPARK generated VHDL	50
3.2	Limited VHDL set of CDFG toolkit	52
3.3	GAUT's CDFG compiler.....	53
3.4	Characteristic of Input C Code Handled by GAUT	53
3.4.1	Data types.....	53
3.4.2	Statements	54
3.4.3	Functions.....	55
3.4.4	Expressions	57
3.4.5	Conditional compilation and comments	58
3.5	Example of a 4-points Finite Impulse Response (FIR) filter	58
3.6	CDFG File Parsing.....	60
3.7	Eclipse as Hierarchical Call Tree Extractor	62
3.8	Proposed CUPSHOP's Front-end Tool Flow.....	65
3.9	Writing rules and guidance	68
3.10	CUPSHOP GUI.....	70
Chapter 4.	Cases Study	71
4.1	Basic C code with Multiple Levels of Hierarchy.....	72
4.2	Finite Impulse Response (FIR) filter with 16 Taps.....	80
4.3	SOBEL Edge Detector	82
4.4	YUV2RGB, GAUT Example of MJPEG.....	85
4.5	CHStone HLS Benchmarks.....	90

4.6	Discrete Cosine Transform ‘DCT’	91
4.7	Adaptive differential pulse-code modulation (ADPCM)	100
4.8	Proposed Writing rules	106
4.9	Results’ Summary	107
Chapter 5. Conclusions and Future Work		108
	Conclusions	108
	Future work	109
	Stand-Alone Code	109
	Graph Visualization	109
	Eclipse IDE Integration	109
	Interface to GCC GNU	109
	CLANG	109
References		110
Appendix A: SPARK’s generated VHDL Code		116
Appendix B: CDFG Generated file for 4-Taps FIR		119
Appendix C: Finite Impulse Response (FIR) filter with 16-Taps’ C Code		122
Appendix D: SOBEL Edge Detector’s C Code		123
Appendix E: YUV2RGB C Code		126
Appendix F: DiscreteCosine Transform (DCT) C Code		128
Appendix G: Adaptive Differential Pulse-Code Modulation (ADPCM) C Code		134
Appendix H: CUPSHOP’s C Front-End Writing Rules		144

List of Figures

Figure 1.1 Traditional Design Flow [33]	4
Figure 1.2 HW/SW partitioning methodology done by CUPSHOP V1.0	6
Figure 1.3 Flowchart of the CUPSHOP V2.0 HW/SW partitioning methodology	10
Figure 2.1 High-level synthesis in the Gasjki-Kuhn Y-chart [8]	13
Figure 2.2 Various phases of High-Level synthesis as per Mohanty [74]	14
Figure 2.3 A synthesis example: step 1 to step 3, Mohanty [74]	17
Figure 2.4 Synthesis example: step 4 to step 6, Mohanty [74]	18
Figure 2.5 High-level synthesis (HLS) design's steps, Coussy et. al. [1]	19
Figure 2.6 Spider web diagram for HLS Tools Evaluation Board [8]	21
Figure 2.7 SPARK HLS Tool Flow [14]	23
Figure 2.8 GAUT's High Level Synthesis Tool Flow [22]	24
Figure 2.9 LegUp High-Level Synthesis design flow [29]	26
Figure 2.10 POLIS's Design Flow [51]	31
Figure 2.11 DFG Example, C Code on the left, DFG on the right [91]	34
Figure 2.12 CFG Example, Pseudo Code and translated CFG [91]	35
Figure 2.13 CDFG Entity Example, operation vs graph [91].	36
Figure 2.14 Converting the input source code to a CDFG [74]	37
Figure 2.15 Constant Folding [58]	38
Figure 2.16 Redundant Operator Elimination example [58]	38
Figure 2.17 Dead Code Elimination example [73]	39
Figure 2.18 Tree height transformation [58]	39
Figure 2.19 An overview of the SPARK HLS flow [64]	40
Figure 2.20 Translation from C input into HTG [67]	41
Figure 2.21 CDFG TOOLKIT's IR for input VHDL code [38]	43
Figure 2.22 Sub-CDFGs for Hierarchical nodes [38]	44
Figure 2.23 GAUT's Input C file code for 4 taps FIR	45
Figure 2.24 GAUT's Intermediate Representation for DFG of 4 taps FIR	45
Figure 2.25 LegUp's Input C Code for 8 taps FIR [30]	46
Figure 2.26 LegUp's IR using LLVM for 8 taps FIR [30]	46
Figure 2.27 ASAP Scheduled DFG [16, 17]	47
Figure 2.28 ALAP scheduled DFG [16, 17]	48
Figure 2.29 List scheduled DFG [16, 17]	48
Figure 3.1 Related graphs generated by SPARK for the input C code	51
Figure 3.2 SPARK data types definitions file	52
Figure 3.3 Long and Short dependency chain	55
Figure 3.4 looping statement, loop statement on the left is not preferred	55
Figure 3.5 Main function's prototyping examples	56
Figure 3.6 Inline Functions	56
Figure 3.7 Main function on the left and associated component on right	57
Figure 3.8 Supported Expressions by GAUT	57
Figure 3.9 Input C code for 4-points FIR	59
Figure 3.10 generated DFG graph by GAUT's CDFG compiler for 4-points FIR	60

Figure 3.11 Proposed CUPSHOP v2.0 Front-End flow	61
Figure3.12 Adjacency Matrix constructed for the HCT Extracted by Eclipse IDE	64
Figure 3.13 HTC Extracted by Eclipse IDE Drawn by CUPSHOP C Front-End	65
Figure 3.14 Proposed CUPSHOP's Front-end Tool Flow.....	66
Figure 3.15 CUPSHOP's C Front-End GUI.....	70
Figure 4.1 Hierarchical Call Tree (HCT) visualized graph	72
Figure 4.2 Separate C file generated for <i>fun4</i> during input C file and HCT parsing.....	73
Figure 4.3 Generated CDFGs by GAUT's CDFG Compiler level: 2, <i>fun4</i>	73
Figure 4.4 <i>funstck</i> entry for <i>fun4</i>	74
Figure 4.5 Operation Matrix of <i>fun4</i>	75
Figure4.6 Input 'A' for Node 'mul0' at <i>fun4</i>	75
Figure 4.7 Input 'Const_5' for Node 'mul0' at <i>fun4</i>	75
Figure4.8 Output 'B' for Node 'mul0' at <i>fun4</i>	75
Figure 4.9 Separate C file generated for <i>fun3</i>	76
Figure 4.10 Generated CDFGs by GAUT's CDFG Compiler level: 2, <i>fun3</i>	76
Figure 4.11 Separate C file generated for <i>fun2</i>	77
Figure4.12 Generated CDFGs by GAUT's CDFG Compiler level: 1, <i>fun2</i>	77
Figure4.13 Separate C file generated for <i>fun1</i>	78
Figure 4.14 Generated CDFGs by GAUT's CDFG Compiler level: 1, <i>fun1</i>	78
Figure 4.15 Separate C file generated for the <i>main</i> function.....	79
Figure 4.16 Generated CDFGs by GAUT's <i>cdgcompiler</i> level: 0, <i>main</i> function	79
Figure 4.17 Hierarchical C code implementation for 16-Taps FIR.....	80
Figure 4.18 HTC for Hierarchical 16-Taps FIR	81
Figure 4.19 CDFG of function <i>Coeff_Mul</i> at level 1 of 16-Taps FIR	81
Figure 4.20 CDFG of function <i>main function</i> at level 1 of 16-Taps FIR	82
Figure 4.21 SOBEL's Edge detector pseudo-code [8].....	82
Figure 4.22HCT of hierarchical SOBEL C Code	83
Figure 4.23 CDFG of function <i>fun_01</i> at level 1 of SOBEL code	83
Figure 4.24 CDFG of function <i>fun_02</i> at level 1 of SOBEL code	84
Figure 4.25 CDFG of function <i>main</i> at level 0 of SOBEL code.....	84
Figure 4.26 Block Diagram of MJPEG baseline Decoder [6]	85
Figure 4.27 HCT for the functions of YUV2RGB Module.....	86
Figure4.28 CDFG of function <i>Saturate</i> at level 2 of YUV2RGB Module.....	86
Figure 4.29 CDFG of function <i>Fun_Mul_Shift_01</i> at level 1 of YUV2RGB Module	87
Figure4.30 CDFG of function <i>Fun_Mul_Shift_02</i> at level 1 of YUV2RGB Module	87
Figure 4.31 CDFG of function <i>Fun_Mul_Shift_03</i> at level 1 of YUV2RGB Module	88
Figure 4.32 CDFG of function <i>Fun_Mul_Shift_04</i> at level 1 of YUV2RGB Module	88
Figure 4.33 CDFG of function <i>main</i> at level 0 of YUV2RGB Module	89
Figure 4.34 Header of DCT Module provided by CHStone for JPEG	92
Figure4.35 HCT for DCT component of CHStone JPEG Benchmark	93
Figure4.36 CDFG of Function MULTCONST1 at level 2 of DCT	93
Figure 4.37 CDFG of Function MULTCONST3 at level 2 of DCT	94
Figure 4.38 CDFG of Function MULTCONST5 at level 2 of DCT	94
Figure4.39 CDFG of Function MULTCONST7 at level 2 of DCT	95
Figure 4.40 CDFG of Function Shift14Bits at level 2 of DCT.....	95
Figure4.41 CDFG of Function Shift16Bits at level 2 of DCT.....	96

Figure 4.42 CDFG of Function MULTCONST4 at level 2 of DCT	96
Figure 4.43 CDFG of Function MULTCONST2 at level 2 of DCT	97
Figure 4.44 CDFG of Function MULTCONST6 at level 2 of DCT	97
Figure 4.45 CDFG of Function fun1 at level 1 of DCT.....	98
Figure 4.46 CDFG of Function fun2 at level 1 of DCT.....	98
Figure 4.47 CDFG of Function fun3 at level 1 of DCT.....	99
Figure 4.48 CDFG of Function main at level 0 of DCT	99
Figure 4.49 Lower sub-band ADPCM, G.722 [9]	100
Figure 4.50 CHStone's ADPCM Source Code Characteristics [9]	100
Figure 4.51 Hierarchical Call Tree for CHStone's ADPCM.....	101
Figure 4.52 CDFG of Function <i>filtep</i> at level 1, CHStone's ADPCM	102
Figure 4.53 CDFG of Function <i>quantl</i> at level 1, CHStone's ADPCM	102
Figure 4.54 CDFG of Function <i>logscl</i> at level 1, CHStone's ADPCM.....	103
Figure 4.55 CDFG of Function <i>scalel</i> at level 1, CHStone's ADPCM	103
Figure 4.56 CDFG of Function <i>upplo2</i> at level 1, CHStone's ADPCM	104
Figure 4.57 CDFG of Function <i>uppol1</i> at level 1, CHStone's ADPCM	104
Figure 4.58 CDFG of Function <i>mainfunction</i> at level 0, CHStone's ADPCM	105

List of Tables

Table 2-1 HLS Tools Summary; W. Meeus' survey [8], GAUT, SPARK and LegUp....	28
Table 4-1 Brief description about codes of CHstone suite's program [84]	90
Table 4-2 Source-Level Characteristics of CHStone's suit's programs [84]	91
Table 4-3 CUPSHOP Verification Examples Results' Survey.....	107

Nomenclature

ADPCM	: Adaptive Differential Pulse Code Modulation.
ALAP	: As Late As Possible
ASAP	: As Soon As Possible
ASIC	: Application Specific Integrated Circuits
ASIP	: Application-Specific Instruction-Set Processor
CDFG	: Control-Data Flow Graph
CFG	: Control Flow Graph
CFSM	: Co-design Finite State Machine
CPU	: Central Processing Unit.
CUPSHOP	: Cairo University Particle Swarm Optimization HW/SW Partitioning.
DAG	: Directed Acyclic Graph.
DCT	: Discrete Cosine Transform.
DFG	: Data Flow Graph.
DSP	: Digital Signal Processing.
FFT	: Fast Fourier Transform.
FIR	: Finite Impulse Response.
FPGA	: Field Programmable Gate Array.
GA	: Genetic Algorithm.
GUI	: Graphical User Interface.
HCT	: Hierarchal Call Tree.
HW	: Hardware.
IR	: Intermediate Representation.
JPEG	: Joint Picture Expert Group.
KPN	: Kahn Process Networks
LE	: Logic Element.
LLVM	: Low Level Virtual Machine
LUT	: Look-Up Table.
MP3	: MPEG-1 Audio Layer 3.
MPEG	: Moving Picture Experts Group
MPSOC	: Multi-Processor Systems-on-Chip.
MUX	: Multiplexer.
OSCI	: Open SystemC Initiative
PPA	: Pipeline of Processing Arrays
PSO	: Particle Swarm Optimization.
RTL	: Register Transfer Logic.
SDF	: Synchronous Data Flow.
SoC	: System on a Chip.
SW	: Software.
TLM	: Transaction Level Models.
VHDL	: Very high speed integrated circuits Hardware Description Language.

Abstract

The first-release of Cairo University Particle Swarm optimization Hardware/software Partitioning (**CUPSHOP**) tool was previously developed at Cairo University. It accepts Behavioral, Un-Timed, Algorithmic-Level, VHDL, designs representation, with limited set of VHDL commands, and outputs a valid Hardware/ Software partition and schedule for the design subject to a set of area/ power/ delay constraints. This thesis introduces a Front-End enhancement CUPSHOP tool by enabling this tool to accept input designs written in ANSI C. The developed Front end can accept hierarchical input C codes of arbitrary number of levels. It extracts a corresponding control and data flow diagram (CDFG) representation of the system, together with all the data base information needed to define the design completely. This CDFG is fully compatible with the first release of CUPSHOP, and can also be used with several HLS and HW/SW partitioning tools.

The work done in this thesis employs several open-source tools towards building a C-input Front end for CUPSHOP. We invoke Eclipse IDE, an ANSI C editing and debugging tool, for Hierarchical Call Tree (HCT) extraction. We also invoke GAUT, an academic open source High-Level Synthesis tool for extracting Data Flow Graph (DFG) from an input design written as a flat ANSI C code. We also use the MATLAB exchange cementer library to add graph visualization capability to CUPSHOP. Finally, the developed Front-End constructs an internal database of the input design that paves the way for the HW/SW partitioning process and further, for final HDL code generation.

The main functionality of CUPSHOP's C Front-End is to convert input designs written in ANSI C language into CDFG representation. It should be highlighted that the CDFG is fully describing the input design; hence it can provide the required backbone for the HW/SW co-designs partitioning tool CUPSHOP and a High-Level synthesis tool as well.

The developed C front end is tested successfully against several published HLS benchmarks, including benchmarks from CHStone suite.