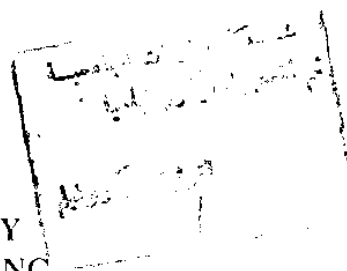


AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
Computer and Systems Engineering Department



**HARDWARE/SOFTWARE CODESIGN
METHODOLOGY FOR SOFTWARE
ACCELERATION**

A Thesis

Submitted in Partial Fulfillment for the Requirements
of the Degree of the Master of Science in Electrical Engineering
(Computer and Systems Engineering)

621.3811

M.A

Submitted By

63148

Mohamed Ahmed Shalan
B.Sc. of Electrical Engineering
(Computer and Systems Engineering)
Ain Shams University, 1993

Supervised By
Prof. Dr. Osman Badr
Dr. Hassan Shehata

Cairo -1997



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بسم الله الرحمن الرحيم

﴿ذلك فضل الله يؤتيه من يشاء و الله ذو الفضل
العظيم﴾

(الجمعة ٤)

ABSTRACT

Codesign is the Integrated design of system implementation using both hardware and software components. It is applied to tightly coupled systems with hardware and software modules that interact to solve a certain task. HW/SW Codesign differs from conventional approaches of the design in that it continuously relates the hardware development cycle to the software one. Hence decisions made at the hardware design effect software design activities and vice versa. The HW/SW codesign techniques can be used for many applications. These applications include: speedup of software execution, design of cost-effective systems, embedded systems and controllers, etc.

In our work we review the concept of HW/SW codesign, its applications and its methodologies. Then we select the speedup of software algorithms as an application and we propose a codesign methodology for it. Some tools are built and others are modified to build a framework for our proposed methodology. Also a general purpose accelerator is built, to be used for the HW part. Finally, we select a case study and we apply our methodology for it.

Key Words : *Hardware, Software, Codesign, FPGA, Synthesis, Partitioning, CPLD, Accelerator.*

ACKNOWLEDGMENT

I would like to thank Dr. Osman Badr and Dr. Hassan Shehata for their great efforts with me. Also I would like to thank my parents. Finally I would like to thank my fiancé and for her I dedicate this thesis.

Contents

1 INTRODUCTION

1.1 INTRODUCTION.....	1
1.2 APPLICATIONS OF HARDWARE-SOFTWARE CODESIGN.....	2
1.2.1 Speedup of software execution	3
1.2.2 Design of cost-effective systems	4
1.2.3 Rapid prototyping of complex system design	4
1.2.4 Embedded systems and controllers.....	5
1.2.5 Instruction-set Processors	6
1.3 THE OPPORTUNITIES OF CODESIGN	6
1.3.1 Modeling the system functionality and performance constraints.....	7
1.3.2 Choosing the granularity of the hardware-software partition	8
1.3.3 Hardware/Software partitioning	8
1.3.4. Specifying and synthesizing the hardware-software interface.	9
1.4. DESIGN METHODOLOGIES AND TECHNIQUES	10
1.4.1 Conventional Approach	10
1.4.1.1 Analysis of constraints and requirements	10
1.4.1.2 System Specification	11
1.4.1.3 Hardware/Software Partitioning	11
1.4.1.4 Hardware Synthesis and Configuration.....	11
1.4.1.5 Software Generation and Parameterization	11
1.4.1.6 Interface Synthesis	11
1.4.1.7. Hardware/Software Integration and Cosimulation	12
1.4.1.8. System Verification	13

<i>1.4.2 Model-Based Approach</i>	13
1.4.2.1 Modeling	13
1.4.2.2 System Model Base.....	14
1.4.2.3 Validation	14
1.4.2.4 Refinement	16
1.4.2.5 Simulation	16
1.4.2.6 Simulation Experiment Base	16
1.4.2.7 Technology assignment	17
1.5 THESIS OUTLINE.....	18

2 Related Works

2.1 INTRODUCTION.....	20
2.2 CAD SYSTEMS FOR H/W-S/W CODESIGN.....	20
2.3 PTOLEMY	21
2.4 CODES	25
2.5.POLIS	26
2.5.1. <i>High Level Language Translation</i>	27
2.5.2. <i>Formal Verification</i>	27
2.5.3. <i>System Co-simulation</i>	28
2.5.4. <i>Design Partitioning</i>	28
2.5.5. <i>Hardware Synthesis</i>	30
2.5.6. <i>Software Synthesis</i>	30
2.5.7. <i>Interfacing Implementation Domains</i>	31
2.6. CASTLE CODESIGN AND SYNTHESIS TOOL ENVIRONMENT.....	31
2.7. ASP.....	34
2.8 SUMMARY	36

3 HW/SW Codesign Framework

3.1 INTRODUCTION.....	38
3.2 TARGET ARCHITECTURE	39
3.2.1 <i>Hardware Accelerator</i>	41
3.2.2 <i>Hardware-Software Interface</i>	42
3.3 METHODOLOGY.....	43
3.3.1 <i>Overview</i>	44
3.3.2 <i>Specification Analysis</i>	45

3.3.2.1 Static Analysis	46
3.3.2.2 Parameter Analysis	48
3.3.2.3 Dynamic analysis (Profiling)	49
3.3.3 Partitioning.....	50
3.3.4 H/W Synthesis.....	52
3.3.5 S/W Compilation.....	53
3.3.6 Partitioning Evaluation.....	53
3.4 SUMMARY	54

4 CASE STUDY

4.1 INTRODUCTION.....	55
4.2 THE UENCODER	56
4.2.1 Algorithm:	56
4.2.2 Algorithm Analysis.....	57
4.2.2.1 Static Analysis	57
4.2.2.2 Dynamic Analysis	58
4.2.2.3 Parameter Analysis	58
4.2.3 Partitioning:	59
4.2.4 Software Compilation.....	60
4.2.5 Hardware Synthesis.....	60
4.2.6 Results	64
4.3 SUMMARY	65

5 Conclusion and Future Work

5.1 CONCLUSION.....	66
5.2 FUTURE WORK	67

A Lattice 2000/V Family Architectural Description

A.1 INTRODUCTION.....	68
A.2 GLOBAL CLOCK STRUCTURE	68
A.3 I/O CELL AND OE STRUCTURE	69
A.4 OUTPUT ROUTING POOL (ORP)	71

B Bool 2

B.1 OVERVIEW	74
B.2 LANGUAGE SPECIFICATION	75
<i>B.2.1 TYPES OF VARIABLES</i>	75
<i>B.2.2 EXPRESSIONS AND OPERATORS</i>	76
<i>B.2.3 CONTROL FLOW STATEMENTS</i>	76
<i>B.2.4 STRUCTURED CODE</i>	77
<i>B.2.5 PREPROCESSOR</i>	77
B.3 THE MINIMIZER FACILITY	77
B.4 THE SIMULATOR FACILITY	78
B.5 THE I/O FACILITY	78

References