ACCELERATED VERILOG SIMULATOR USING APPLICATION SPECIFIC MICROPROCESSOR

TAN TZE SIN

UNIVERSITI SAINS MALAYSIA

2017

ACCELERATED VERILOG SIMULATOR USING APPLICATION SPECIFIC MICROPROCESSOR

by

TAN TZE SIN

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

May 2017

ACKNOWLEDGEMENTS

I would like to express my gratitude to Associate Professor Dr Bakhtiar Affendi Bin Rosdi for willing to supervise my work in this research program. His experience has proven valuable in guiding the success of this endeavor. Through countless meetings and discussions over the years, he supported and encouraged the progress of this project.

Being a collaborative research program of Universiti Sains Malaysia (USM) and Altera (now part of Intel), I like to thank the management team in Altera for their understandings and supports in developing my skill set through this opportunity. My thanks go to the management team including Dato' Dr. Mohamad Sofi, Man On, Siew Ling, Vincent, and others. This research was funded in part by Malaysian Ministry of Higher Education (MOHE) via MyBrain15 program. I would like to thank MOHE for offering me the opportunity to participate in Industry PhD program.

My warmest feeling goes to my wife, Chain Yun, for her willingness to bear with me over the loss times. To my children, your smiles reminded me to move forward during the toughest time. Not to forget my beloved parents, brother, and sister for their encouragements. My appreciation goes to friends, colleagues, and many others who have been supportive over the duration. Thank you all!

TABLE OF CONTENTS

LIST OF PUBLICATIONS

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF PLATES

LIST OF ABBREVIATIONS

ALM	Adaptive Logic Module
ALU	Arithmetic Logic Unit
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
ASIP	Application Specific Instruction-Set Processor
ATPG	Automatic Test Pattern Generation
BIST	Built-In Self Test
CAD	Computer Aided Design
CISC	Complex Instruction Set Computing
CPU	Central Processing Unit
DDR	Double Data Rate
DFT	Design For Test
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processor
ECC	Error Correction Code
EDA	Electronic Design Automation
FF	Flip-flop

FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
GLN	Gate Level Netlist
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HBM	High Bandwidth Memory
HDD	Hard Disk Drive
HDL	Hardware Description Language
IC	Integrated Circuit
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IO	Input Output
IP	Intellectual Property
LAN	Local Area Network
LE	Logic Element
LSB	Least Significant Bit
MAC	Media Access Controller
MOHE	Malaysian Ministry of Higher Education

MSB	Most Significant Bit
Mutex	Mutual Exclusive
NCP	Native Code Processor
Opcode	Operation Code
OS	Operating System
OSI	Open Systems Interconnection
PAL	Programmable Array Logic
PC	Personal Computer
PDES	Parallel Discrete Event Simulator
PHY	Physical Layer
PLA	Programmable Logic Array
PLL	Phase Locked Loop
RAM	Random Access Memory
RHS	Right Hand Side
RTL	Register Transfer Level
SDF	Standard Delay Format
SDRAM	Synchronous Dynamic Random Access Memory
SEU	Single Event Upset
SIMD	Single Instruction Multiple Data

SoC	System On-Chip
SRAM	Static Random Access Memory
SPICE	Simulation Program with Integrated Circuit Emphasis
SRAM	Static Random Access Memory
SSD	Solid State Drive
STA	Static Timing Analysis
Tcl	Tool Command Language
TCP	Transmission Control Protocol
UDP	User Defined Primitives
USB	Universal Serial Bus
USM	Universiti Sains Malaysia
UVM	Universal Verification Methodology
VCD	Value Change Dump
VHDL	Very High-Speed Description Language
VLSI	Very Large Scale Integration

LIST OF SYMBOLS

$Chit$	Cache hit rate
$\in \{\}$	Element in the set
$f()$	Simulation transfer function
i_{last}	Last iteration in the simulation time step
$m_{i,t}$	State variable of simulation model where i is event index and t is time
N_{core}	Number of *VerCPU* core
$Ntran$	Number of transaction
$n_{i,t}$	Input stimuli to simulation model where i is event index and t is time
$\sum$	Summation
T_{1core}	Simulation time for single core prototype (VERSYS_1CORE)
T_{cache}	Simulation time for cached quad-core prototype (VERSYS_CACHE)
T_{ch}	Cache access latency
T_{multi}	Simulation time for quad-core prototype (VERSYS_MCORE)
$Tmem$	Memory access latency
$Tproc$	Transaction processing time
t_{last}	Last simulation time

PENYELAKU VERILOG TERPECUT MENGGUNAKAN MIKRO-PEMPROSES APLIKASI KHUSUS

ABSTRAK

Simulasi logik merupakan satu langkah penting dalam pembangunan Litar Bersepadu Skala Sangat Besar (VLSI IC). Kemajuan Bahasa Takrifan Perkakasan (HDL) menjadikan Verilog salah satu pengantara dominan yang digunakan untuk pemodelan litar berdigit dan kes ujiannya. Pembekal Perisian Rekabentuk Elektronik (EDA) ada menawarkan perisian dan juga perkakasan untuk tujuan simulasi. Walau bagaimanapun, perisian simulasi adalah perlahan manakala simulasi bantuan perkakasan tidak menawarkan cirian simulasi sebagai mana yang ditetapkan dalam Verilog. Dalam projek penyelidikan ini, sebuah perkakasan simulasi yang berupaya menjalankan simulasi Verilog dicadangkan iaitu Sistem *VerCPU* untuk menangani kelemahan pelantar sedia ada. Pemproses *VerCPU* merupakan mikro-pemproses aplikasi khusus yang direka untuk menjalankan simulasi Verilog. Pemproses ini melakukan hitungan data Verilog dalam bentuk asalnya di samping menyokong aliran program acara selari untuk mencapai kelajuan yang unggul. Sebagai pelantar simulasi yang berasaskan kod, cirian keluaran dibekalkan dengan memberikan keputusan dan penelitian yang sama seperti perisian simulasi. Sebuah sistem prototaip yang lengkap berfungsi, *VerCPU* dibina di atas papan pembangunan Litar Aturcara Dalam Bidang (FPGA). Sistem ini berjaya disahkan dan dibandingkan dengan sebuah perisian simulasi sedia ada iaitu Synopsys VCS®. Sistem *VerCPU* berupaya mencapai kelajuan sehingga 6 kali ganda walaupun dengan hanya menggunakan teknik-teknik asas untuk mempercepatkan penilaian. Fungsi sistem ini sudah disahkan berkesan sebagai cara

simulasi Verilog alternatif yang berupaya memenuhi keperluan simulasi masa depan.

ACCELERATED VERILOG SIMULATOR USING APPLICATION SPECIFIC MICROPROCESSOR

ABSTRACT

Logic simulation is an important step in Very Large Scale Integration (VLSI) IC development. Advancement in Hardware Description Language (HDL) has made Verilog a widely adopted language used to model digital circuit and verification test bench. Electronic Design Automation (EDA) vendor provides software and hardware-assisted approaches to carry out simulations. However, software-based simulator is slow whereas hardware-assisted simulator does not offer the same simulation fidelity stipulated in Verilog. In this research project, a hardware-assisted Verilog simulator, *VerCPU* System, was proposed to address shortcomings in existing platforms. The simulator core is a custom designed application specific microprocessor specifically adapted to handle Verilog simulation. The microprocessor computes Verilog data in its native form while supporting event-driven parallelism directly to achieve speed supremacy. Being a compiled-code simulator, simulation fidelity compliancy is retained to offer the same result and visibility like the software-based solution. A functional system, *VerCPU*, was developed and prototyped on a Field Programmable Gate Array (FPGA) development board. This system was successfully verified and benchmarked against a software-based compiled-code simulator, i.e. Synopsys VCS®. *VerCPU* System can already achieve up to 6 times speed superiority with basic speed improvement techniques applied. The simulator had proven to be a viable alternate Verilog simulator to meet future simulation needs.

CHAPTER ONE

INTRODUCTION

1.1 Background

Very Large Scale Integration (VLSI) design complexity has grown exponentially since early 1970s. Circuit with billions of transistors in an Integrated Circuit (IC) has started to become the norm (Arden et al., 2010). Engineering cost has grown exponentially as fabrication process node shrinks. Hence, iterations of design change must be minimized. Manual design and analysis of circuit are no longer practical. Instead, a large proportion of design activities are assisted by Computer Aided Design (CAD) tools. Thorough verifications are carried out before a design is handed over for fabrication. Early circuit design was schematic entry. This has become impractical as circuit complexity increases. Text-based Hardware Description Language (HDL) was developed in mid-1980s (Cavanagh, 2007). HDL is now the dominant design entry for multi-million gate digital VLSI. The language is capable of modeling a circuit at different abstraction levels aligned with design maturity at different development stages.

A typical digital IC design flow is summarized in Figure 1.1. Some of the steps are iterative until convergence is achieved. Specification of feature set is the first step. High level behavioral model is suitable at this exploration stage where the focus being functional transformation algorithm of inputs into outputs. Viability and competency of the design are studied. Model is created based on the specifications to select the optimum design parameters that can offer the required feature set and performance. This is the earliest step where HDL can be applied to represent a circuit

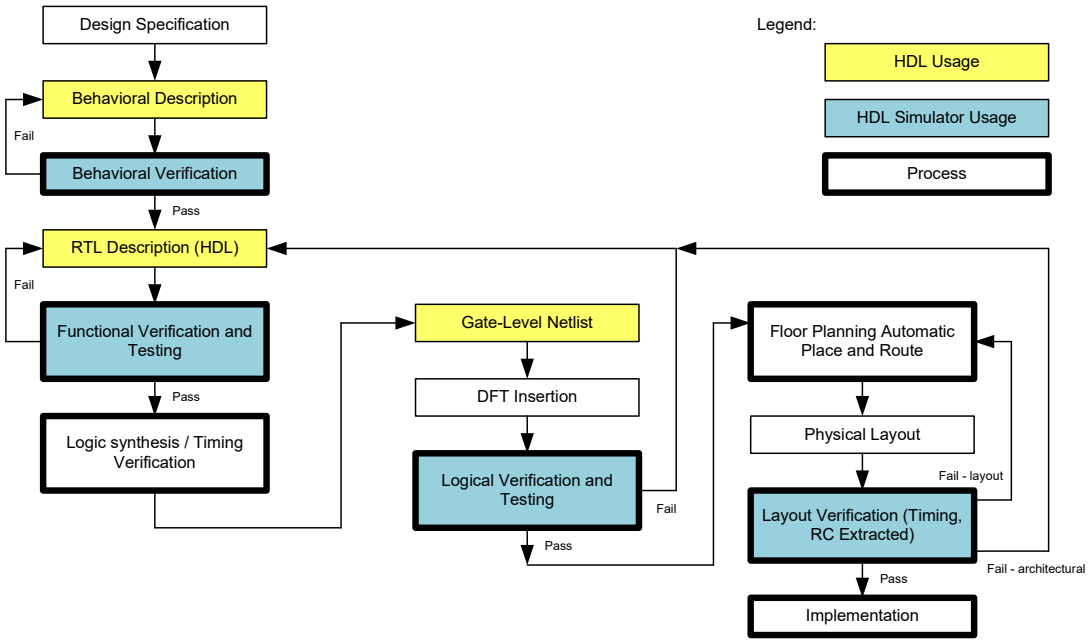

Figure 1.1: Typical digital VLSI design flow (Palnitkar, 2003)

model. The model is expressed in behavioral code and simulated on HDL compliant simulator. Once key parameters are finalized, the design is partitioned into manageable sub-systems for Register Transfer Level (RTL) coding. The codes are restricted to synthesizable Verilog content offered by the chosen technology. At this stage, data flow through circuit level implementation of the design is coded.

Next, RTL is synthesized into connections of primitive components offered by the process technology. The primitive components include but not limited to logic gate, flip-flop, memory, Phase Locked Loop (PLL), etc. Transformation is carried out by automated synthesis tools. Timing constraints are provided to the tool in order to optimize the circuit for required performance. This is accomplished through physical layout aware component placement and driver strength adjustment. Output from synthesis process is a circuit model with structural connected primitive components known as Gate Level Netlist (GLN).