Ain Shams University

Faculty of Engineering

Computer and Systems Engineering Department

On the Hardware Architecture for Satisfiability Problems

A Thesis

Submitted in partial fulfillment for the requirements of the degree of Doctor of Philosophy in Electrical Engineering

Submitted by:

Mona Mohamed Hassan Safar

M.Sc. of Electrical Engineering

(Computer and Systems Department)

Ain Shams University, 2007.

Supervised by:

Prof. Dr. Ashraf Salem

Dr. Mohamed Watheq El-Kharashi

Dr. Mohamed Shalan

Cairo 2011

Abstract

The Boolean Satisfiability problem (SAT) is a central problem in artificial intelligence, mathematical logic, and computing theory with wide range of practical applications in Electronic Design Automation. Several approaches have been proposed to accelerate the NP-complete SAT using reconfigurable computing. Much of the performance improvement achieved by state-of-the-art software SAT solvers is related to the implementation of conflict analysis, which enables the solver to perform non-chronological backtracking and conflict driven learning. Such advanced techniques have been ignored by the majority of hardware SAT solvers or are executed on some coupled software running on an attached host processor.

In this thesis, we propose a new conflict directed search algorithm, best suited for a reconfigurable hardware implementation. The algorithm performs conflict analysis and hence nonchronological backtracking. The algorithm enables learning without explicit new clauses addition avoiding consumption of new hardware resources.

We present a pipelined SAT solver in which the execution of the proposed algorithm is divided into five stages with all stages executed in hardware without any communication with a host processor. The pipelined architecture provides significant speedup while retaining the same clock frequency of an equivalent non-pipelined implementation. Performance is improved by increasing the throughput of assigning SAT variables while evaluating and checking the validity of the formula with each new variable assignment. All possible data and control hazards are handled.

For different SAT problem instances, SAT instance's specific data is stored only in memory modules. A memory module stores the instance's clauses and another memory module stores the effect of the assignment of each of the instance's variables on the clauses. No instance-specific circuit is employed. Mapping different SAT problem instances into the proposed reconfigurable SAT solver requires only reloading those memory modules eliminating compilation, synthesis, and place-and-route overhead. This enables achieving real speedup compared to current state-of-the-art software SAT solvers.

Finally, we presented a new approach for certifying the solver's output. Based on the latest explored search space and the last encountered conflict clauses, the solver produces a refutation proof assuring that no non-redundant search space has been erroneously pruned

and that no hardware malfunction has occurred.

We compared our SAT solver with other hardware SAT solvers through instances from DIMACS benchmarks suite. The simplicity of our architecture enables achieving higher clock rates and fewer resources utilization.

Acknowledgements

I was honored to have Prof. Dr. Rolf Drechsler on the examination committee of my doctoral dissertation. I am immensely grateful for his consideration. I was thrilled with his kind announcement awarding me the doctorate degree.

I am cordially grateful to Prof. Dr. Ayman Wahba for his interest in my work and being a member of the examination committee.

My sincerest gratitude and appreciation goes to Prof. Dr. Ashraf Salem for his kind supervision, endless patience, precise advice, exceptional guidance, sincere encouragement, and insightful thoughts. I have experienced a tremendous personal and professional growth under his supervision.

I am deeply indebted to Dr. Mohamed Watheq El-Kharashi for his constant support, concern, dedication to research, invaluable help, and attention to detail throughout all phases of this work.

I wish to express my gratitude to my supervisor Dr. Mohamed Shalan for his kind help, insight, fruitful discussions, and constructive criticism.

I address my great thanks to my mother for her tireless sacrifice, trust, love, and cheerful encouragement. I am in no way capable of appropriately thanking my father for his unconditional love, care, and non-fading support. Heartfelt thanks goes to my brother and my sister for their love and support. Finally, words are not enough to express the thanks and gratitude I owe to my husband for his continuous encouragement and support.

Mona Mohamed Hassan Safar
Computer and Systems Engineering Department
Faculty of Engineering, Ain Shams University
Cairo, 2011

Statement

This thesis is submitted to Ain Shams University for the degree of Doctor of Philosophy in Electrical Engineering (Computer and Systems).

The work included in this thesis was carried out by the author at Computer and Systems Engineering Department, Ain Shams University.

No part of this thesis has been submitted for a degree or qualification at any other university or institute.

Mona Mohamed Hassan Safar
Computer and Systems Engineering Department
Faculty of Engineering
Ain Shams University
Cairo, Egypt
2011

Contents

List of Tables

List of Figures

List of Abbreviations

ALU	Arithmetic and Logic Unit
ASIC	Application-Specific Integrated Circuit
ASIP	Application-Specific Instruction set Processor
ATPG	Automatic Test Pattern Generation
BCP	Boolean Constraint Propagation
BRAM	Block Random Access Memory
CA	Conflict Analysis
CAD	Computer-Aided Design
CD	Conflict Detection
CDB	Conflict Directed Backjumping
CDCL	Conflict Driven Clause Learning
CDJ	Conflict Directed Jumping
CE	Clause Evaluation
CEC	Combinational Equivalence Checking
CJS	Conflict Jump Set
CLB	Configurable Logic Block
CNF	Conjunctive Normal Form
CSP	Constraint Satisfaction Problem
DAG	Directed Acyclic Graph
DIMACS	Center for Discrete Mathematics and Theoretical Computer Science
DP	Davis and Putnam
DPLL	Davis, Putnam, Logemann, and Loveland
DV	Decision Variable
DDV	Dummy Decision Variable
EDA	Electronic Design Automation
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine

FV	Free Variable
GPU	Graphic Processing Unit
GRASP	Generic seaRch Algorithm for the Satisfiability Problem
HDL	Hardware Description Language
I/O	Input/Output
JS	Jump Set
JV	Jump Variable
LUT	Look-Up Table
MUX	Multiplexer
NFV	Next Free Variable
PE	Priority Encoder
PCI	Peripheral Component Interconnect
PLD	Programmable Logic Device
PODEM	Path-Oriented DEcision Making
PoS	Product-of-Sums
RAM	Random Access Memory
RUP	Reverse Unit Propagation
SAT	Boolean Satisfiability
VD	Variable Decision
VEOC	Variable Effect On Clause
VF	Variable Fetch
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit