



وَقَالَ رَبِّ نِزِّلْنِي



Computer Systems Department
Faculty of Computer & Information Sciences
Ain Shams University

Performance Evaluation of DNA-based Self-Assembled Computers

A thesis submitted as a partial fulfillment of the requirements for the degree
of Master of Science in Computer and Information Sciences.

By

Ramy Medhat Mohamed Abdelaal

B.Sc. in Computer & Information Sciences

Demonstrator at Computer Systems Department

Faculty of Computer and Information Sciences, Ain Shams University

Under supervision of

Prof. Dr. Mohamed Essam Khalifa

Former Dean of the Faculty of Computer and Information Sciences

Ain Shams University

Prof. Dr. Hossam Faheem

Head of the Computer Systems Department

Faculty of Computer and Information Sciences, Ain Shams University

Cairo, Egypt 2011

Acknowledgement

All praise to Allah, the gracious and merciful, for helping me complete this work.

No words can describe my gratitude towards my parents, pushing me forward every step of the way. I could not have asked for a more supportive mother, doing everything she can to help. My father put no limits to the sacrifices he was willing to make to see me succeed. My sister is the greatest sister a man could have, always putting me first. To them I owe the accomplishment of this work, and to them I put it forward as the fruit of their sacrifices.

I also present my sincerest thanks to my thesis advisors: Prof. Dr. Essam Khalifa, and Prof. Dr. Hossam Faheem. Prof. Essam put me on the path after months of struggling to pinpoint a research area, and Prof. Hossam has provided me with immeasurable support and invaluable advice throughout the work. I thank him for the hours and hours spent brainstorming our way over obstacles. I'm lucky to have had such supportive advisors.

I put forward my sincerest appreciation of the help of Prof. Chris Dwyer. This work would not have been completed without his great support and prompt assistance.

I sincerely appreciate the help of Prof. Dr. Samy ElGhoneimy, supporting me with great and insightful advice. I thank Mahmoud Fayeze for his selfless help, and all my colleagues for their limitless support and encouragement.

Abstract

Silicon-based technology has in no doubt played the pivotal role in our civilization. Our lives have dramatically changed in every way, our productivity has increased, and our quality of life has advanced beyond imagination, and this all was majorly due to the discovery of semiconductors. Decades have passed, and we now have at our disposal advanced electronics as a product of a very mature technology. The advances in hardware capabilities have been taking steady steps for years, and our fabrication techniques have matured into stable low-error processes.

However, the natural evolution of technology is inevitable, and this is what the silicon based technology will face sooner or later. Moore's law has been accurately applied steadily since its inception, but the core of the law cannot be indefinitely applicable. Although transistor sizes have decreased dramatically since their invention, with the 50% decrease rate Moore postulated, transistors are now approaching atomic level. The unavailability of fabrication technology at this scale, the instability at this resolution, and the increased error rate led to the conclusion that we need a replacement for silicon-based technology soon.

Various technologies have been proposed to replace silicon-based semiconductors. Carbon nanotubes are among the more prominent replacements due to their advantages in speed, power and size. The challenge is how to assemble carbon nanotubes to form a circuit. Our current top-down approach used in photolithography is not capable of assembling devices with that small size, and thus an

alternative approach to the whole fabrication methodology was researched.

Self-assembly relies on the bottom-up construction of the circuit, without the interference of assembling equipment. The concept is similar to how biological cells assemble autonomously to form a complex organ. Yet in the nanoscale DNA was used to assemble carbon nanotubes; an autonomous process without our interference.

By leveraging the unique structure of DNA and its ability to form shapes through controlling the sequences of a DNA strand, a lattice was built autonomously using DNA. The lattice will act as the scaffold on which the carbon nanotubes are assembled to form a circuit.

This thesis presents a novel parallel architecture to perform realtime median filtering of a stream of images. The architecture is then put through the design flow of DNA-based self-assembled circuits to produce the final output, consisting of a set of DNA sequences and their assembly order. The carbon nanotube based circuit is simulated to produce power and timing results and these results are compared to the CMOS implementation of the architecture. The advantages of using carbon nanotubes in both power and delay are demonstrated. The tools developed to aid in the design of self assembled circuits are presented.

Table of Contents

List of Figures	V
List of Tables	X
List of Abbreviations	XII
Chapter 1: Introduction.....	3
1.1 Overview	3
1.2 DNA Self-Assembled Design.....	6
1.3 Previous Work.....	8
1.4 Motivation.....	9
1.5 Objectives	9
1.6 Thesis Contributions	10
1.7 Thesis Outline.....	11
Chapter 2: Principles of DNA Self Assembly.....	13
2.1 Introduction.....	13
2.2 Deoxyribonucleic acid (DNA)	13
2.3 Approaches to Self-Assembly.....	21
2.4 Building Addressable Grids	23
2.5 Metrics and Design Rules.....	25

Table of Contents

2.6	Sequence Selection Methods.....	27
2.7	Cross-Melt	28
2.8	Hierarchical Assembly	29
2.9	DNA Grids as Scaffolds	29
2.10	DNA-based Circuit Building Blocks.....	31
2.11	Construction of a NAND Gate	34
2.12	Case Studies	41
2.13	Design Flow	42
2.14	Final Output	43
2.15	Summary.....	44
Chapter 3: Median Filter Architecture		46
3.1	Introduction.....	46
3.2	Background	47
3.3	Previous work.....	48
3.4	The proposed architecture	52
3.5	Implementation & Simulation	59
3.6	Further Enhancements	65
3.7	Summary.....	67

Table of Contents

Chapter 4:	DNA-based	Self-Assembled	Median	Filter
Architecture	69			
4.1	Introduction.....	69		
4.2	Augmenting the architecture for DNA	69		
4.3	Transistor-based Implementation	71		
4.4	Manual Layout Tool	75		
4.5	3D Rendering Tool	76		
4.6	Simulation and comparison with CMOS	76		
4.7	Optimum Sequence Set Extraction.....	77		
Chapter 5:	Results.....	103		
5.1	Introduction.....	103		
5.2	CMOS vs. CNFET Comparison	106		
5.3	Final Clock Results	111		
5.4	Discussion	112		
5.5	Summary.....	113		
Chapter 6:	Conclusion	115		
6.1	Advantages of using DNA-based Self Assembly	115		
6.2	Challenges	116		
6.3	Future Work.....	118		

Table of Contents

References..... 122

Appendix A: Transistor Netlist Synthesis..... 128

Appendix B: Processing Element Netlist..... 135

Appendix C: Manual Layout Tool 156

Appendix D: 3D Rendering Tool 160

Appendix E: OpenCL Sequence Set Selection..... 162

List of Figures

Fig. 2-1: DNA.....	13
Fig. 2-2: Chemical structure of DNA. Base-pairs and their backbones.....	14
Fig. 2-3: Top, a GC base pair. Bottom, an AT base pair.	16
Fig. 2-4: Schematic of a cruciform motif [8].....	20
Fig. 2-5: Two fully addressable DNA grids. Top: a 10-tile structure. Bottom: a 16-tile structure spelling the word DNA. [8]	23
Fig. 2-6: Hierarchical approach to build grids from separate strands.....	24
Fig. 2-7: 3D front view of the top layer of a DNA grid cell.	32
Fig. 2-8: 3D back view of the top layer of a DNA grid cell.	32
Fig. 2-9: 3D front view of the bottom layer of a DNA grid cell.	33
Fig. 2-10: 3D back view of the bottom layer a DNA grid cell.	33
Fig. 2-11: CMOS NAND.	34

List of Figures

Fig. 2-12: Assembly order of cavities to form the grid [9].....	35
Fig. 2-13: NAND 2D layout [9].	36
Fig. 2-14: 3D Rendering of CNFET NAND, top view.	40
Fig. 2-15: 3D Rendering of CNFET NAND, bottom view.	40
Fig. 2-16: The implemented design flow based on the design flow in [9].....	42
Fig. 3-1: A sample histogram. The first value in each column is the actual frequency and the second is the cumulative frequency. The cut-off line is at the center of the window, and thus in the example, pixel value 3 is the median of that window since its cumulative frequency is the first to surpass 5: the median index.	49
Fig. 3-2: A sliding window in operation. The pixel at (3,3) is the first pixel to calculate. Its window is from (1,1) to (5,5). When sliding the window to the next pixel (3,4), the column on the right ((1,6) to (5,6)) is added to the histogram and the column on the left ((1,1) to (5,1)) is removed from the histogram.....	50
Fig. 3-3: Vertically sliding window. All calculations are reused.....	51
Fig. 3-4: Overview of the proposed architecture.	52

List of Figures

Fig. 3-5: Layout of the histogram	53
Fig. 3-6: Block diagram of a processing element.....	56
Fig. 3-7: Control Unit Mode logic. A = Mode(1), B = Mode(0).	59
Fig. 3-8: Control flow in the control unit. The diagram shows the conditions that cause a transition from one mode of operation to another.	59
Fig. 3-9: Waveform of processing element simulation.	62
Fig. 3-10: Floorplan of processing element on FPGA.	63
Fig. 3-11: RTL schematic of the processing element.	64
Fig. 3-12: Block diagram of the system using the output bus.	65
Fig. 3-13: The shadow processing elements (SPEs) used in the system.	67
Fig. 4-1: Synthesis layouts against the ASIC Design Kit. (a) RTL Schematic. (b) Technology Schematic.	72
Fig. 4-2: Histogram loaded in Design Architect.	73
Fig. 4-3: Row Buffer loaded in Design Architect.	73
Fig. 4-4: Processing Element loaded into Design Architect.....	74
Fig. 4-5: Screenshot of the manual layout tool. The	

List of Figures

schematic is of a two-input NAND.	75
Fig. 4-6: The hierarchy of memory in an OpenCL device. [32].....	83
Fig. 4-7: Pseudocode for TLM calculation.....	87
Fig. 4-8: Coalesced load from global memory. The diagram assumes we have 4 sequences (A-D), each of length two characters. Rounds indicate the iterations of coalescing loads from global memory to local memory.	89
Fig. 4-9: Code segment for Optimized TLM, using shared memory.	91
Fig. 4-10: Results from Configuration 1. OpenCL 1 represents the implementation without shared memory optimizations, and OpenCL 2 represents the implementation using shared memory.	94
Fig. 4-11: Results from Configuration 2.....	94
Fig. 4-12: OpenCL 1 on CPU performing only marginally slower than the OpenCL 2, and much faster than OpenMP.....	96
Fig. 4-13: Global memory use vs. constant memory use.	97
Fig. 4-14: Binary Melting results on the higher-end configuration	97

List of Figures

Fig. 4-15: OpenCL CPU vs OpenMP for binary melting.	99
Fig. 4-16: Comparing native vs. non-native implementations.	100
Fig. 4-17: Comparing normal loops vs. loop unrolling.....	100
Fig. 5-1: Gate propagation delay.	105
Fig. 5-2: Plots of the output waveforms of a) Two input NAND, b) Full Adder, and c) D Flip-flop. CNT and CMOS responses demonstrated. Time is in nanoseconds.	109
Fig. 5-3: Plots of the output waveforms of a) 4x16 Cumulative Decoder, and b) Histogram Register. CNT and CMOS responses demonstrated. Time is in nanoseconds.	111