

ANASIC DESIGN AND TEST METHODOLOGY FOR AN UNDERGRADUATE
DESIGN AND FABRICATION PROJECT

Arun Joseph Kurian, MSCE

Department of Electrical and Computer Engineering

APPROVED:

Eric MacDonald, Ph.D.

John Moya, Ph.D.

Ryan B. Wicker, Ph.D.

Benjamin C. Flores.
Dean of the Graduate School

Copyright ©

by

Arun Joseph Kurian

2012

Dedication

To My Parents and Friends

PREVIEW

AN ASIC DESIGN AND TEST METHODOLOGY FOR AN
UNDERGRADUATE DESIGN AND FABRICATION PROJECT

by

Arun Joseph Kurian, MSCE

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

DECEMBER 2012

UMI Number: 1533234

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1533234

Published by ProQuest LLC (2013). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ACKNOWLEDGEMENTS

I would like to thank my graduate thesis advisor Dr. Eric MacDonald for giving me the opportunity to work in the field I am most passionate about: Digital Circuit Design. I could have never asked for a better major advisor. Dr. MacDonald has not only expanded my skills in VLSI area but has elevated my confidence to do well in my professional endeavors. Thank you Dr. MacDonald, I will always be grateful for everything you have done for me.

Also, I would like to extend my most humble gratitude to my team in Intel Folsom, California who gave me an opportunity for internship. This was a great motivating experience to complete my studies and get back to working again.

Additionally, I wish to give my sincere thanks to members of UTEP's ASICS Lab for their support and interest in this research. In particular, I would like to thank Praveen Palakurthi not only for his welcomed suggestions and endless assistance, but for making me feel I was not alone in the intricate but rewarding field of Digital Circuit Design.

I am also compelled to recognize that without the assistance of the UNIX system administrators and the staff from the ECE Dept. I would have never completed my work.

To Dr. Moya, I am obliged for his academic instruction, for his professional advice, and for motivating me to never quit. To Dr. Wicker I am truly grateful for taking a sincere interest in my thesis topic and for highlighting areas of improvement.

Finally, I would like to thank my family and friends who have always believed that I can accomplish my goals. My parents, and my brother Tharun are everything to me and I would never be here, without their support.

ABSTRACT

During the 1990's the main focus of chip design methodologies was on the timings and area constraints. Power consumption was considered significant only after a drastic increase of device densities from 130nm on as well as dramatic increases in sub threshold leakage. As technology advanced from 130nm to 90nm and below there was a significant increase in leakage current due to lower threshold voltage and the influence of the deep submicron effects. High power consumption causes different problems such as increasing the cost of the product, reducing the reliability, reducing the battery life among others. Therefore EDA tools were designed to maximize the speed while minimizing area and only recently focused on improving power.

The main objective of this thesis is to complete a study of an ASIC (Application Specific Integrated Circuit) design and test flow to establish a full design methodology for an undergraduate class chip design and fabrication project from Verilog RTL to GDS2 for fabrication. The tools include Synopsys Design Compiler to generate a netlist of the physical design and Synopsys IC Compiler to perform the placement and optimization followed by clock tree synthesis, routing and lastly corechecking. The core is then inserted and connected with the chip pad frame using Synopsys Custom Designer. The final chip GDS generated will be sent to Mosis for fabrication. The Verification of the final chip design will be done using Cadence Virtuoso. This project gives an overview of different steps in the development of an ASIC, front end and back end design using Synopsys Design Compiler and IC compiler flow. In this thesis a simple 8 bit counter is considered as an example.

This Thesis will provide the students with familiarity with the current industry standard tools from vendors like Synopsys and Cadence and the students will be well versed with a comprehensive ASIC design flow. The final design will be sent to Mosis for fabrication and the student teams will

have working silicon in their hands with five packaged chip per project the demonstration of which will be beneficial when interviewing for a job in the chip industry.

PREVIEW

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	V
ABSTRACT	VI
TABLE OF CONTENTS	VIII
LIST OF FIGURES	X
CHAPTER 1: INTRODUCTION	1
1.1 WHAT IS AN ASIC?	1
1.2 TYPES OF ASICs:	1
1.2.1 Full Custom ASICs:	2
1.2.2 Semi-Custom ASICs:	3
1.2.3 Standard Cell ASICs:	3
1.2.4 Gate Array ASICs:	3
1.2.5 Programmable ASICs:	5
1.3 CHAPTER ORGANIZATION	7
CHAPTER 2: BACKGROUND AND MOTIVATION	8
2.1 DIGITAL CIRCUIT DESIGN:	8
2.3 DIGITAL LOGIC BASICS	9
2.4 IMPLEMENTATION OF GATES	9
2.5 PROPAGATION DELAYS:	10
2.6 POWER CONSUMPTION IN DIGITAL CIRCUITS	11
2.7 LOW POWER SYNTHESIS:	12
2.8 PLACEMENT ALGORITHMS:	13
2.9 STATIC TIMING ANALYSIS:	13
2.10 POWER ESTIMATION AND REDUCTION:	14
CHAPTER 3: LOGIC SYNTHESIS	15
3.1 IMPORTANT DC PARAMETERS:	16
3.2 DESIGN OBJECTS:	17
3.3 DESIGN ENTRY	18
3.4 TECHNOLOGY LIBRARY:	19
3.5 DESIGN ATTRIBUTES AND CONSTRAINTS:	19
3.5.1 Design Attributes:	20
3.5.2 Design Constraints:	20
CHAPTER 4: SYNTHESIS OPTIMIZATION TECHNIQUES	24
4.1 MODEL OPTIMIZATION:	24
4.1.1 Resource Allocation:	24
4.1.2 Common sub-expressions and Common factoring	26

4.1.3	Removing Redundant Code	27
4.1.4	Constant folding and Dead code elimination:	28
4.1.5	Flip-flop and Latch optimizations:	28
4.1.6	Using Parentheses:	28
4.2	OPTIMIZATIONS USING DESIGN COMPILER:	30
4.2.1	Compile the design	30
4.2.2	Flattening and structuring.....	31
4.2.3	Removing hierarchy	32
4.2.4	Optimizing for Area	33
4.3	TIMING ISSUES:	33
4.3.1	Compilation with map_effort high.....	35
4.3.2	Group critical paths and assign a weight factor.....	35
4.3.3	Register balancing	35
4.3.4	Choose a specific implementation for a module	36
4.3.5	Balancing heavy loading Designs.....	36
CHAPTER 5: IC COMPILER.....		37
5.1	FLOOR PLANNING:.....	38
5.2	CONCEPT OF FLATTENED VERILOG NETLIST.....	42
5.3	PLACEMENT.....	45
5.4	CLOCK TREE SYNTHESIS	46
5.5	ROUTING.....	49
CHAPTER 6: FULL METHODOLOGY.....		51
6.1	DESIGN COMPILER.....	51
6.2	IC_COMPILER:	52
6.3	CUSTOM DESIGNER:	61
CHAPTER 7: CONCLUSION AND FUTURE WORK.....		75
REFERENCES.....		76
APPENDIX B SYNTHESIS TCL SCRIPT		80
APPENDIX C PLACE AND ROUTE TCL SCRIPT		82
VITA.....		87

LIST OF FIGURES

FIGURE 1: TYPES OF ASICS	2
FIGURE 2: STANDARD CELL BASED ASIC.....	3
FIGURE 3: GATE ARRAY ASIC	4
FIGURE 4: PLD	5
FIGURE 5: GATE TRUTH TABLE	9
FIGURE 6: NOR GATE USING TRANSISTORS.....	10
FIGURE 7: SYNTHESIS FLOW.....	16
FIGURE 8: BAD RESOURCE ALLOCATION	25
FIGURE 9: GOOD RESOURCE ALLOCATION	26
FIGURE 10: SYNTHESIS WITHOUT PARENTHESIS	29
FIGURE 11: SYNTHESIS WITH PARENTHESIS	29
FIGURE 12: SETUP AND HOLD TIMES.....	34
FIGURE 13: ASIC DESIGN FLOWCHART.....	37
FIGURE 14: FLOOR PLAN EXAMPLE.....	39
FIGURE 15: FLOOR PLANNING FLOWCHART	45
FIGURE 16: CLOCK TREE SYNTHESIS STRATEGY	47
FIGURE 17: CLOCK DISTRIBUTION LEVELS	47
FIGURE 18 : WAVEFORM 1	48
FIGURE 19: FLOOR PARTITIONED INTO GCELLS	50
FIGURE 20: DESIGN COMPILER INVOKE.....	51
FIGURE 21: IC COMPILER DESIGN IMPORT	52
FIGURE 22: FLOOR PLAN CREATION.....	53
FIGURE 23: POWER RING CREATION	54
FIGURE 24: : METAL LAYERS.....	55
FIGURE 25: PLACEMENT OF CELLS	56
FIGURE 26: ADDING POWER STRAPS	57
FIGURE 27: INSERTING CLOCK TREE.....	58
FIGURE 28: ROUTING	59
FIGURE 29: ADDING FILLS	60
FIGURE 30: CREATING NEW LIBRARY.....	61
FIGURE 31: COPYING CONTENTS TO NEWLY CREATED LIBRARY.....	62
FIGURE 32: ADDING THE PAD FRAME TO THE NEW LIBRARY.....	63
FIGURE 33: IMPORT GDS (MAIN)	64
FIGURE 34: IMPORT GDS (LAYER MAP FILE)	64
FIGURE 35: CREATING SYMBOL AND SCHEMATIC (MAIN)	65
FIGURE 36: CREATING SYMBOL AND SCHEMATIC (OPTIONS)	66
FIGURE 37: SYMBOL CREATED	67
FIGURE 38: SCHEMATIC CREATED	67
FIGURE 39: CORE PORTS	68

FIGURE 40: CHIP LAYOUT	69
FIGURE 41: CHIP CONNECTIONS	70
FIGURE 42: VDD AND GND CONNECTION.....	71
FIGURE 43: ENABLE TIED LOW	71
FIGURE 44: ENABLE TIED HIGH.....	72
FIGURE 45: FINAL CHIP SCHEMATIC.....	73
FIGURE 46: GDS EXPORT.....	74

PREVIEW

CHAPTER 1: INTRODUCTION

1.1 What is an ASIC?

Integrated circuits are usually fabricated on silicon wafers with the wafer containing hundreds of dice structures. ASIC stands for Application Specific Integrated Circuit and an IC designed for a specific application while using standard cell library is referred to an ASIC. Examples of ASICS include chips designed for satellites, chips designed for automotive systems, and chips designed as interfaces between memory and CPU on a personal computer. Examples of IC's which are not ASIC include memories, microprocessors etc.

1.2 Types of ASICS

ASICS are categorized based on the technology used for manufacture. The various types are full-custom ASICS and semi-custom and semi-custom can be further classified as standard cell based ICs (CBICs), Gate Array (GA) type.

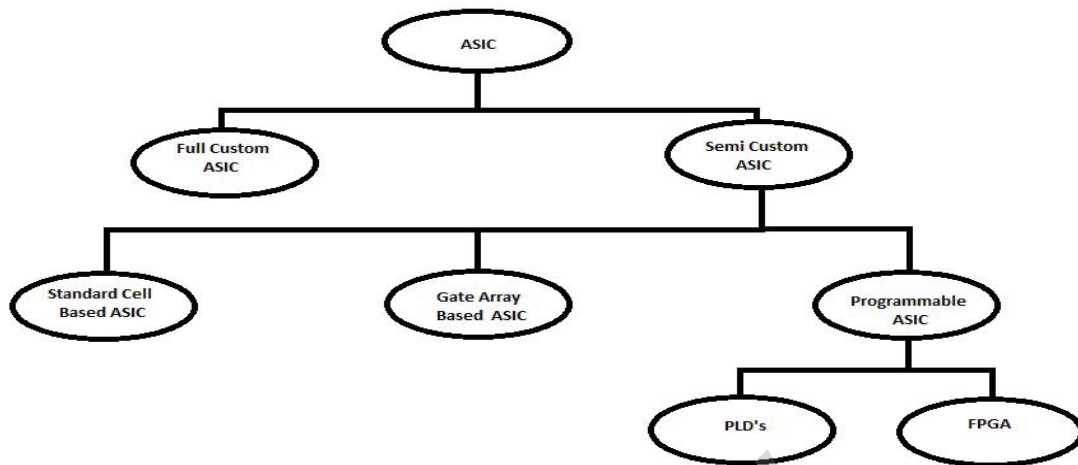


Figure 1: Types Of ASICS

1.2.1 FULL CUSTOM ASICS

The mask layers are customized in a full-custom ASIC where a majority of the design is done at the transistor level in terms of layout and simulation. Full custom ASICS are designed if there are no standard libraries available or when particular care is required to meet performance or power requirements of an application. The full custom methodology offers the highest performance with the disadvantages of increased design time, complexity, design expense, and highest risk. Microprocessors are generally full-custom, but designers are increasingly turning to semicustom ASIC techniques in this area as well. Other examples of full-custom ASICS are high-voltage automobile, analog/digital (communications), or sensors and actuators.

The disadvantages of full custom ASIC are:

1. Design complexity
2. Lack of automation in the tools
3. Substantial design time

1.2.2 SEMI-CUSTOM ASICS

Semi-custom ASICS are ASICS that are customized in one or two areas. A semi-custom ASIC is manufactured with the masks for the diffused layers already fully defined, so the transistors and other active components of the circuit are already fixed. The customization of the final ASIC product to the intended application is done by varying the masks of the interconnection layers, e.g., the metallization layers.

1.2.3 STANDARD CELL ASICS

A standard cell-based ASIC uses standard cells which are predesigned logic cells (AND gates, OR gates, multiplexers, and flip-flops, for example). The standard-cell areas (also called flexible blocks) in a Standard Cell Based ASIC are built of rows of standard cells. The standard-cell areas may be used in combination with larger predesigned cells like memories or analog circuits.

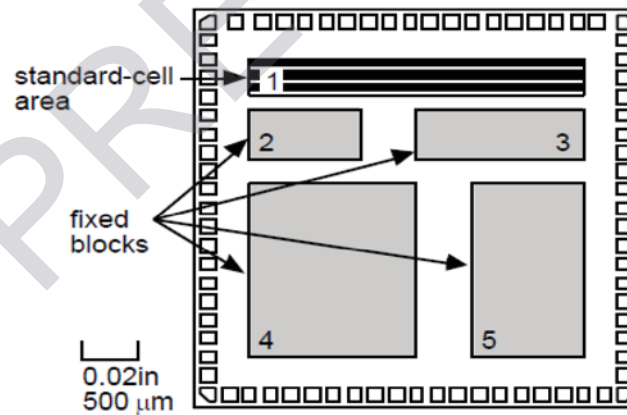


Figure 2:Standard Cell Based ASIC

1.2.4 GATE ARRAY ASICS

Gate array ASICS are partially finished with rows of transistors and resistors required but the transistors are unconnected. The chip is completed by connecting the required transistors and

resistors with the back-end metal layers. The Gate array is made of “basic cells”, where individual cells contain some number of transistors and resistors depending on the vendor. Using a cell library (gates, registers, etc...) and a macro library (more complex functions), the customer designs the chip and the vendor’s software generates the interconnection masks.

These final masking stages are less costly than those associated with designing a full-custom ASIC from scratch.

A gate array circuit is a prefabricated circuit with no particular function in which transistors, standard logic gates, and other active devices are placed at regular predefined positions and manufactured on a wafer, usually called **Master Slice**. Creation of a circuit with a specified function is achieved by connecting the required elements using metal layers at the time of manufacture and this can be done on wafers that have already been fabricated with complete transistors (front end) and thus eliminate the amount of time between completing the design and obtaining silicon. The gate array drawbacks are low density and low performance as compared to full custom or standard cell ASICs.

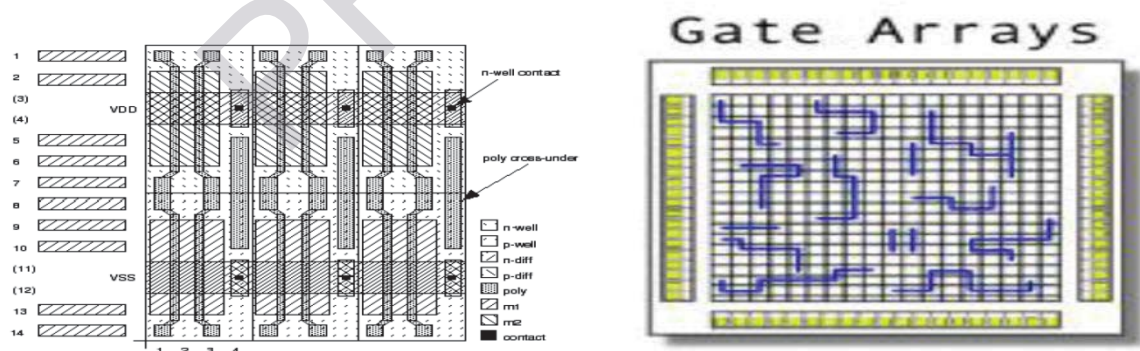


Figure 3: Gate Array ASIC

1.2.5 PROGRAMMABLE ASICS

a) PLDs are programmable logic devices that can be used to perform complex functions. PLDs can be programmed for specific applications. PLDs use different technologies to allow programming of the device.

PLDs are having these common features:

- No customized mask layers or logic cells
- Fast design turnaround
- A single large block of programmable interconnect
- A matrix of logic macro cells that usually consist of programmable array logic followed by a flip-flop or latch.

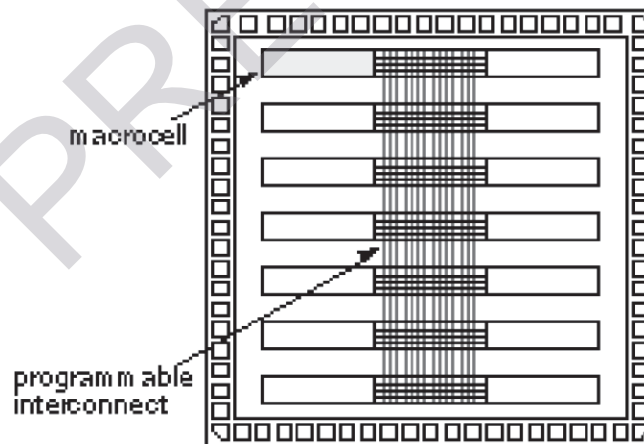


Figure 4:PLD

b) FPGA (Field Programmable Gate Array) are first developed in the mid of 80's by Xilinx. FPGAs are better in terms of functionality in comparison with PLDs and can be used for more complex and denser designs.

FPGAs are composed of logic blocks instead of unwired transistors.

The core of the FPGA is an array for logic cells that can perform combinational and sequential logic

FPGAs have the following features:

- Less dense than custom mask
- Higher unit cost
- Ready to be used out the box
- Reprogrammable
- In this project, in which a methodology is established to be used in an undergraduate

design project, which ends with a free 0.5 micron CMOS fabrication, an FPGA-based emulation process is used to evaluate and validate a design written in Verilog – similar to the approach taken by industry. This significantly reduces the risk of the chip design being fabricated that is wrong.

Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs) are different types of custom chips, which differ in their properties, cost, and manufacturing process. The choice of which to use depends on the required application and its requirements and in some applications, FPGAs are used to prototype a design given the low NRE expense, but are replaced in production by ASICs to improve the per device costs [2].

Older style ASICs were of gate array type which consisted of unconnected transistors. The most common type of ASIC currently used is the standard cell based ASIC which accounts for most digital logic fabricated today. Many electronics companies with a core competency outside of the chip industry (e.g. routers, cell phones, graphics processing) can design customized chips with the

standard cell library and fabricate the chips at a silicon foundry. Examples of such fab-less companies include NVidia or Qualcomm and this business model is gaining popularity given the huge costs associated with silicon manufacturing and the availability of reliable and inexpensive foundry services. The most popular foundry services are now located in Taiwan and China with companies like the Taiwanese Semiconductor Manufacturing Company or Universal Manufacturing Company. Companies like IBM and Texas Instruments in the USA also provide these services generally providing better performance but at an increased cost per die [2].

1.3 Chapter Organization

The focus of this thesis is to develop a methodology to generate chips at an education level for undergraduates using standard industry EDA tools by vendors like Synopsys and Cadence. The remainder of this thesis is organized as follows: In chapter 2 the main focus is on how this new ASIC design methodology has made the whole process of chip production fast, easy and efficient. The chapter also discusses digital logic basics defining terms like propagation delays with a brief description of the power consumption in digital circuits. Chapter 3 focuses mainly on the tool Design Compiler (DC) by Synopsys used for the synthesis of RTL. Chapter 4 mainly talks about the synthesis optimization techniques that could be used for improving area and speed of the final chip. Chapter 5 describes focuses on the tool IC Compiler by Synopsys which is used mainly for the place and route operation to generate the final core of the chip in design. In chapter 6 the methodology used is discussed. Chapter 7 provides the conclusion and future work. Appendix A lists the Verilog code for the simple 8 bit counter used to test the methodology. Appendix B shows the synthesis TCL script that was used. Appendix C lists the IC Compiler TCL script that was used for design place and route operation.