MIGRATION FROM ASYNCHRONOUS TO SYNCHRONOUS 8254 PROGRAMMABLE INTERVAL TIMER FOR EFFECTIVE STATIC TIMING ANALYSIS

ONG CHUN KIAT

UNIVERSITI SAINS MALAYSIA

2013

MIGRATION FROM ASYNCHRONOUS TO SYNCHRONOUS 8254 PROGRAMMABLE INTERVAL TIMER FOR EFFECTIVE STATIC TIMING ANALYSIS

By

ONG CHUN KIAT

**Thesis submitted in fulfillment of the requirements
for the degree of
Masters of Science**

July 2013

ACKNOWLEDGEMENT

This thesis would not have been possible without the help and guidance from several individuals whom I have met throughout my research. They have gratefully expressed their willingness to spend their extra time in the contribution towards my understanding and thoughts of the research. First and foremost, I wish to offer my sincere gratitude to my project supervisor, Dr. Mohd Tafir Mustaffa, for his continuous support, encouragement, and guidance towards me in producing a better quality thesis.

Also, I would like to express my sincere appreciation to my mentor, Goh Ling How, who guides me along the research towards the direction in completing my thesis. Besides, the contribution of support from my manager and colleague shall be recognized. Special thanks to Eugene Lim, Lee Ming Chew, and Goh Keng Hoo for their support in ramping up my knowledge.

I would also like to thank Intel Microelectronics (M) Sdn. Bhd for entrusting me to use their Intellectual Properties (IP) as the main core of this research as well as for providing all facilities and resources required to support my research.

Lastly, I would like to thank all my family members, who motivated me the most during my hard time throughout this research. Without their encouragement, I would have given up in the middle of this research.

TABLE OF CONTENTS

ACKNOWLEDGEMENT ii

TABLE OF CONTENTS iii

LIST OF TABLES vii

LIST OF FIGURES viii

LIST OF ABBREVIATIONS xii

ABSTRAK xiii

ABSTRACT xv

CHAPTER 1 : INTRODUCTION 1

1.1 Background 1

1.2 Problem Statement 4

1.3 Objectives 6

1.4 Scopes 6

1.5 Research Contribution 7

1.6 Thesis Outline 7

CHAPTER 2 : LITERATURE REVIEW 9

2.1 Introduction 9

2.2 8254 Programmable Interval Timer 11

2.3 Digital Design Flow 17

2.4 Static Timing Analysis 19

2.5 Gate Level Simulation 22

2.6 Asynchronous 8254 PIT with Latch 25

2.7 Clock Domain Crossing in asynchronous 8254 PIT..27

2.8 Threat in Asynchronous 8254 PIT...29

2.8.1 Bug in Asynchronous 8254 PIT..29

2.8.2 Timing Violation..30

2.9 Summary..30

CHAPTER 3 : METHODOLOGY...33

3.1 Introduction...33

3.2 Overview of the design implementation..34

3.3 Asynchronous path finding...35

3.3.1 Combinational Timing Loop..36

3.3.2 Non-clocked Flip-flop and State machine 37

3.4 Technique conversion from asynchronous design to synchronous design 39

3.4.1 Replacement of non-clocked flip-flop with system clock 41

3.4.2 Delay elements..48

3.4.3 Combinational timing loop 51

3.4.4 Latch 52

3.4.5 Clock domain crossing..54

3.4.6 Reset...60

3.5 Overview of Synchronous 8254 PIT implementation 62

3.6 Functional Validation....64

3.6.1 RTL coding....64

3.6.2 Testbench....65

3.6.3 Functional Test....66

3.7 Summary....70

CHAPTER 4 : RESULTS AND DISCUSSION....71

4.1 Introduction....71

4.2 Functional validation results & discussion....71

4.2.1 Test 5 results....72

4.2.2 Test 7 results....75

4.2.3 Test 9 results....76

4.2.4 Test 11 results....78

4.2.5 Test 20 results....81

4.2.6 Overall synchronous 8254 PIT regression run....83

4.3 RTL synthesis....83

4.4 Advantages of synchronous 8254 PIT....84

4.4.1 Static Timing Analysis Coverage....85

4.4.2 Bug in asynchronous 8254 PIT solved....87

4.5 Limitation of Synchronous 8254 PIT....87

4.5.1 Power consumption....88

4.5.2 Area....89

4.6 Overall Discussion 89

CHAPTER 5 : CONCLUSION 91

5.1 Conclusion 91

5.2 Future Recommendation 93

REFERENCES 95

LIST OF PUBLICATIONS 98

APPENDICES 99

APPENDIX A 100

APPENDIX B 104

APPENDIX C 106

APPENDIX D 107

LIST OF TABLES

Table 4.1: Total power of asynchronous and synchronous 8254 PIT 88

Table 4.2: Area and Gate Count for asynchronous and synchronous 8254 PIT 89

LIST OF FIGURES

Figure 1.1: Example of synchronous design 1

Figure 1.2: Example of asynchronous design............ 2

Figure 1.3: Example of asynchronous design with different clock domain 2

Figure 2.1: Block diagram of 8254 PIT (Intel 8254 PIT 1981 Datasheet)............ 12

Figure 2.2: Architecture of asynchronous 8254 PIT 13

Figure 2.3: Element inside asynchronous 8254 PIT............ 14

Figure 2.4: Overview of 8254 PIT clock system generation............ 15

Figure 2.5: Read and write signal generation state machine 15

Figure 2.6: Waveform description of the read and write state machine............ 16

Figure 2.7: General Design Flow 17

Figure 2.8: Delay versus Process............ 20

Figure 2.9: Delay versus Voltage 21

Figure 2.10: Delay versus Temperature 21

Figure 2.11: SOC structure............ 23

Figure 2.12: Difference between output of flip-flop and latch............ 26

Figure 2.13: Latch design............ 26

Figure 2.14: Design with different clock domain............ 28

Figure 2.15: Waveform simulation for design in Figure 3.30 28

Figure 2.16: Bug found in asynchronous 8254 PIT 29

Figure 3.1: Design implementation flow............ 34

Figure 3.2: Combinational timing loop 36

Figure 3.3: Unstable loop 36

Figure 3.4: Stable but stuck-at loop............ 37

Figure 3.5: Non-clocked flip-flop...38

Figure 3.6: Non-clocked state machine...39

Figure 3.7: Failure of timer clock to capture the read and write line 40

Figure 3.8: Simulation's waveform for design in Figure 3.5 41

Figure 3.9: Synchronous design with AND gate control signal...............................42

Figure 3.10: Simulation's waveform for design in Figure 3.9 43

Figure 3.11: Synchronous design with multiplexer as control signal 44

Figure 3.12: Simulation's waveform for design in Figure 3.11 44

Figure 3.13: Simulation's waveform for design in Figure 3.11 with different input 45

Figure 3.14: Synchronous design with pulse generation..46

Figure 3.15: Simulation's waveform for design in Figure 3.14 46

Figure 3.16: Simulation's waveform of asynchronous 8254 PIT.............................47

Figure 3.17: Simulation's waveform of synchronous 8254 PIT 48

Figure 3.18: Asynchronous design with delay elements..49

Figure 3.19: Simulation waveform for design in Figure 3.18 49

Figure 3.20: Delay element used for "punch through" technique 50

Figure 3.21: Simulation waveform for design in 3.20...50

Figure 3.22: Timing loop elimination by flip-flop insertion 51

Figure 3.23: Timing loop elimination for unstable loop 51

Figure 3.24: Timing loop elimination for Stable but stuck-at loop...........................52

Figure 3.25: Transformation from latch based to flip-flop based 53

Figure 3.26: Simulation waveform for latch based design.......................................53

Figure 3.27: Simulation waveform for flip-flop based design 53

Figure 3.28: Implementation of synchronizer for design in Figure 3.30...................54

Figure 3.29: Simulation waveform for design in Figure 3.28 55

Figure 3.30: Asynchronous 8254 PIT with 2 different clock domain 56

Figure 3.31: Simulation waveform for design in Figure 3.29 57

Figure 3.32: Synchronous design with proper clock domain crossing 58

Figure 3.33: Simulation waveform for design in Figure 3.32 58

Figure 3.34: Synchronous design without synchronizer 59

Figure 3.35: Simulation waveform for design in Figure 3.34 60

Figure 3.36: Reset recovery time 61

Figure 3.37: Reset removal time 61

Figure 3.38: Clock system of asynchronous 8254 PIT 62

Figure 3.39: Clock system of synchronous 8254 PIT 63

Figure 3.40: RTL coding for a multiplexer 65

Figure 3.41: RTL coding for a flip-flop 65

Figure 3.42: Validation environment 66

Figure 4.1: Write signal for asynchronous 8254 PIT 72

Figure 4.2: Write signal for synchronous 8254 PIT 73

Figure 4.3: Test 5 with asynchronous 8254 PIT 74

Figure 4.4: Test 5 with synchronous 8254 PIT 74

Figure 4.5: Test 7 with synchronous 8254 PIT 75

Figure 4.6: Test 7 with asynchronous 8254 PIT 76

Figure 4.7: Test 9 with synchronous 8254 PIT 77

Figure 4.8: Test 9 with asynchronous 8254 PIT 78

Figure 4.9: Test 11 with synchronous 8254 PIT 79

Figure 4.10: Test 11 with asynchronous 8254 PIT 80

Figure 4.11: Test 20 with synchronous 8254 PIT 81

Figure 4.12: Test 20 with asynchronous 8254 PIT 82

Figure 4.13: Results of the regression run for synchronous 8254 PIT 83

Figure 4.14: Clock constrain created for synchronous 8254 PIT 84

Figure 4.15: Synchronous 8254 PIT report analysis coverage 85

Figure 4.16: Asynchronous 8254 PIT report analysis coverage 85

Figure 4.17: Synchronous design implementation 87

LIST OF ABBREVIATIONS

ASIC:	Application-specific Integrated Circuits
BFM:	Bus Functional Model
CAD:	Computer Aid Design
CTS:	Clock Tree Synthesis
DC:	Design Compiler
EDA:	Electronics Design Automation
GLS:	Gate Level Simulation
GPIO:	General Purpose Input Output
HDL:	Hardware Description Language
IP:	Intellectual Property
LPC:	Low Pin Count
PC:	Personal Computer
PCH:	Platform Control Hub
PCIE:	Peripheral Component Interconnect Express
PIT:	Programmable Interval Timer
PV:	Performance Verification
PVT:	Process, Voltage and Temperature
RTL:	Register Transfer Logic
SATA:	Serial Advanced Technology Attachment
SDF:	Standard Delay Format
SMB:	System Management Bus
STA:	Static Timing Analysis
USB2:	Universal Serial Bus 2
VHDL:	VHSIC Hardware Description Language
VHSIC:	Very-High-Speed Integrated Circuit

PERALIHAN DARIPADA PEMASA JEDA 8254 BOLEH ATURCARA TAK SEGERAK KEPADA SEGERAK BAGI KEBERKESANAN ANALISIS PEMASAAN STATIK

ABSTRAK

Dengan pengecutan saiz transistor, kesukaran rekabentuk untuk memenuhi pemasaan telah meningkat. Pengecutan saiz transistor yang berterusan dari masa ke semasa juga telah meningkatkan perubahan pada ‘die’ dari segi Proses, Voltan, dan Suhu (PVT) cip. Memandangkan alat Prestasi Pengesahan (PV) proses atau Analisis Masa statik (STA) tidak dapat menampung pengesahan masa rekabentuk tak segerak, rekabentuk tak segerak perlu ditukar kepada rekabentuk segerak supaya alat STA dapat memastikan masa silikon boleh dipatuhi dalam perubahan PVT. Rekabentuk tak segerak adalah rekabentuk di mana transaksi data dikawal oleh jam sedangkan rekabentuk tak segerak adalah rekabentuk di mana transaksi data adalah tidak berdasarkan pada jam. 8254 Jeda Pemasa Boleh Aturcara (PIT) Intel telah digunakan sebagai kajian kes dalam penyelidikan ini. Satu aliran rekabentuk termasuk pengenalan laluan segerak dan pelaksanaan teknik dari tak segerak ke segerak telah dicadangkan. Pelbagai teknik penukaran dari tak segerak kepada segerak tanpa perubahan fungsi termasuk penempatan jam, kelewatan, gelung bergabungan, selak, menyeberangi domain jam, dan reset dilaksanakan pada PIT 8254. Fungsi PIT segerak 8254 disahkan menggunakan 22 kes ujian. Keputusan menunjukkan liputan analisis masa 8254 PIT oleh STA telah meningkat dari 40% dalam rekabentuk tak segerak berdasarkan kepada 91% dalam rekabentuk segerak berasaskan. Liputan analisis masa baru segerak 8254 PIT akan menjadi 100% jika tidak termasuk “reset” tak segerak. Namun terdapat kelemahan pada rekabentuk segerak iaitu terdapat peningkatan kuasa dan luas ‘die’ PIT 8254. Pada masa hadapan, penambahan boleh dilakukan pada rekabentuk segerak untuk mengurangkan kuasa dan luas rekabentuk. Secara keseluruhannya, pelaksanaan PIT segerak 8254 telah berjaya mencapai objektif kerana rekabentuk tidak perlu bergantung pada GLS (Gate Level Simulation) untuk pengesahan masa dan ini telah mengurangkan masa yang digunakan untuk membina model GLS kelompok dan menyelesaikan masalah rekabentuk tak segerak.

MIGRATION FROM ASYNCHRONOUS TO SYNCHRONOUS 8245 PROGRAMMABLE INTERVAL TIMER FOR EFFECTIVE STATIC TIMING ANALYSIS

ABSTRACT

As the size of transistor is shrinking, the difficulty of a design to meet timing has increased. Also, continuously shrinking of transistor size from time to time has increased the on-die variation such as Process, Voltage, and Temperature (PVT) variation of the chip. Since Performance Verification (PV) process or Static Timing Analysis (STA) tool is unable to cover the timing verification of asynchronous based design, asynchronous design is changed to synchronous based for the STA tools to ensure the silicon timing can be met across various PVT. Synchronous design is a design where the transaction of data is governed by clock whereas asynchronous is a design where the transaction of data is event based which is not based on clock. Intel 8254 Programmable Interval Timer (PIT) which is an asynchronous based design is used as a case study in this research. A design flow which included the asynchronous path identification and technique implementation from asynchronous to synchronous without functionality changes has been proposed. Techniques conversion from asynchronous to synchronous including the clock placement, delays, combinational loop, latch, clock domain crossing, and reset are implemented on the 8254 PIT. The functionality of the synchronous 8254 PIT is validated with total of 22 test cases. Result shows the timing analysis coverage of 8254 PIT by STA has been increased from 40% in asynchronous design based to 91% in synchronous design based. The timing analysis coverage of new synchronous 8254 PIT will be 100% if excluding the asynchronous reset path. On the drawback of synchronous design, the result shows the power and gate count of 8254 PIT has increased. Future work can be done on the synchronous design to reduce the power and gate count. Overall, the implementation of synchronous 8254 PIT has successfully achieved the objectives as the design is no longer required to rely on GLS (Gate Level Simulation) for timing verification and this has reduced the time spent on building up a cluster GLS model and debugging the asynchronous design.

CHAPTER 1

INTRODUCTION

1.1 Background

In the world of digital design, there are seven basic logic gate which is AND, OR, NOT, NOR, NAND, XOR, and XNOR. All the logic gates can only take in logical input and then produce logical output which is either in “0” or “1”. With the combination of all these logic gates, some common logic such as flip-flop, latch, multiplexer, and state machine can be designed. A digital design can consist of logic elements and functioning based on the clock supply to the design. When it comes to clock related design, there will be two different types of design which is asynchronous design and synchronous design.

Synchronous design is a design where all the transaction of data is governed by clock. In synchronous design, transaction of data from one storage elements to another is synchronized with same clock. An example of synchronous design is shown in Figure 1.1 where both unit A and unit B are clocked by same clock source. Both Unit A and Unit B are synchronous to each other.

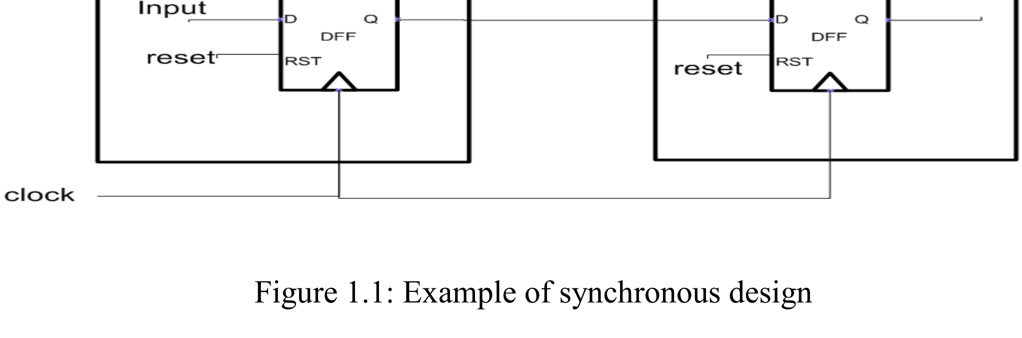

Figure 1.1: Example of synchronous design

In contrast to synchronous design, asynchronous design is a design where the transaction of data is not based on clock. The transaction of data in asynchronous design is event based instead of clock based where the data change only happens when there is an event happened. A simple asynchronous design is shown in Figure 1.2. The flip-flop in Unit B will only trigger whenever there is an output change from the flip-flop in Unit A. The term asynchronous is also used to describe a design that consists of more than one clock sources. In Figure 1.3, Unit A and Unit B are asynchronous to each other as clock A and clock B are from different source. Even if the clock frequency of clock A and clock B are same, they are still considered asynchronous to each other as there's totally no relationship in between both clock.

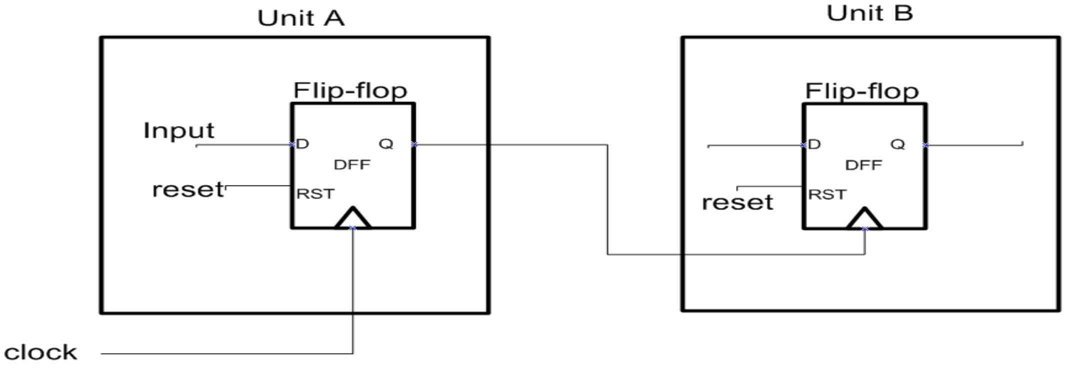

Figure 1.2: Example of asynchronous design

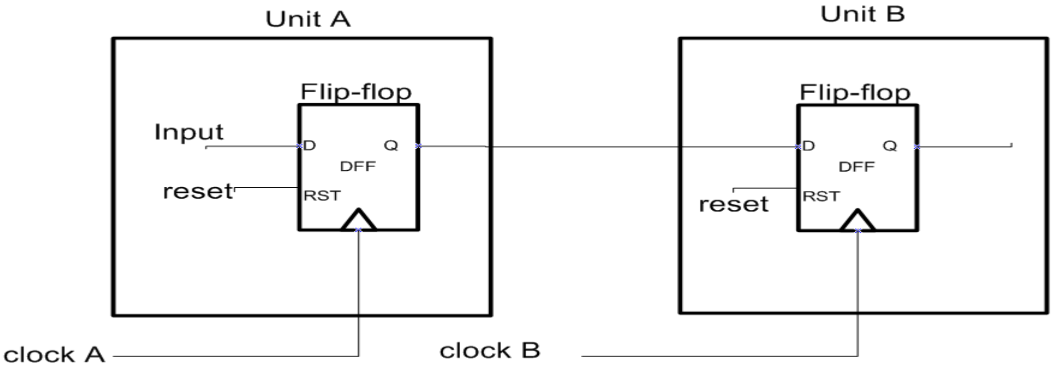

Figure 1.3: Example of asynchronous design with different clock domain

In the past, there have been many debates regarding asynchronous design and synchronous design. Asynchronous design is well known as a low power design as because the clock line at the register only toggles when there's an event. For synchronous design, the clock will continuously toggle even when there is no activity and this will consume a lot of power in the design. Besides, there are some advantages of asynchronous design such as no clock skew and better performance of asynchronous design stated by many people (Scott Hauck 1995; Jens Sparsø 2006; Michael Brandon Roth 2004). Since asynchronous design do not have globally distributed clock, hence there is no clock skew problem. Asynchronous design is believed to be faster as it does not need to wait until all possible computations to be completed before latching the results. From the research, majority of the people are discussing on asynchronous design rather than synchronous design in the recent year. To the author's knowledge, the only person discussing about synchronous design is about 20 years ago (Forshaw et al. 1990). Despite the popularity of asynchronous design, synchronous design still predominates in digital design. The main reason is currently there are no existing tools that can support asynchronous design and subsequently cause the difficulty of timing verification in asynchronous design (Kondratyev & Lwin 2002; Jung-Lin Yang et al. 1995).

According to Moore's law, the size of transistor is decreasing for approximately every 18 months since 1971. In the beginning of 1971, the size of transistor was 10 µm. Until now, the size of transistor has been dramatically decreased to 22 nm and will be even lower in the future. With the current technology, i.e. 32 nm, Process, Voltage, and Temperature (PVT) variation has become one of the main concerns for the design to meet the timing (Zuchowski et al.

2004). This is due to the increased of on-die variation in the chip as the size of transistors continuously shrinks. The sensitivity of circuit towards PVT variation has increased as the transistor size is shrinking (Bowman et al. 2002). Circuit performance and yield will be affected if the process control of PVT is not handled well. There are several methodologies being proposed to reduce and compensate the PVT variation (Ik Joon Chang et al. 2010; S.K. Shin et al. 2005; S. V. Kumar et al. 2008). From the design perspective, there is something that can be done to guarantee the design to operate at different PVT which is by using synchronous design instead of asynchronous design.

1.2 Problem Statement

Performance Verification (PV) is a pre-silicon activity in which circuit timing is assessed with timing analysis tools to determine attainable frequencies. During the process of PV, a Static Timing Analysis (STA) tool, typically PrimeTime which is developed by Synopsys is used to calculate the expected timing of a design without running simulation. This tool is capable to verify the timing for all possible paths of a design in order for a design to be able to operate at different operating condition such as PVT. However, there is a limitation of STA tool as it can’t handle asynchronous design due to asynchronous circuit only assert signal after alerted by some events. For this reason, asynchronous designs are black-boxed during PV and therefore timing verification is not covered. In this case, the timing of asynchronous path in the design has to be design guaranteed to ensure the proper handshake in between the asynchronous paths. Therefore, the work around this will not be as accurate as when using STA tool because the delay will varies in the actual silicon.

To ensure asynchronous design is working correctly, the whole chip needs to be run on Gate Level Simulation (GLS). GLS is an activity to validate the netlist produced after synthesis. During RTL simulation, all the timing in the design is assumed to be ideal. In GLS, the design contains the timing information such as Standard Delay Format (SDF) files that stores the timing data generated by Electronics Design Automation (EDA) tools. The design will be verified with the actual behavior of the silicon. The asynchronous design will be relying on the GLS model to verify their timing to ensure the design meets the timing requirement without violation such as setup, hold, recover, removal, and width violation.

However, the efficiency of GLS is not comparable to STA timing verification as it only subjected to test pattern and also, it does not have the ability to exercise all possible paths moreover might be a minor potential threat to post-silicon where a bug can be found even though all the tests passed during GLS run. Despite of that, the designer has no choice but to completely rely on GLS to catch as many bugs as possible because there is currently no tool that supports asynchronous design. Besides, building of GLS model are also time consuming as at least typically, three days is needed to set up the GLS model from the experience of an engineer. Time spent to debug asynchronous design in GLS will be another factor as it will usually take up to two weeks and this subsequently cause a threat to the timeline of a project which will gate the tape-in schedule.

In prior to overcome the problem of asynchronous design, the idea has come to industries who decide to migrate all the existing asynchronous based design in the chip to synchronous based design in the future. With all existing asynchronous design in the chip changed to synchronous design, GLS run on cluster level will not be needed. All the timing of the design can be handled during PV and the design does not need to rely completely on GLS. GLS will become an optional rather than a must on cluster level and it will now serve as bonus verification to the design. Time spent on building and debugging the GLS model can be neglected and this has greatly increase the buffer of the project timeline as engineer no longer need to spend extra time debugging on asynchronous design.

1.3 Objectives

The main objectives of this work are as below:

(i) To develop a new flow to change asynchronous design to synchronous design without excessive architecture changes.

(ii) To evaluate the advantage of synchronous design over asynchronous design.

(iii) To increase the project buffer time without running Cluster Gate Level Simulation (GLS).

(iv) To increase timing analysis coverage of a design.

1.4 Scopes

This work will focus on the following scopes:

(i) To convert the current asynchronous design based 8254 PIT to synchronous design based on the proposed flow.

(ii) To validate the functionality of the synchronous 8254 PIT after conversion to ensure the functionality is maintained.

(iii) The synchronous 8254 PIT design will be completed up to the stage of STA which is after synthesis.

1.5 Research Contribution

The contributions of providing the asynchronous to synchronous design flow in this research are as below:

- Provide asynchronous to synchronous design flow and technique without massive changes on the architecture while maintaining the design's functionality.
- Implementation of synchronous design 8254 PIT will guarantee the design's timing verification with STA.
- Cluster GLS will be abolished with the implementation of synchronous design 8254 PIT and thus eliminate the GLS model setup time which can take up to two weeks.
- Most of the time has been spent by engineer on debugging asynchronous design over synchronous design. With implementation of synchronous 8254 PIT, time spent on debugging 8254 PIT will be reduced due to the failure are mostly on timing problem where the asynchronous design is unable to meet the timing requirement during GLS.

1.6 Thesis Outline

This thesis consists of five chapters. Chapter one introduces the background of the work which related to digital design. The problem statement has been stated clearly. Besides, the objective, scopes and the research contributions are clearly clarified.

Chapter two gives an overview of current asynchronous design of 8254 PIT. The overview of the design flow from RLT development until tape in will also be discussed. Basic knowledge regarding the advantages and disadvantages of STA and GLS will be included in this chapter as well.

The proposed methodology will be in chapter three. This chapter will discuss about different types of asynchronous design used in current 8254 PIT. The proposed methodology to covert the current asynchronous based design to synchronous based design is presented with 8254 PIT being used as a test case.

In chapter four, the experimental result of the modified synchronous design based on the proposed technique will be carried out. Comparison between the synchronous design and asynchronous design of 8254 PIT will be discussed.

In the final chapter which is chapter five, the conclusions of this thesis and future work are presented.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

8254 Programmable Interval Timer (PIT) is a timer device designed to solve the common timing control problem in microcomputer system design. 8254 PIT is one of the units used in the chip where the design does not change. It was introduced in the IBM Personal Computer (PC) since 1981. The device can provide some timer function such as real time clock, event counter, and square wave generator. In the uni-processor system, the 8254 PIT is mainly used to provide periodic clock interrupt and generate speaker "beep" sound by using square wave generator. The design of 8254 PIT is always asynchronous based since when it was introduced. Due to the stability of the design throughout so many years, no engineer would like to take the extra effort to modify the design which might potentially break the design and need a lot of effort in debugging and validating the brand new design.

With the experience of this few years, there are some reasons that encourage industries, particularly Intel to take the initiative to migrate the current asynchronous based design 8254 PIT to synchronous based design. First, the on-die variation in new process technology is getting larger as the transistor size is shrinking and this increase the sensitivity of circuit towards process, voltage and temperature (PVT) variation. As the limitation of Computer Aid Design (CAD) tools available in the market, the only way to guarantee the design to operate in different PVT variation is by using STA tools to calculate the timing for the minimum and maximum corner of a design (Forzan & Pandini 2007). As mention in previous chapter, STA does not support asynchronous design. This led to designer with no other choice but to rely on GLS to verify the timing of asynchronous based design 8254 PIT. GLS does not completely verify the entire possible path in the design as it is just test pattern based verification. Second, the growth of design size causing the GLS throughput time increased exponentially and become very inefficient to debug. This is due to more spaces are available when transistor size shrinking and thus more features have been added which drastically increase the design gate count.

In this chapter, the overview of asynchronous based design 8254 PIT will be discussed. Section 2.2 will explain about functionality and asynchronous design of 8254 PIT in more detail for better understanding. In Section 2.3, the design flow will be discussed and focus of research will be explained.

STA is a method of calculating the expected timing of a design without running simulation. There are a lots of STA tools available in the market such as PrimeTime from Synopsys, common Timing Engine from Cadence, Time craft from Incentia, and SST velocity from Mentor Graphic. The important of STA tools to the design will be discussed in Section 2.4 mainly on PrimeTime from Synopsys.

In Section 2.5, the GLS methodology on full chip and cluster level will be discussed. The advantages and disadvantages of running GLS will be listed. Comparison between STA and GLS will be made base on the efficiency and limitation of both methodology.

2.2 8254 Programmable Interval Timer

8254 PIT is a programmable timer device design for the use of Intel microcomputer system. 8254 PIT has three independent 16-bit counters, with each capable of handling up to 10 MHz of clock inputs. There are a total of six different modes can be programmed into the 8254 PIT where each different mode will generate different types of output. With all the six different modes, several common function used for microcomputer such as real time clock, event-counter, digital one-shot, programmable rate generator, square wave generator, binary rate multiplier, complex waveform generator, and complex motor controller can be implemented. In the original IBM PC, the first counter which is counter 0 is typically programmed for the use of system timer to generate interrupt. Second counter, counter 1 is used to trigger the refresh of DRAM memory. The third counter which is counter 2 is used to generate the tone of the speaker PC. Further understanding regarding the behavior of each mode programmed can be read through datasheet (Intel 8254 Programmable Interval Timer datasheet). The main input and output of 8254 PIT are data (D7-D0), read line (RD), write line (WR), address select (A0 and A1), chip select (CS), clock (CLK 0, CLK 1, and CLK 2), gate (GATE 0, GATE 1, and GATE 2), and output (OUT 0, OUT1, and OUT2) as shown in Figure 2.1. D7-D0 is the 8 bit data input that can be programmed into the device. RD is the read signal to be triggered whenever user read something from the counter whereas WR is the write signal to be triggered whenever user program the data into the Timer Control Word (TCW) register or write count value into the counter.

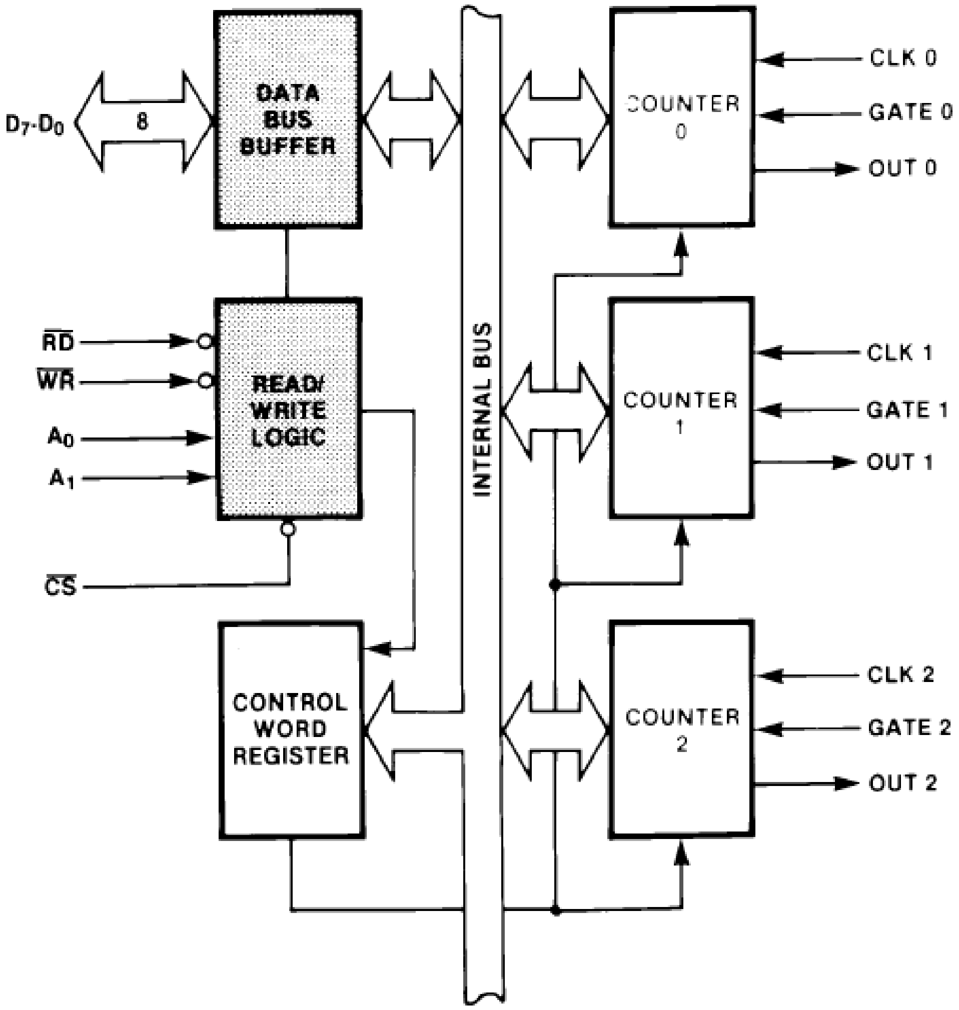

Figure 2.1: Block diagram of 8254 PIT (Intel 8254 PIT 1981 Datasheet)

In the architecture of current asynchronous 8254 PIT, it mainly consists of four different blocks which is DTU, CRDWR, CSEQ, and CNT. DTU is a block that is responsible to decode all the signal. All the input signals of 8254 PIT are going in the DTU block for decode before it goes to the other blocks. CRDWR is a block that handles the read and write signals of the 8254 PIT. CSEQ is a block where all the programming mode and functional logics are located. This CSEQ block is in charge of controlling the behavior of the counter according to the mode programmed. CNT is a block that consists of the counter's logic. The counters will behave according to the command sent from CRDWR and CSEQ block. The architecture of the current asynchronous 8254 PIT can be seen in Figure 2.2.

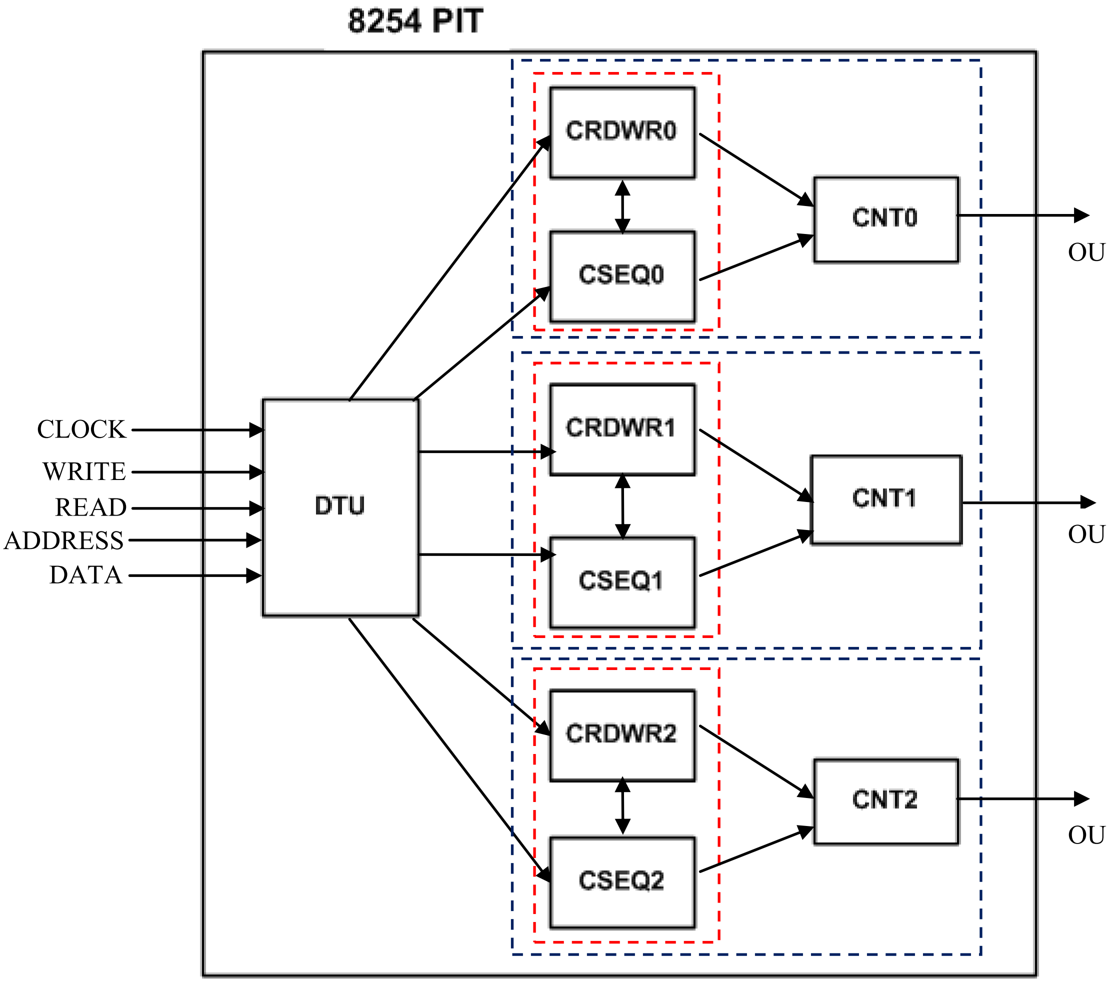

Figure 2.2: Architecture of asynchronous 8254 PIT

As each counter is independent of each other, each counter has its own CRDWR, CSEQ, and CNT block. Inside these blocks, there are some elements such as flip-flop, state machine and latch as shown in Figure 2.3. DTU only consist of combinational logics.

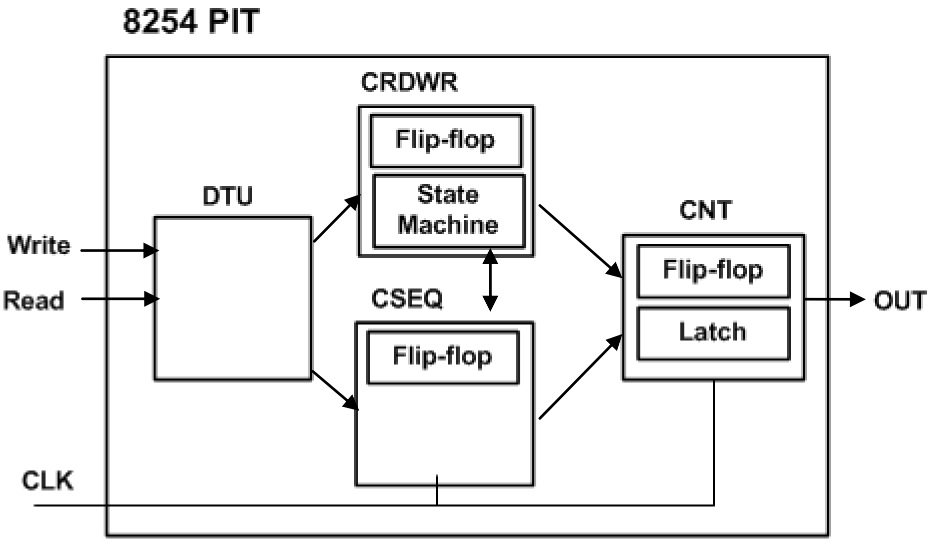

Figure 2.3: Element inside asynchronous 8254 PIT

For a design to be synchronous, all the flip-flops and state machines must be supplied with a clock. From Figure 2.3, only CSEQ and CNT block are supplied with clock. CRDWR are not supplied with any clock but consist of flip-flops and state machines. The current 8254 PIT is labeled as asynchronous design because the logic of triggering the read and write signal inside CRDWR block is not governed by clock. The logics inside CRDWR block are to handle the command inserted which is the data to be passed to the CSEQ and CNT block when read or write signal occurred. Therefore, the command such as which mode to be programmed, which counter to be selected, and binary/Binary-Coded Decimal (BCD) select is fully depends on the event of read and write signal occurred.

In overall system view, there is a block outside the 8254 PIT that generates the read and write signal as described in Figure 2.4. The logic inside the read and write signal generator block is clocked by the system clock that operates at 33 MHz.

The 1.19 MHz clock which derived from a 14.32 MHz external clock and is synchronized through the 33 MHz system clock before it is supplied to the 8254 PIT.

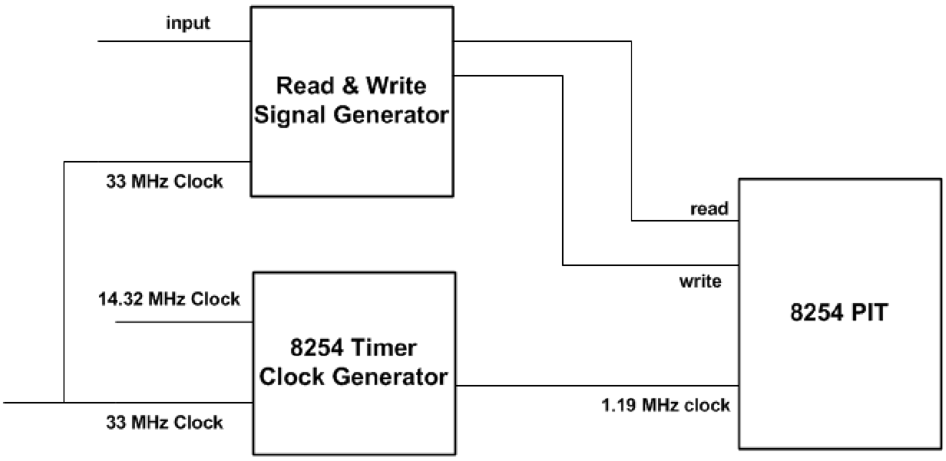

Figure 2.4: Overview of 8254 PIT clock system generation

A state machine is sat under the Read & Write Signal Generator block which control the generation of the read and write signal supplied to 8254 PIT. All the read and write operation of the 8254 PIT are tied up to this state machine which is shown in Figure 2.5.

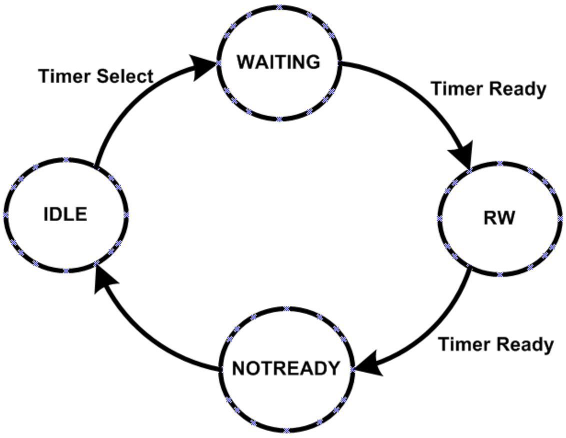

Figure 2.5: Read and write signal generation state machine

There are 4 states which are IDLE, WAITING, RW (Read and Write), and NOTREADY state. The state machine is always in IDLE state when the 8254 PIT is not in used. When a Timer Select signal is triggered, the state machine will go from IDLE to WAITING state. A counter inside this block will count from 0 to 3 and a Timer Ready signal will be generated at count 3. The state machine will go from WAITING to RW state after Timer Ready signal is triggered. This is the state where the read or write signal is generated. The counter will count from 0 to 3 again to generate another Timer Ready signal. This second Timer Ready signal will move RW state to NOTREADY state. After two clock cycles of NOTREADY state, the state machine will move back to IDLE state again. Figure 2.6 below shows the waveform of the state machine transaction. This whole process is to generate a read or write pulse to be used in 8245 PIT.

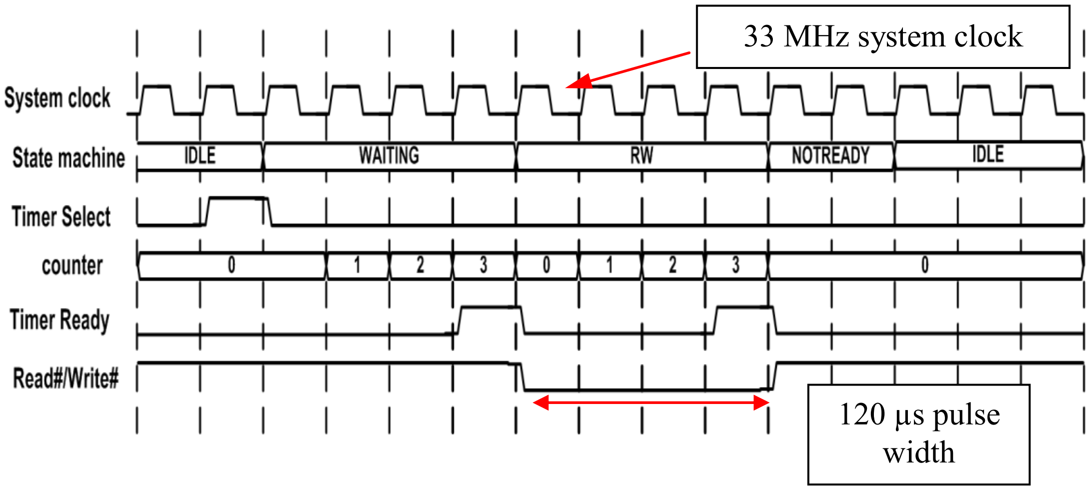

Figure 2.6: Waveform description of the read and write state machine

2.3 Digital Design Flow

In the process of designing a product, a lot of process is involved starting from the design of the Register Transfer Logic (RTL) until the product is taped in. The general design flow is shown in Figure 2.7.

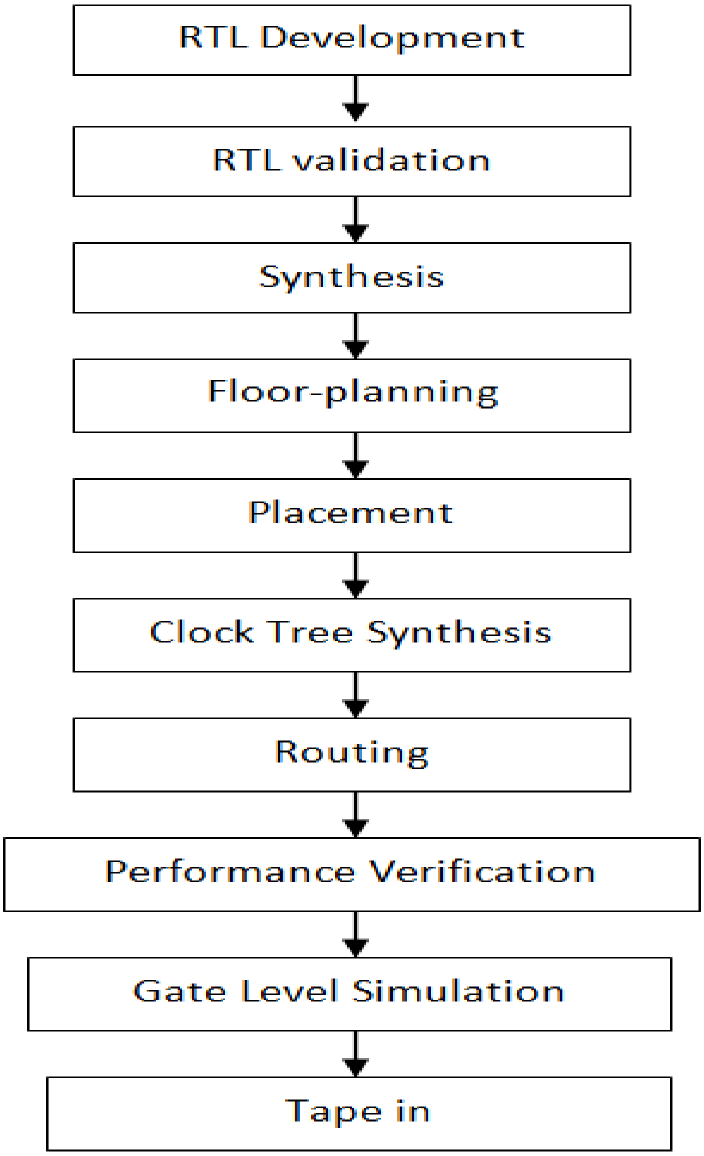

Figure 2.7: General Design Flow

On the first stage of the design, it will be the development of RTL. This is where all the coding of the design being done based on the design architecture and the specification required. All the RTL are coded in either, VHDL (VHSIC Hardware Description Language), Verilog, or SystemVerilog (SV) language which is the extension of Verilog language.

Validation is always an important process in a design flow. Without validation, a design will highly prompted to have errors due to some bugs in the design. In the process of validation, a testbench which contains the entire verification component is needed to be built in order to drive and validate the design (Nicolescu et al. 2002). All the function of the design based on the specification will be tested to ensure the design behave as expected.

Synthesis is a process of converting the RTL into logic gates implementation called netlist (Yunsup Lee 2009; Derek Lockhart 2012). After the RTL get synthesized into netlist, there will be floor-planning and placement process where the planning of the functional blocks to be placed in schematic. Clock Tree Synthesis (CTS) will be taken place after that with all the buffer of the clock is placed to minimized the skew and delay (Rajaram & Pan 2011). After CTS, routing will be done to connect the components placed and route to specify metal layers with respect to the design rules.

As mentioned in previous chapter, Performance Verification (PV) is a pre-silicon activity where the circuit timing is assessed with timing analysis tools to determine attainable frequencies (Holt et al. 2009). The minimum and maximum delay of the circuit is calculated during PV by using Static Timing Analysis (STA) tools to ensure the design can operate at different PVT. These delays will be used in Gate Level Simulation (GLS) to mimic the behavior of the actual silicon for verification purpose. The only difference between RTL verification and GLS is the netlist used. Netlist is used instead of RTL in the GLS model with timing information of the design included. GLS is run before tape in as this is the last chance to verify the design mainly on clock domain crossing and asynchronous unit due to the timing information of asynchronous unit are not handled during PV. Lastly, tape in is occurred after GLS when the layout schematic is passed to the foundry to process out a silicon.

In this research, the development of synchronous design of 8254 PIT which is more to RTL development will be focused. A lot of understanding is needed on the design itself in order to modify the existing asynchronous based design to synchronous based without changing the behavior and functionality of the design. An existing testbench will be used to validate the implemented synchronous design to ensure the design works correctly.

2.4 Static Timing Analysis

Static Timing Analysis (STA) is a methodology of computing the expected timing of a circuit without running the simulation. In the industry, there is a tool named PrimeTime and it is developed by Synopsys. Primetime is a full chip static analysis tool containing STA methodology that can fully analyze a multimillion gate Application-specific Integrated Circuits (ASIC) in a short time (George Michael

2006). The main advantage of STA tools is that it has ability to analyze a design over various PVT variations by exercising all the possible critical path and check that each path meets the setup and hold requirement to ensure there is no timing violation (Onaissi et al. 2011).

PVT variation is always the main concern for a real chip as it is a deciding factor whether a chip can operate under different condition or environment. Process variation is accounts for deviation in the fabrication process where the die is manufactured in different condition such as pressure, temperature, and doping concentration. Different type of process will cause the propagation delay of the chip to be varies (Verma et al. 2011). The best process will have lesser delay whereas the worst process will have larger delay as shown in Figure 2.8.

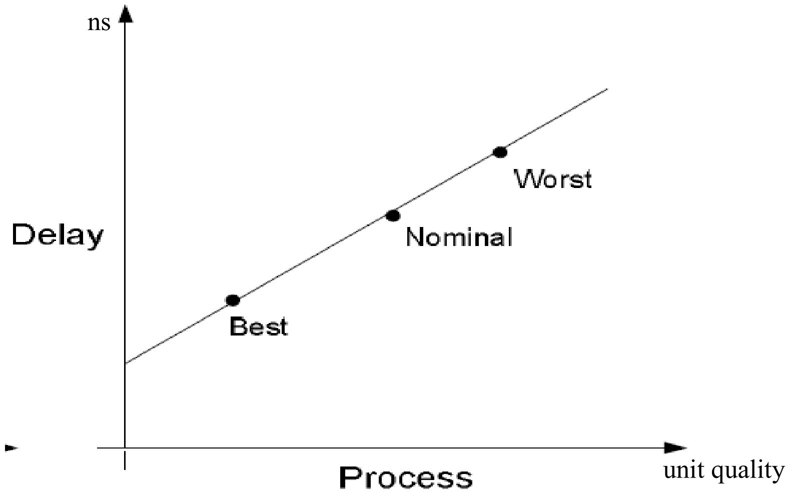

Figure 2.8: Delay versus Process

Higher voltage can make a cell faster and thus reduced the propagation delay. Figure 2.9 shows the delay of a chip varies at different voltage supply where low voltage will have worst delay and high voltage will have the minimum delay.

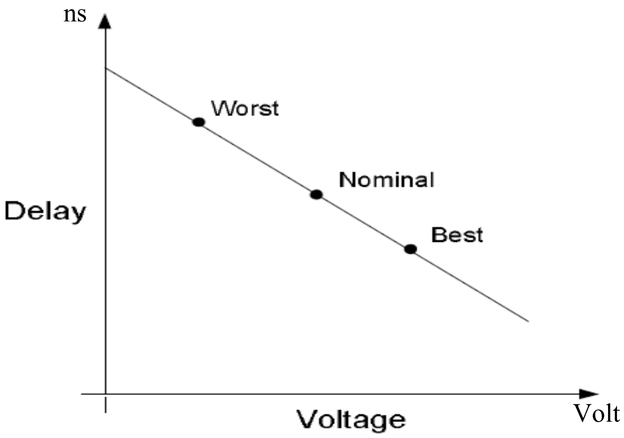

Figure 2.9: Delay versus Voltage

Temperature is also one of the factors that influence the variation of the propagation delay. Figure 2.10 describes the delay increased when the temperature increased.

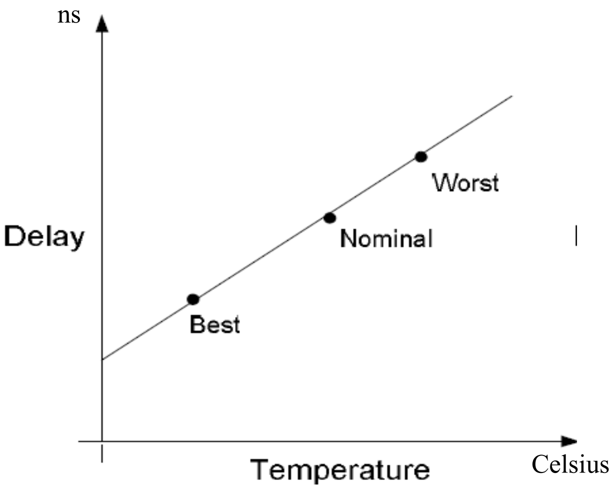

Figure 2.10: Delay versus Temperature

In STA, there will be 2 different corners which is min and max corner being run. Min corner, which also labeled as best corner will have the least delay. This means that it will have best process, high voltage supply and lowest temperature. On the other hand, max corner which is the worst case delay will have worst process, lowest supply voltage, and highest temperature. By ensuring the chip does not fail at these min and max corner, it will boost up the guarantee of the chip to be functioning correctly in real world.

Despite the excellent timing analysis of STA tool, the modern world still unable to replace simulation completely with STA due to the limitation of STA. STA is unable to handle asynchronous design such as path without clock or combinational timing loop. Asynchronous path is the path where the register is not clocked by a system clock. A combinational timing loop is a path that formed when a signal can reach back to itself (feedback) without encountering any sequential device such as flip-flop along the path (Sanjay Churiwala et al. 2010). This result in asynchronous design is not compatible of using this STA tools and will need to use another timing verification methodology called GLS.

2.5 Gate Level Simulation

Gate Level Simulation (GLS) is one of the validation flows which the simulation is done on the netlist instead of RTL. The only difference between GLS model and functional model is that GLS has cell delay included into the simulation. GLS is always used as the final stage to catch the design bug before fabrication. GLS is used for both functional simulation and dynamic timing analysis and it is important to the overall design flow as STA cannot cover the asynchronous logic in the design.

Unit block of 8254 PIT is a small unit inside the chip of Platform Control Hub (PCH) where it is described as full chip. The 8254 PIT is located under a cluster which contained some of the other IP such as Low Pin Count (LPC), System Management Bus (SMB), and General Purpose Input Output (GPIO) are located. Under the System On Chip (SOC), it contains a lot of clusters including the cluster as mentioned and other huge cluster such as Universal Serial Bus 2 (USB2), Peripheral Component Interconnect Express (PCIE) and Serial Advanced Technology Attachment (SATA). Figure 2.11 shows a diagram that describes the location of the 8254 PIT which sits under the wrapped cluster.

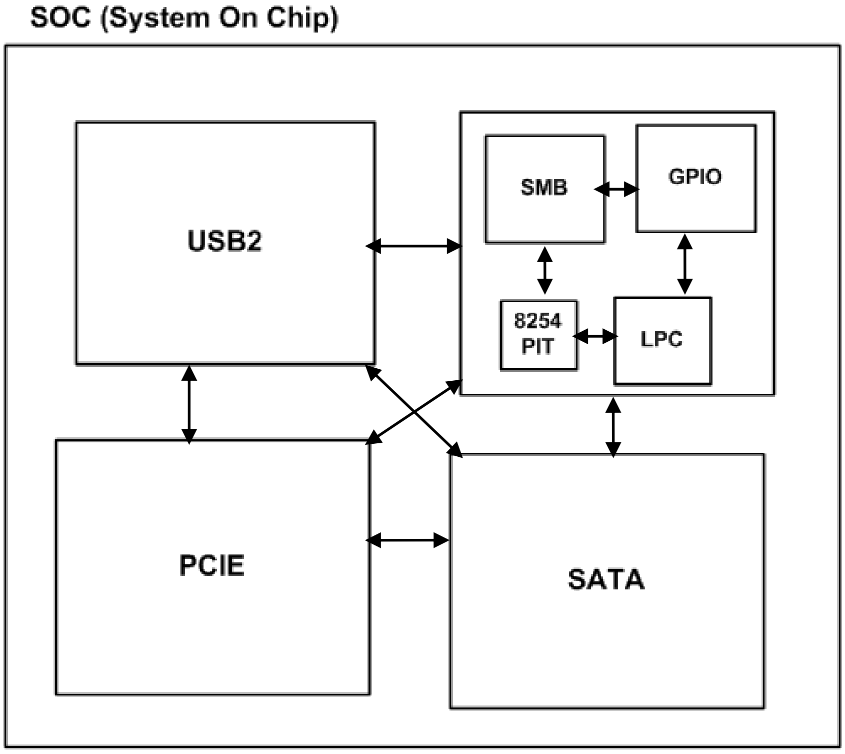

Figure 2.11: SOC structure

In the past, GLS was only run at the full chip level. GLS model is built using the full chip netlist with the timing back-annotated from the Standard Delay Format

(SDF) file. With process technology continuously shrinking, many more features are added into the design due to more area available and thus causes the gate count increased drastically. This result in the accommodation of full chip GLS to the current design becomes harder in term of efficiency, capacity, performance, and cost.

With the extremely long throughput time at full chip GLS, the efficiency or coverage of the GLS will be lesser because only some higher priority test was run prior to tape-out, which will cause potential hole in validation coverage. Besides, huge capacity of physical memory is needed to run full chip GLS because of the gate count increased, therefore the cost to enable full chip GLS is also increased. In terms of performance, high memory requirement and design complexity have caused extremely slow GLS speed (Suresh 2008). In average, a full chip test with 100 μs simulation time needs 4.5 hours to simulate. This requires huge debugging effort and time needed.

Due to the limitation at full chip GLS, a methodology regarding running GLS at cluster level has been introduced (Keng Hoo Goh et al. 2006). The netlist was extracted from the full chip level and a cluster model GLS is built. The runtime of the cluster level GLS is significantly reduced compared to full chip level as only the design of one cluster is included.

With the introduction of cluster GLS, it has solved the capacity and cost problem faced at full chip GLS. However, there are still plenty of disadvantage of cluster GLS. To build a cluster GLS model, a script is developed and run to extract the full chip netlist into cluster netlist. After that, another script is run to connect the