DESIGN OF 8 BITS PARALLEL-TO-SERIAL CONVERTER FOR BASEBAND USING VHDL

Oleh

Ruhaifi b. Abdullah Zawawi

Disertasi ini dikemukakan kepada
UNIVERSITI SAINS MALAYSIA

Sebagai memenuhi sebahagian daripada syarat keperluan untuk ijazah dengan kepujian

SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)

Pusat Pengajian Kejuruteraan
Elektrik dan Elektronik
Universiti Sains Malaysia

ABSTRACT

This report included code writing, schematic, layout and simulation result for front end and backend process. This design is actually done for digital signal processing and still need more effort in research since this project is still new in our university. Front end process consists of behavioral description, RTL description and gate level description. Synthesis is done throughout this process to specify the maximum frequency for clock signal. This design has achieved frequency of 204MHz at typical process with 162238.88um^2 in total area and only 0.0294mW in power consumption. Our target is to get high speed system to produce high data rate. We also want to reduce the power consumption especially the dynamic and static power. For front end, the design had been carried out for behavioral description, synthesis and verilog in for gate level simulation. Then, the gate level netlist will be imported for back end process included its timing constrain. A simple floorplan is generated, blocks or cells are placed, power routing is done and all analysis for timing, violation ,clock tree and fixing all problems in our layout . Our target is to get GDSII file with no violation in setup and hold time, also no signal integrity problems. For back end analysis, there are no violation since the slack minimum slack value is 6.553ns for rising edge signal and 7.739ns for falling edge signal. The last process is very important because the parasitic resistor and capacitance have been considered so that the timing and power analysis will be more accurate. In addition, clock tree synthesis is carried out to arrange all standard cells at certain place in the core so that the clock signal will arrive at same time for all cells.

ABSTRAK

Tesis ini menunjukkan cara menulis kod, gambarajah litar hasil daripada proses penterjemahan kod kepada litar dan bentangan litar tersebut. Projek ini adalah untuk pemprosesan digital dalam cip dan masih memerukan usaha secara mendalam kerana projek ini masih baru di Universiti Sains Malaysia. Proses pertama dalam projek ini adalah menulis kod VHDL dan membuat simulasi terhadap kod tersebut. Untuk menguji kod tersebut,isyarat masukan untuk setiap data input perlu di berikan dan ini dilakukan dengan menulis satu lagi kod yang dinamakan ‘testbench’. Kemuadian, setelah mendapat hasil simulasi yang betul, kod tadi ditukar kepada litar . Seterusnya, frekuensi maksimum untuk sistem yang dibina dicari. Frekuensi ini yang akan menentukan bilangan data yang akan keluar sebagai output bagi setiap saat. Frekuensi maksimum untuk keadaan tipikal adalah 204MHz dengan keluasan 162238.88um^2 . Analisa untuk kuasa juga dilakukan dan litar perlu diubah jika kuasa lesapan yang tinggi berlaku, ini dilakukan dengan meletakkan penimbal pada setiap masukan dan keluaran. Kuasa keseluruhan, 0.0294mW yang diperoleh akan digunakan untuk menentukan lebar cincin kuasa pada bentangan. Kemudian, litar tadi ditukar kepada bentangan menggunakan ENCOUNTER untuk mendapatkan fail GDSII. Proses terakhir ini penting untuk mendapatkan nilai sebenar bagi analisa masa, kuasa dan keluasan cip. Analisa ini akan mengambil kira semua nilai rintangan dan kapasitor bagi seluruh sambungan untuk setiap get logic. Setiap sel disusun pada setiap lapisan bentangan supaya jarak antara sumber isyarat jam dengan semua sel adalah sama, ini bertujuan supaya isyarat jam diterima pada masa yang sama untuk setiap sel.

ACKNOWLEDGEMENT

Throughout the whole semester doing the final year project, I have truly learned a lot of digital implementation and at the same time enrich my knowledge. Thanks to Allah for giving me a strength to finish my final year project in this semester before I begin my working life. I must also thank to school of Electrical and Electronic, University Science Malaysia for providing this project.

I would like to express my gratitude to many people who have made my undergraduate studies possible and made it such rewarding experience. First and foremost, my greatest honor and appreciation go to Dr Tun Zainal Azni Zulkifli, my supervisor as well as lecturer, for his tireless desiccation and effort in guiding me to complete this project. It has been privilege to be a part of his group.

Thank also to all my friend for their time and guidance which inspire me in working my project. Their guidance, encouragement and suggestions have greatly influence the success of this project.

Next I would like to thank to my family who have supported me through this year. Thank to my mother and father who had always encourages me in academic fields.

My acknowledgments would not be complete without expressing my appreciation to all members of University Science Malaysia

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

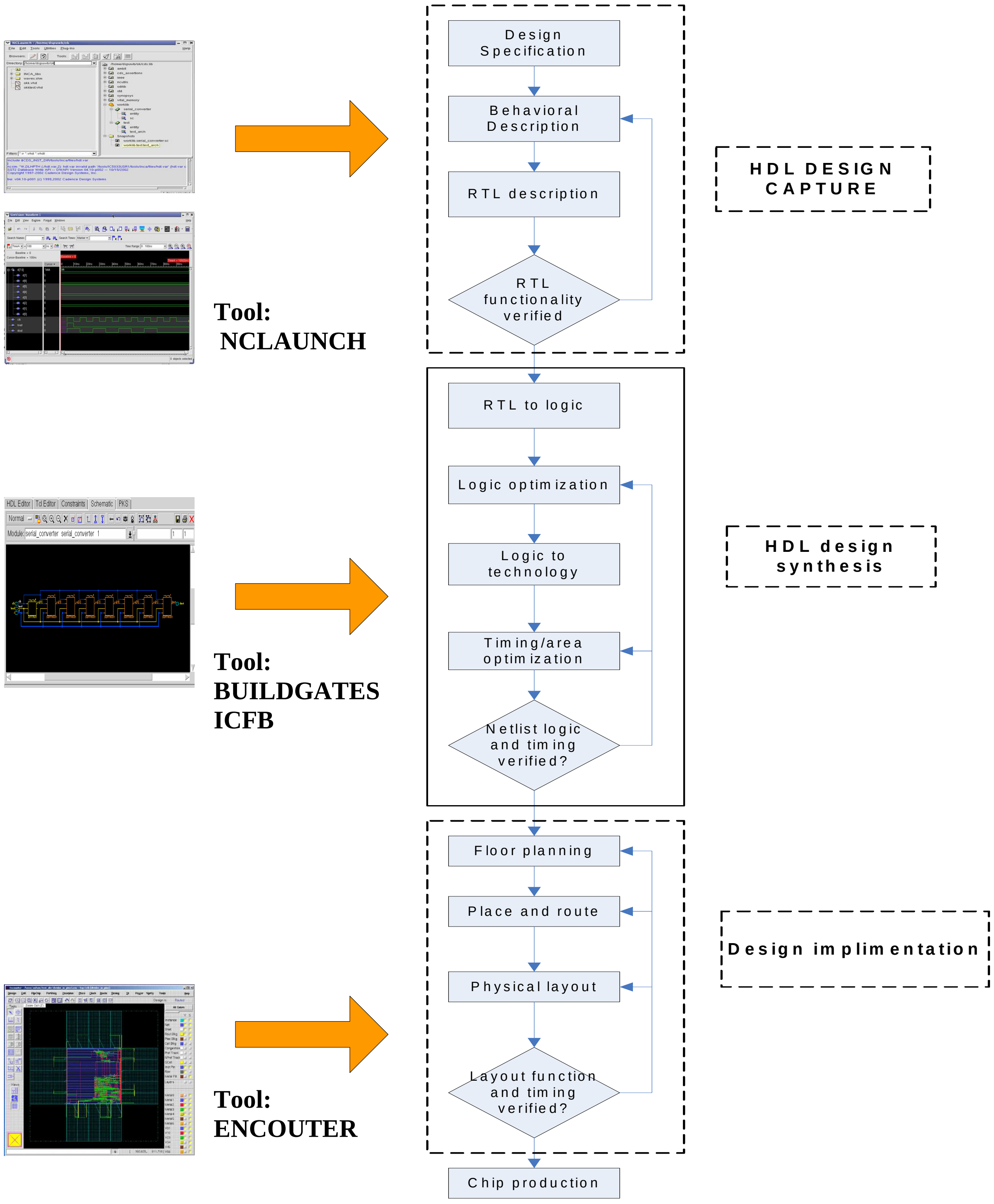

Figure 1.1: Process flow chart for ASIC design

1.1 Introduction To ASIC Design

Application Specific Integrated Circuits(ASIC), is used to design entire systems on a single chip. ASICs are interconnect of standard cells which have been standardized by fabrication foundry. With the integration of more and more system components on single IC, the complexity of IC fabrication has increased. Modern day system design involves complex layout issues. Specifications of cells are provided by the vendors in form of Technology library which contains information about geometry, delay and power characteristics of cells.

1.1.1 Overview process of ASIC design

For this design approach, we will first write a description of how the circuit behaves. In the most simple case, for a given set of inputs we will specify what the outputs will be. A compiler tool will translate the behavioral description into a gate-level description of the circuit, or netlist to synthesize the behavioral description. The circuit will be compiled with pre-defined cells from a standard cell library. The standard cell library will contain the electrical characteristics of the circuit, timing information, and layout information for each gate. Next, the synthesized netlist of the circuit is read into Virtuoso, where it will be represented as a schematic. At this point, the circuit may be simulated with verilog-XL to verify correct operation. Next, a layout is generated from the schematic. The first step of this is to generate and place the cells. The next step is to connect them together. This can be done with an automated tool with the options to minimize certain characteristics such as area, power, or delay. From the layout, parasitic components may be extracted, and the circuit re-simulated. If specifications are not met, changes to the layout may need to be made in order to correct the problem. Finally, the layout can be converted into a GDSII file for fabrication.

1.2 Introduction To Digital Signal Processing

Digital Signal Processor(DSP) a special-purpose CPU used for digital signal processing applications. It provides ultra-fast instruction sequences, such as shift and add, and multiply and add, which are commonly used in math-intensive signal processing. DSP chips are widely used in a myriad of devices, including cellphones, sound cards, fax machines, modems, hard disks and digital TVs. The first DSP chip used in a commercial product was believed to be in the very popular Speak & Spell game, introduced by TI in the late 1970s. A category of techniques that analyze signals from sources such as sound, weather satellites and earthquake monitors. Signals are converted into digital data and analyzed using various algorithms such as Fast Fourier Transform. Once a signal has been reduced to numbers, its components can be isolated, analyzed and rearranged more easily than in analog form. DSP is used in many fields, including biomedicine, sonar, radar, seismology, audio, speech and music processing, imaging and communications. DSP chips are specialized integrated circuits for digital signal processing applications.

1.3 Introduction To OFDM Transceiver

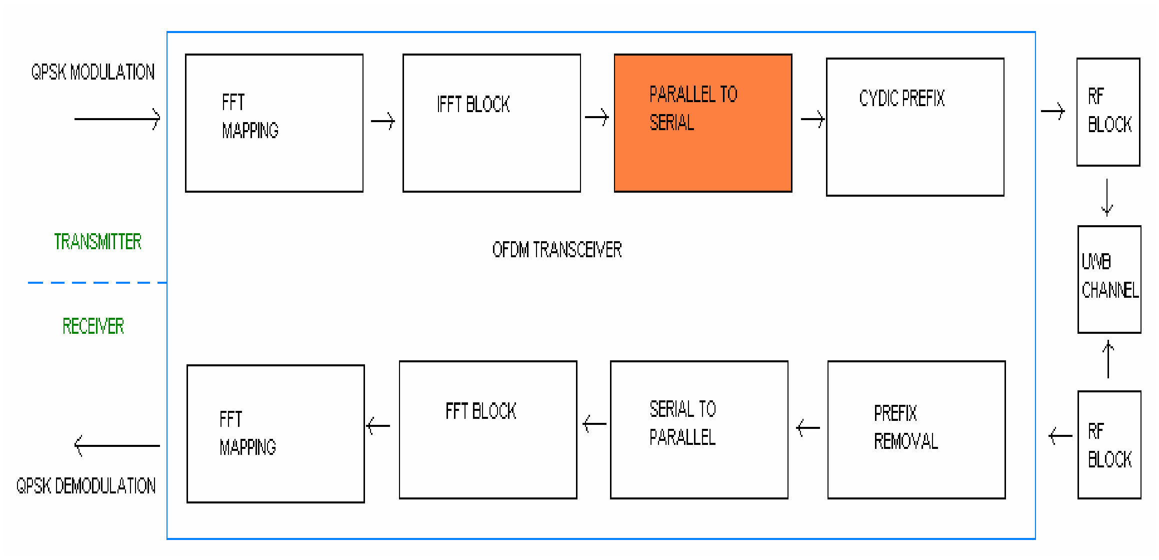

Figure 1.2: Multiband OFDM Diagram

The basic idea of multicarrier is to divide the transmitted bit stream into many different sub streams and send these over many different subchannels. Typically the subchannels are orthogonal under ideal propagation conditions, in which case multicarrier modulation is often referred to as orthogonal frequency division multiplexing(OFDM). QPSK modulation in **Figure 1.2** above is used to increase the data rate without increasing the bandwidth by encoding more than one bit per phase change. The FFT Mapping and FFT Block is then receiving the data and improve the performance of line coding. In serial transmission one bit follows another, so we need one communication channel rather than more than one to transmit data between two communication devices. . The cyclic is added to the output of parallel to serial converter to induce circular convolution of channel frequency response. This project is actually emphasized more on designing parallel to serial converter using VHDL codes and finally gets the layout for ic fabrication

1.4 Objective Project

The project objective is to design 8 bits parallel to serial converter for baseband using VHDL. The performance of the design is analyzed so that the high speed clocking system is achieved. Our goal is to get high data rate system with low power consumption

1.5 Report Organization

For ease of understanding, this part will summarize all chapters in this report. Chapter 1 will introduce overview of ASIC design as depicted in **Figure 1.1**. This is included methodology for digital IC design. According to this, basic operation of OFDM transceiver is introduced so that we know the operation of every blocks in the system. Chapter 2 is then described in detail the operation of parallel to serial converter. Timing analysis is a must for IC design and this topic is described in the following chapter. Nowadays, chip designers are competing with the others to reduce the power consumption in the chip. Chapter 4 will describe some approaches to reduce the power. Design implementation is mentioned in chapter 5 and followed by result and discussion in chapter 6.

.

CHAPTER 2

OPERATION O F PARALLEL TO SERIAL CONVERTER

2.1 Introduction

This chapter will introduce the concept of parallel to serial converter. The data from IFFT will be traveling to serial converter simultaneously in parallel and then converted into serial data. The next section will describe the process of converting to serial data.

2.2 Project Description

The idea is described below:

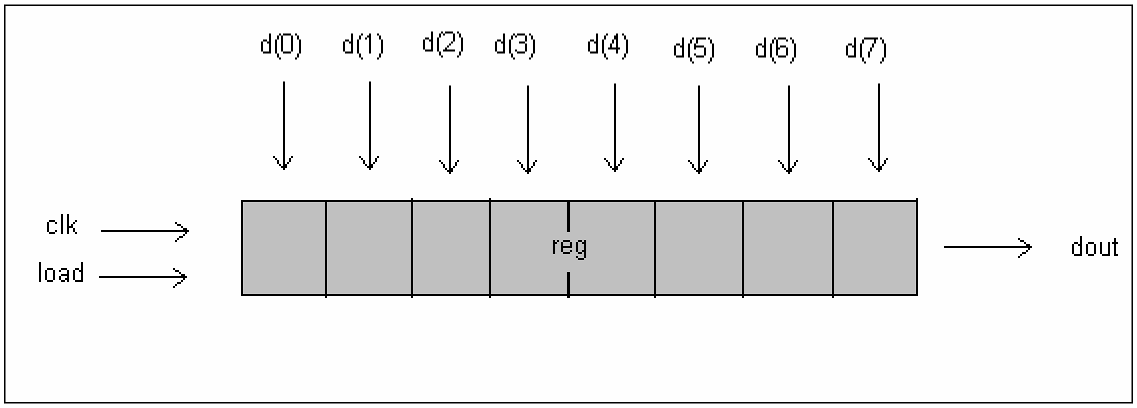

Figure 2.1:Parallel to Serial Converter Diagram

A parallel to serial converter is a typical application of shift registers. It consists of sending out a block of data serially. The need for such converters arises, for example in ASIC chips when there is not enough pins available to output all data bits simultaneously.

A diagram of parallel to serial converter is presented in figure above d(7:0) is the data vector to be sent out, while dout is the actual output. There are also two other inputs: clk and load. When load is asserted, d is synchronously stores in the shift register reg. while load stays high, the MSB, d(7), remains available at the output. Once load is returned to ‘0’, the subsequent bits are presented at the output at each positive edge of clk. After all eight bits have been sent out, the output remains low until the next transmission.

CHAPTER 3

DIGITAL SYSTEM CLOCKING

3.1 Introduction

Clocking is an important aspect of digital system design. The performance of our system depends on our clock. High speed clock will produce high data rate but we need to avoid the highest negative impact on the reliability of an improperly designed system. This chapter is very important especially for synthesis. We will go through some timing analysis for a simple pipeline system. Before that, this chapter will explain some of important parameters we used in our synthesis.

3.2 Timing Parameters

Timing parameters are very important to look into such as clock to output delay, setup time, hold time and minimum pulse width. Using standard cells from Artisan library, all timing parameters have been set up, so the timing analysis is performed according to the timing parameters. Propagation delay is the time of the data at the output change after an input signal has been applied. This parameter value is used to calculate the minimum clock period. Every standard cell will has it setup and hold time. Setup time is the minimum interval required for the logic levels to be maintained constantly on the inputs prior to the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop. Hold time is The minimum interval required for the logic levels to remain on the inputs after the triggering edge of the clock pulse in order for the levels to be reliably clocked into the clocked.

3.3 Analysis of Synchronous System

Clocking is one of the most important decisions that must be considered by the designer. Nowadays we have achieved a very high speed system and of course the clock is the dominant problem if the clock uncertainties or clock skew have not been scaling proportionally with the clock frequency. Clock skew is actually the different in arrival time for two or more clock pins in synchronous system because of differences in line lengths from the clock source to the clock register. It will effect the performance of our system in positive and negative aspect. To solve this problem, clock tree analysis is performed. When we design a big system with a lot of sub modules running at the same time in every clock cycle, of course the accuracy of clock system is very important. Clock tree is actually designed to meet this purpose. Let us analyze the circuit below for ideal and non-ideal condition.

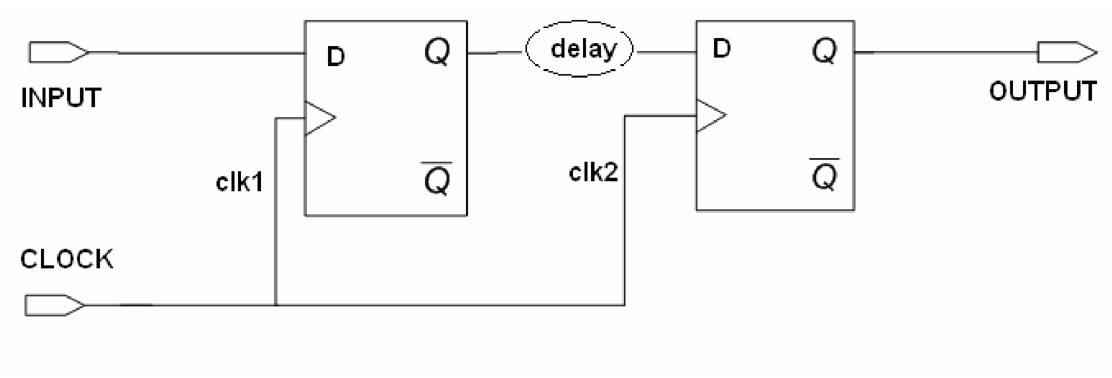

Figure 3.4: Synchronous system

Assumption

For this analysis, we make some assumptions as follow:

1. Hold time, t_h = 2ns
2. Setup time, t_s = 2ns
3. Propagation time for both flip flop , t_{cq} = 3ns
4. delay = 2ns
5. skew, t_{skew} = 1ns
6. Clock period,T

In order to know the performance of the system, we need to find out the maximum frequency for the system. Furthermore, the clock has to be active for some minimum amount of time in order to ensure reliable capture of data. We will go through this analysis for ideal and non ideal condition.

Case 1: ideal condition

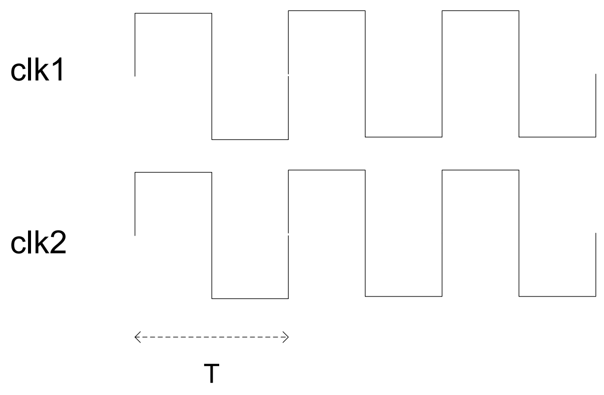

Figure 3.5: Clock signal at both flip flop

It can be seen from **Figure 3.5** above that the clock signal arrives at same time for both flip flop. To determine the clock period ,we use the following equation:

$$T + tskew \geq tcq + tdelay + ts \qquad \textbf{(3.1)}$$

In this case, t_{skew} = 0ns

Thus, $T \geq 3 + 2 + 0 + 2\,\mathrm{T}$

$\geq 7ns$

So, the maximum frequency for the system is

$$F = \frac{1}{T} \qquad \textbf{(3.2)}$$

$$= \frac{1}{7ns}$$

$$= \boldsymbol{142.86MHz}$$

The maximum frequency allowed for clock is 142.86MHz . If we apply clock frequency bigger than 142.86MHz, lets say 150MHz, violation will occur because data does not meet it setup time:

$$T = \frac{1}{F}$$

$$= \frac{1}{150MHz}$$

$$= \boldsymbol{6.67ns}$$

Remain time for setup time is = 7 – 6.67

= **0.33ns**

Obviously, this value is less than 2ns (setup time). Consequently, violation will occur because data does not meet it setup time.

Case 2 : non ideal condition (positive skew)

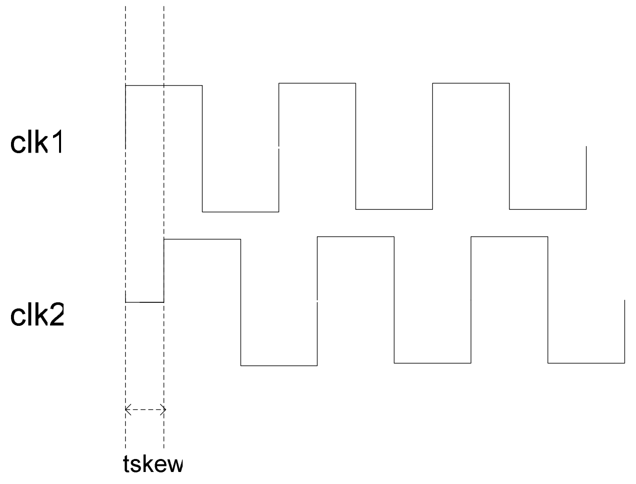

Figure 3.6: Skew at clk2

In this case, we see that clk2 arrive 1ns after clk1. Using equation (3.1), the following value we get for clock period:

$$T + 1 \geq 3 + 2 + 2$$

$$T \geq \boldsymbol{6ns}$$

We actually improve the performance because we increase the clock frequency but make t_{hold} harder to meet because t_{skew} seems increasing the hold time. Let us figure out why t_{hold} harder to meet.

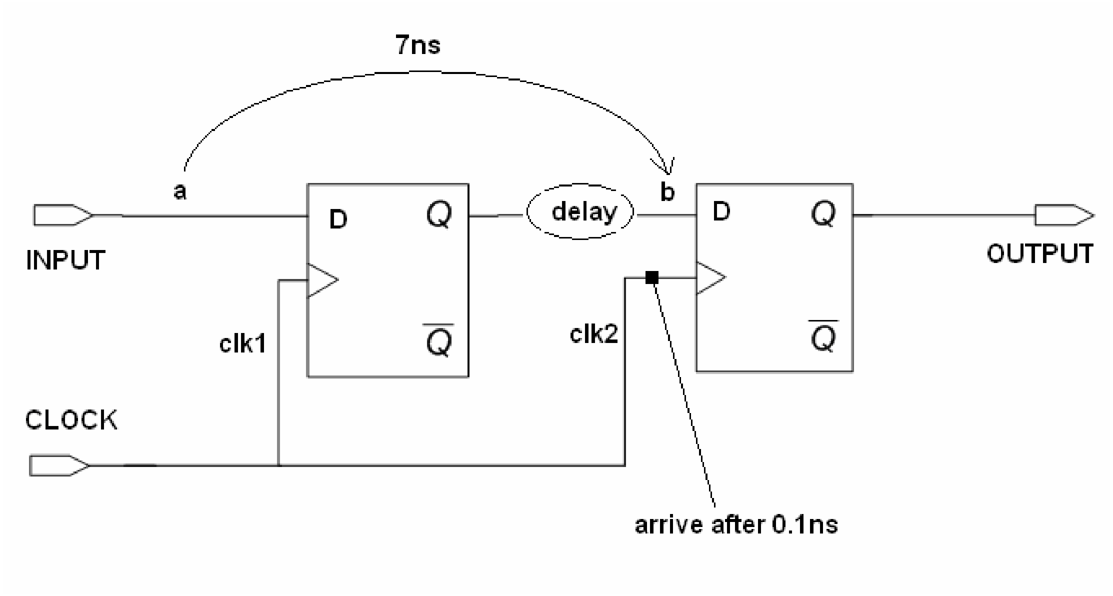

Figure 3.7: System with positive skew

Propagation time from point *a* to point *b* is 7ns. Clock signal for clk2 arrive 1ns after clk1. From definition, hold time is minimum time for data at point *b* to remain constant at rising edge of clock before new data from point *a* arrives. For this case, the clock skew causing the signal at clk2 arrives 1ns late, so the data at point *b* facing additional time before the clock signal rising. Now, the signal at point *b* need 3ns to meet it hold time. Imagine if we have a very short propagation time for flip flop 1,lets say 2ns,and no delay from output of flip flop 1 to input of flip flop 2, so the data will propagate from point *a* to point *b* of about 2ns. Obviously, violation will occur for hold time because the data at point *b* does not has enough time to remain at point *b* for 3ns.Although we increase the speed of the system, see equation below for clock period, we are actually getting hold time violation. Using equation (3.1):

$$T + 1 \geq 2 + 2$$

$T \geq \boldsymbol{3ns}$ (previous is 6ns)

Case 3 : Non ideal (negative skew)

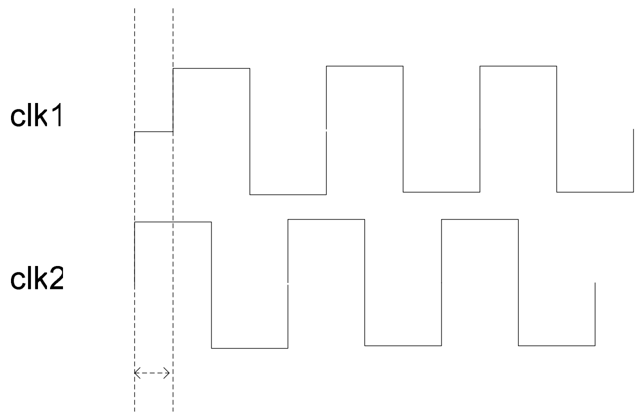

Figure 3.8: Skew at clk1

In this case, clk1 arrives 1ns after clk2. Using equation (2.1), the following value we get for clock period:

$$T - 1 \geq 3 + 2 + 2$$
$$T \geq 8ns$$

We actually decreased the performance because we decrease the clock frequency but make t_{hold} easier to meet .Again, we figure out why t_{hold} is easier to meet.

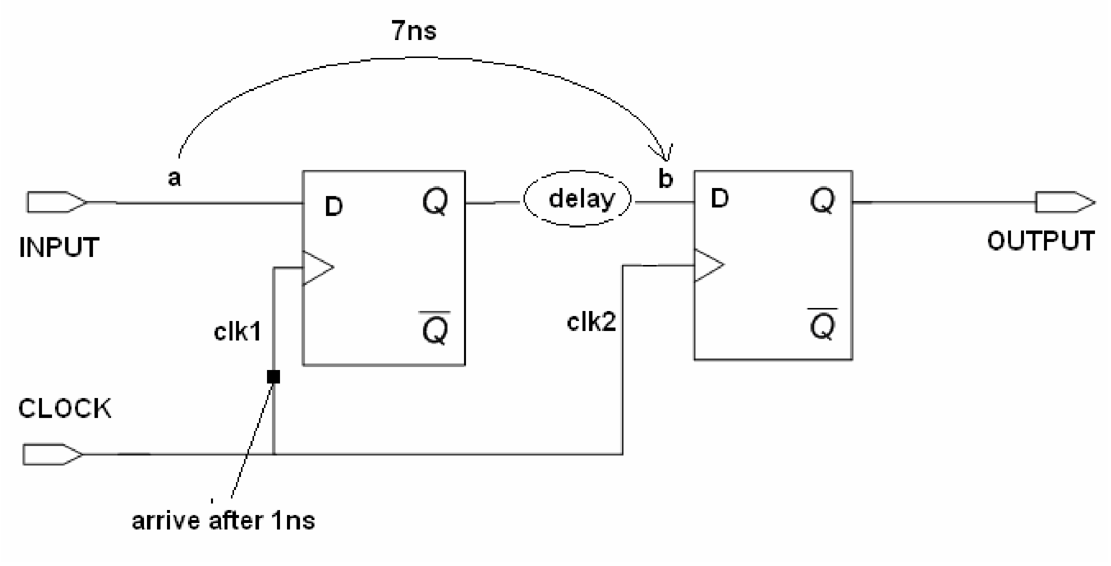

Figure 3.9: System with negative skew

The data at point *a* will start traveling after 1ns because of clock skew at clk1. Consequently, it will arrive 1ns late at point *b*. Obviously we give another 1ns for data at point b to meet it hold time. That is why t_{hold} easier to meet.

Case 4 : non ideal (clock jitter)

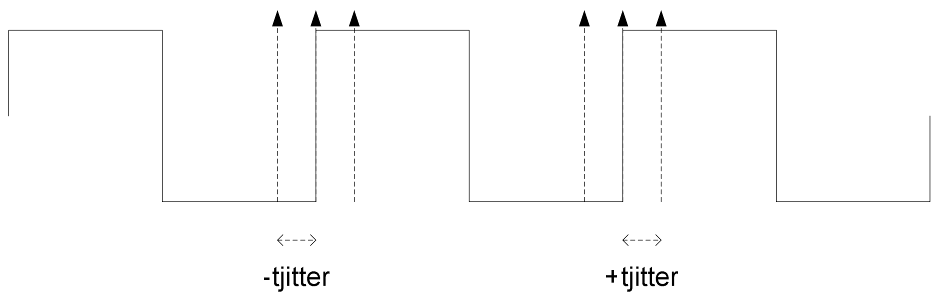

Figure 3.9: Clock jitter

Because of high speed logic design, we are more concern about cycle to cycle clock jitter, because it affects the time available to the logic. Figure 3.9 shows the effect of clock jitter to the clock signal. Jitter directly reduces the performance of a sequential circuit because we increase the clock period. The following equation defines the minimum period we can apply to clock if jitter happens.

$$T \geq tcq + tdelay + ts + 2tjitter \qquad \textbf{(3.3)}$$

CHAPTER 4

LOW POWER SYNTHESIS

4.1 Introduction

Today, low power is emerging as a critical issue in ASIC design. The primary driving factors are the remarkable success and growth of personal computing devices (portable desktops, audio- and video-based multimedia products), and wireless communications systems (personal digital assistants, pagers, and cellular phones). These technologies demand high-speed computation and complex functionality, making low-power consumption a crucial design concern. Another development that emphasizes the need for low-power solutions is the growing number of large, high performance designs. Millions of transistors switching at high clock speeds leads to power consumption problems that can be difficult to solve late in the design cycle. The exorbitant cost associated with packaging and cooling strategies for high-performance designs is yet another factor that makes it imperative to have a seamless low-power methodology.

4.2 Power Consumption

Power consumption for a circuit includes two general types of power,static and dynamic power.

4.2.1 Static Power Dissipation

Static power is generally dependent on the state of the cell. Static power dissipation can be defined as power that is lost while a circuit's signals are not actively switching. Static power dissipation includes leakage power caused by sub-threshold leakage current in CMOS circuits and standby power dissipation caused by the DC current being continuously drawn from power to ground

4.2.2 Dynamic Power Dissipation

The overall dynamic power of a cell is based on the effects of voltage and temperature, load, input slew rate, as well as the cell's state. Dynamic power dissipation is defined as power lost while a circuit is actively switching at a given frequency. Dynamic power dissipation includes two types of power. First is short circuit power which is dissipated when a short exists between the voltage supply and ground rails created during output transitions that cannot be instantaneously turned on and off. Second is switching power or capacitive load power which is the power required to switch the entire load. The capacitive load includes internal capacitance, capacitance of the net, and the capacitance of the input pins. Power consumed by the internal capacitance is modeled as part of the cell power.

4.3 Clock Gating

In many design situations, data is infrequently loaded into registers, but the clock signal continues to switch at every clock cycle, driving a large capacitive load. This makes the clock signal a major source of dynamic power dissipation. Significant amounts of power can be saved, by identifying periods of inactivity in the registers and disabling the clock during those periods.

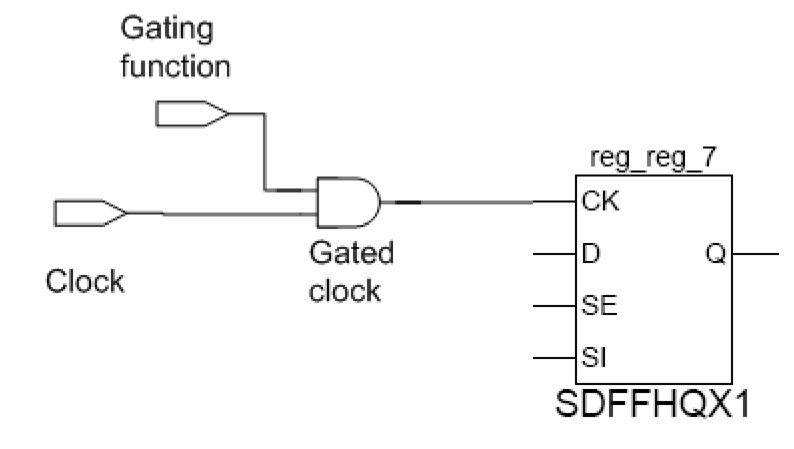

Figure 4.1: Clock gating

4.4 Gate Sizing

Gate sizing is the process of changing the CMOS gate size, smaller or larger, so that the total power is minimized without violating timing constraints. The goal is to find the gate size that consumes the least power. In general, reducing the size of a gate leads to a decrease in the power and an increase in the delay. The overall internal cell power of a gate consists of leakage power ,short-circuit power and power consumed by the parasitic capacitance

Sizing the gate reduces the short-circuit power and power consumed by the parasitic capacitance. Gate sizing could, however, increase the short-circuit power of the fanout logic because the slew of the output signal could increase when this transformation is performed. LPS makes tradeoffs between the gate sizing and the increase of power due to slew in order to optimally reduce the power of the design.

4.5 Pin Swapping

Pin swapping matches nets connected to the symmetric pins so that power dissipation can be minimized. Pins with higher capacitance are matched to nets with lower switching activity. If not done carefully, pin swapping can actually increase the delay of the design. LPS works to find the optimal configuration that reduces power without affecting timing.

4.6 Slew Optimization

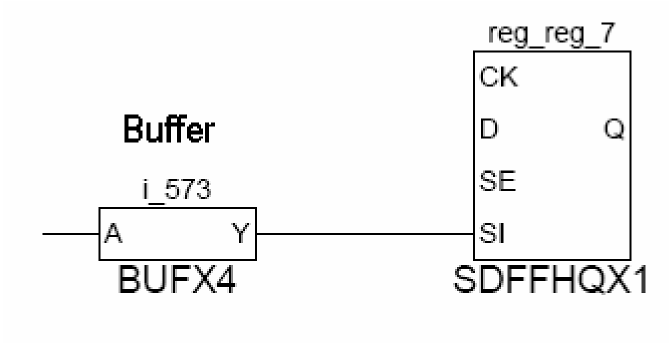

Figure 4.2: Adding buffer for slew optimization

The internal power of a cell depends largely on the slew of the input signals. During that time, power is drained from the design power rails to ground. If a signal has a large slew, driving the signal through a buffer improves the signal’s slew and reduces the overall power of the cells driven by the signal.

CHAPTER 5

DESIGN IMPLEMENTATION

5.1. Introduction

The Cadence Encounter family of products provides an integrated solution for an RTL-to-GDSII design flow. This phase involves floor planning. Placement of cells on the chip area. Placement of Input/Output pads on the chip area. Clock tree synthesis is performed in order to minimize power consumed by clock signal. Placement and routing is carried out on this design. So far the interconnect delays and parasitic values are based on wire-load model. Now resistance, capacitance and inductance (latest feature) is calculated for a placed and routed netlist. Design rule violations are identified and corrected. Static timing analysis is carried out to find timing violations.

5.2 VHDL Code for Design Description.

We start writing VHDL code for parallel to serial converter after we do design specification**.** To test the codes we write another code to generate clock, data and load to map all inputs terminal with a proportional input values. **Table 5.1** and **Table 5.2** describe the behavioral and test bench respectively for the design

Table 5.1: VHDL code for behavioral description

	CODE	COMMENT

Library	`LIBRARY ieee;`
	`USE ieee.std_logic_1164.all;`
		Library for coding

Entity	ENTITY serial_converter IS
	
	PORT (d:IN STD_LOGIC_VECTOR (7 DOWNTO 0);
	clk,load:IN STD_LOGIC;
	dout: OUT STD_LOGIC);
	
	END serial_converter;
		Define the input and output pins.
		
		Input: d,clk, and load
		Output: load

Architecture	ARCHITECTURE sc OF serial_converter IS
	
	SIGNAL reg: STD_LOGIC_VECTOR(7 DOWNTO 0);
	BEGIN
	PROCESS(clk)
	BEGIN
	IF(clk'EVENT AND clk='1') THEN
	IF(load='1') THEN reg<=d;
	ELSE reg<=reg(6 DOWNTO 0) & '0';
	END IF;
	END IF;
	END PROCESS;
	dout<=reg(7);
	END sc;
		Behavioral description for serial converter. See chapter 2 for detail.

Table 5.2: Testbench code

	CODE	COMMENT

Library	library ieee;
	use ieee.std_logic_1164.all;
	use ieee.std_logic_arith.all;
	use ieee.std_logic_unsigned.all;
		Library for coding

Entity	entity test is end test;	No input and output port

Architecture	architecture test_arch of test is
	
	component serial_converter
	port (
	
	d :IN STD_LOGIC_VECTOR (7 DOWNTO 0);
	clk,load :IN STD_LOGIC;
	dout :OUT STD_LOGIC
	);
	
	end component;
		Define the input and output pins.
		
		Input: d,clk, and load
		Output: load

	signal d : std_logic_vector(7 downto 0);
	signal clk : std_logic ;
	signal load : std_logic ;
	signal dout : std_logic;
		Signal declaration for inputs and output.

	begin	

	UUT : serial_converter
	      port map
	              (
	          d => d,
	        clk => clk,
	
	       load => lo
	       dout => dout
	                 );
		Map all inputs and output pins between behavioral description code and test bench code.

	d <= "10101010";	Input signal
	load <= '1' after 5 ns, '0' after 10 ns;	Load signal

	clk <= '1' after 5ns, '0' after 10 ns, '1' after 15 ns, '0' after 20 ns, '1' after 25 ns, '0' after 30 ns, '1' after 35 ns, '0' after 40 ns, '1' after 45 ns, '0' after 50 ns, '1' after 55 ns, '0' after 60 ns, '1' after 65 ns, '0' after 70 ns, '1' after 75 ns, '0' after 80 ns, '1' after 85 ns, '0' after 90 ns, '1' after 100 ns;
	
	end test_arch;
		Clock signal

Figure 5.1 shows the input signal we apply in the test bench. Clock frequency is 100MHz and load signal is triggering up after 5ns and return back to zero after 5ns. All data signal, d(0),d(1),d(2),d(3),d(4),d(5),d(6) and d(7) are remain constant for all time as depicted in the figure below**.**

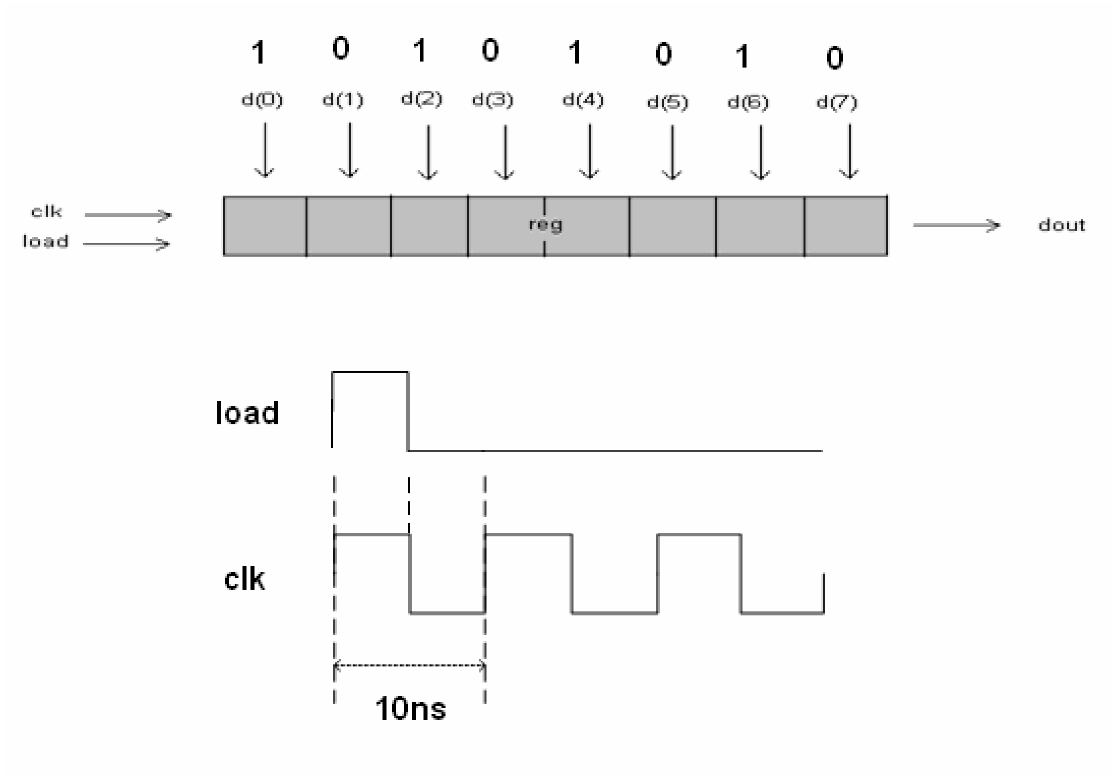

Figure 5.1: Data, load and clock signal

The design had been carried out for synthesis and behavioral description has been transformed to gate level description as shown in **Figure 5.2**.

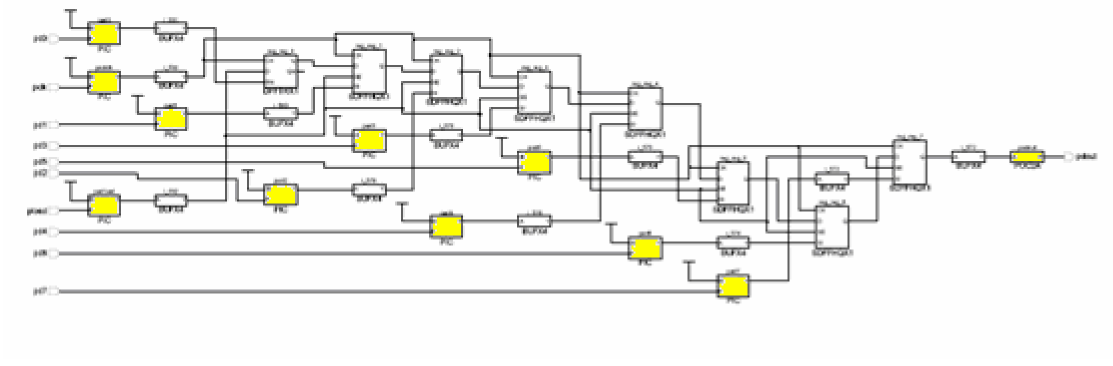

Figure 5.2: Schematic for parallel to serial converter