CHAPTER ONE

INTRODUCTION

CHAPTER TWO

METHODOLOGY

CHAPTER THREE

RESULT AND DISCUSSIONS

CHAPTER FOUR

CONCLUSION AND FUTURE WORKS

Ministry Of Higher Education
And Scientific Research
University of Diyala
College of Engineering
Electronic Department

Design and Simulation of Pipelined Phase Accumulator Using Verilog HDL Code

The Project Submitted to the Department of Electronic Engineering
University of Diyala in Fulfillment of the Requirements for the
Degree of Bachelor in Electronic Engineering

By

Narmeen Hassan Mirza
Sara Haseeb Burhan

Supervised by

Dr. Salah Hassan

2016

وزارة التعليم العالي والبحث العلمي
جامعة ديالى
قسم الالكترونيك

تصميم وتنفيذ المركم المرحلي وفق التقنية الانبوبية باستخدام لغة البرمجة الوصفية HDL

المشروع مقدم الى قسم الهندسة الالكترونية في جامعة ديالى كجزء من متطلبات نيل الحصول على شهادة البكلوريوس في الهندسة الالكترونية

ساره حسيب برهان
نرمين حسن ميرزا

باشراف

الدكتور صلاح حسن

2016

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴾

صدق الله العظيم

﴿سورة المجادلة ﴾(١١)

الإهداء

إلى ..
بستان الأمل والمحبّة الخضراء دوماً ...

(والدي الحنون)

إلى ..

من تحت أقدامها بُسطتْ لحنانها وعطائها سجادة الجنة ...

(أمي الحبيبة)

إلى ..

سلسلة الصدق والوئام التي ما انقطعت قلادتها من جيد حياتي طيلة مكوثي على كرسي " إقرأ "

(أصدقائي وصديقاتي)

إلى ...

رسل العلم والمعرفة والثقافة الذين نقف لهم لنفهم التبجيلا ، مقدرون لدورهم التعليمي ورسالتهم النبيلة ...

(أساتذتي الافاضل)

أهدي إليكم جميعاً شلال النور الذي قادني إلى منابر العلا ،

لأرى الحياة بعين العلم سماءً احتوت كياني

ABSTRACT

This project presents design and simulation of pipelined phase accumulator (PA). The 4-bit ripple carry adder (RCA) is combined with the pipelining technique to achieve high-frequency resolution and high-speed throughput, whereas, a shifted clock technique is proposed to reduce the number of registers in the phase accumulator design. A PA with multiple pipeline stages increases the repeated registers, which leads to higher power consumption; therefore, a shifted clocking technique is used to reduce the number of registers while preserving the high-speed operation. Applying this method to the proposed design, the pre-skewing registers decreases by 58.3%. The proposed project has been coded Verilog hardware describe language (HDL), designed and simulated using ALTERA Quartus II software.

CONTENTS

List of Figures

List of Abbreviations

PA	Phase Accumulator
DFF	D flip flop
RCA	Ripple carry adder
CLA	Carry look-ahead adder
KS	Kogge-Stone adder
DDFS	Direct digital frequency synthesizer
RTL	Register transfer logic
FPGA	Field programmable gate array

CHAPTER ONE
INTRODUCTION

1.1 Overview

Phase accumulator (PA) is a generator that generates the digital phase of 0 to 2π range values. PA architecture consists of adder and register, as shown in Figure 1.1.

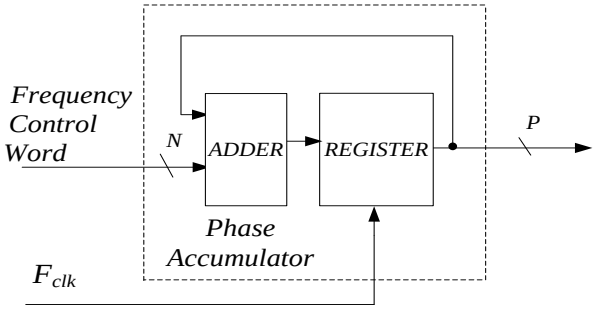


Figure 1.1 Phase accumulator architecture

The adder is the key element of the PA, whereas the register is the storage element of the data. The Frequency Control Word (FCW) input bit adds to the adder at every clock pulse and internally combines with sum bits feedback of the second register to perform an accumulation[1]. This operation demonstrates that the phase out (sawtooth shape) steadily increases with every clock pulse to the extreme limit (2^N -1) and then resets back to zero.

The PA output is in incremental form and represents the address of the ROM LUT according to the corresponding increment of the angle. The PA can be illustrated as a phase wheel in the DDFS system, as shown in Figure 1.2.

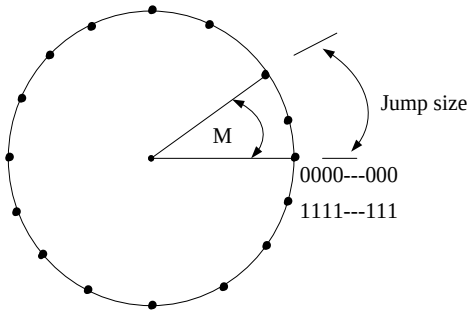


Figure 1.2 Digital Phase Wheel Source: [2]

The generated phase increments have a repetitive angular phase rotating around the phase wheel in the range of 0° to 360°[2]. One rotation of the vector around the phase wheel, at a constant speed, produces one complete cycle (0 π to 2 π) of the output sine wave.

1.2 Phase Accumulator Design Based on Pipelining Technique

PA is the key component that contributes to the speed performance of the DDFS that mainly comes from the pipelining technique, thereby improving the performance of PA speed. Pipelining is technically defined as a technique in which the given task is partitioned into a number of sub-tasks that should be sequentially performed [3-5]. Two types of pipelining technique are used in the PA design: multiple-pipelined PA with mono-bit stage register-adder and multiple-stage pipelined PA with group-bit stage of register-adder.

1.3 Phase Accumulator Application

The Phase accumulator application is use as a phase generator for direct digital frequency synthesizer (DDFS). The DDFS is a circuit system design to produce an analog sine waveform from a digital source. This system is

defined as a technique that uses the frequency digital word to generate digital phase waveform and then convert it to analog sine wave.

The phase accumulator may be viewed as a programmable digital of fractional divider that divides the reference sample frequency.

1.3 Objective

To achieve high-speed phase accumulator throughput, fast adders are used and combined with pipelining technique. However, the complexity of the adder circuit and the increase of the repetitive registers are the constraints in the implementation of the proposed PA design hence careful tradeoffs are needed.

To obtain high-frequency resolution [6], a wide frequency tuning word is required.

Thus, based on the aforementioned issues, this study proposes a design that employs a novel PA design based on pipelining technique to achieve high-frequency resolution and high-speed throughput output. In this project, a shifted clock technique has been used to reduce the number of the repetitive registers, which leads to reduce the power consumption.

1.4 The aim of the project

The aim of the proposed design is to:

. Design ripple carry adders and DFF registers

1. Use Verilog HDL Code tools
2. Use hierarchical design techniques

3. Model and simulate the 20-bit pipelined phase accumulator using Verilog HDL.

CHAPTER TWO
METHODOLOGY

2.1 Phase Accumulator Architecture

The phase accumulator is used to provide the phase output of the DDFS system. The resolution of the DDFS system as well as the speed of the DDFS depends on the PA design. A detailed explanation of the proposed PA architecture will be discussed in this section.

In order to achieve high Frequency resolution, it is preferable to design a PA with large FCW bits input. However, a large ROM size is required to implement all the 2^N bit of PA output. Due to this reason, part of the MSB phase output is used to address the phase to amplitude converter or (ROM Lookup Table) while maintaining high frequency resolution.

2.2 4-bit Ripple carry adder (RCA)

A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascaded, with the carry output from each full adder connected to the carry input of the next full adder in the chain. Figure 2.1 shows the interconnection of four full adder (FA) circuits to provide a 4-bit ripple carry adder [7]. Notice from Figure 2.1 that the input is from the right side because the first cell traditionally represents the least significant bit (LSB). The sum and the carry outputs are represented by the following equations:

$$S_{3:0} = x_{3:0} \oplus y_{3:0} \oplus C_{3:0} \quad (2\text{-}1)$$

$$Cout = (x_{3:0} \oplus y_{3:0}) . C_{3:0} + x_{3:0} . y_{3:0} \quad (2\text{-}2)$$

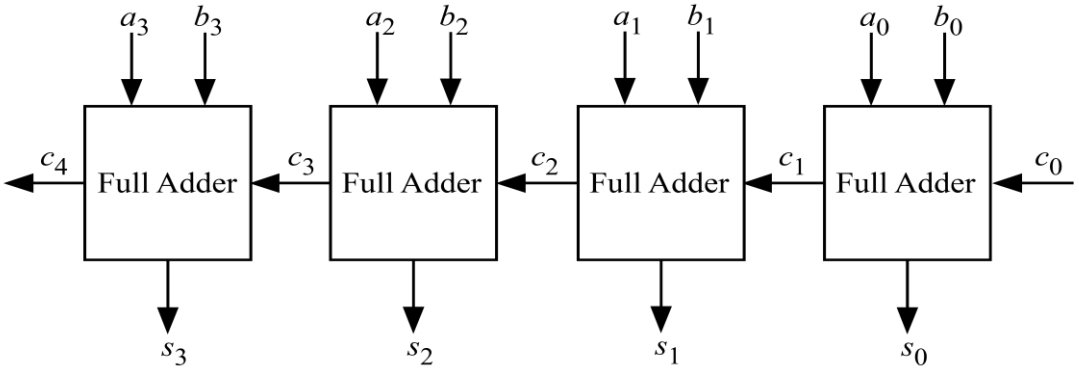


Figure 2.1 block diagram of 4-bit RCA

2.3 4-Bit D Flip Flop Register

Shift registers are a type of sequential logic circuit, mainly for storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop.

A basic 4-bit DFF register can be constructed using four D flip-flops, as shown in Figure 2.2.

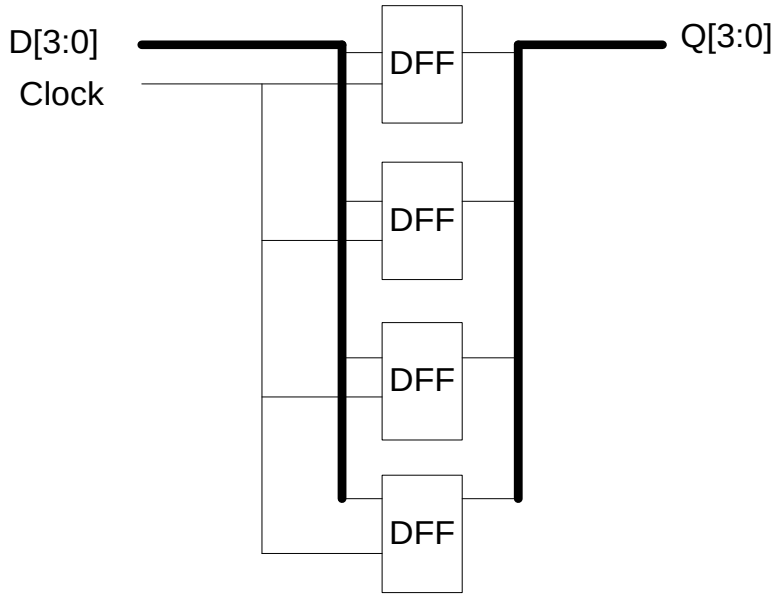


Figure 2.2 4-bit DFF register

2.4 Pipelining Technique

Pipelining technically is defined as a technique that the given task is partitioned into a number of sub-tasks that need to be performed in sequence [3]. Therefore, this technique is a best choice to increase the throughput of the PA phase output, and this throughput will doubles with the number of pipeline stages[8]. The block diagram of 20-bit pipelined phase accumulator is shown in Figure 2.3.

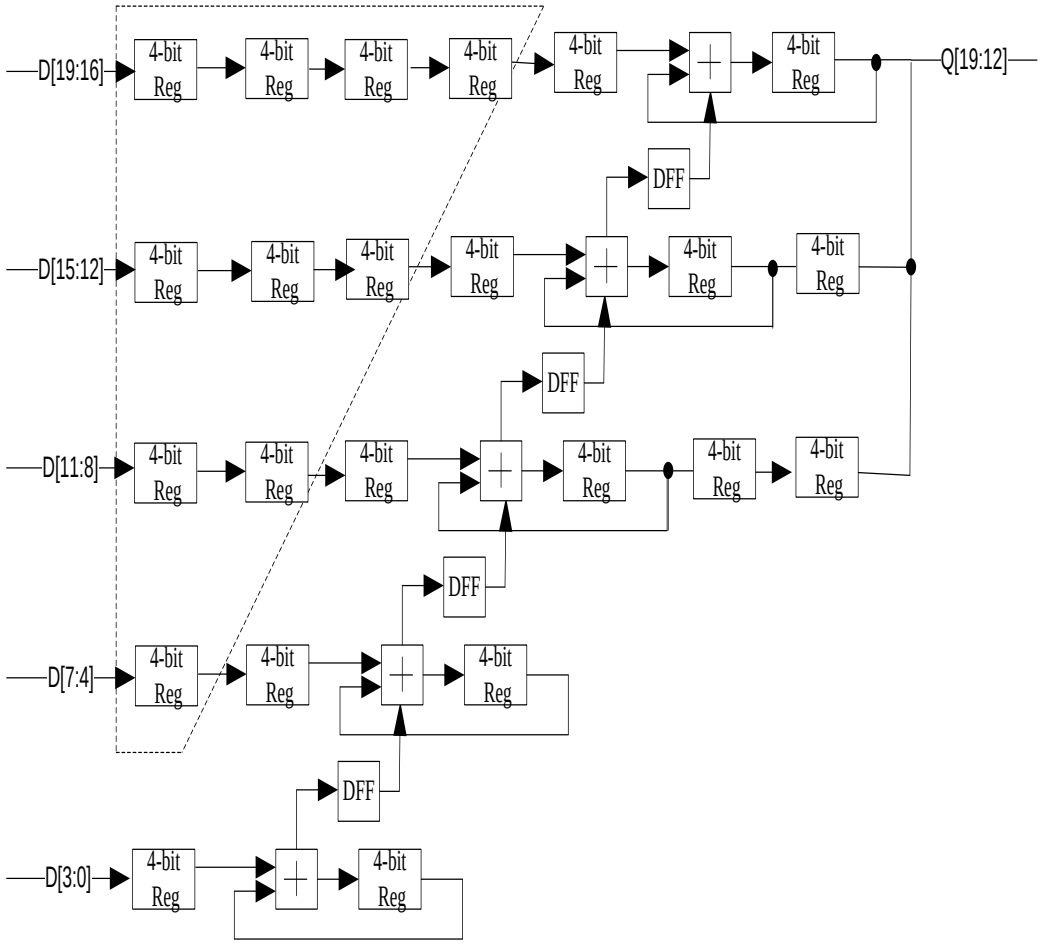


Figure 2.3 20-bit pipelined phase accumulator

The pipelined PA with mono-bit stage register-adder uses small-bit input (below 12-bit). For example the mono-bit pipelining stages are applied in [9] to obtain 9-bit pipelined PA.

2.5 Shifted Clock Technique

The number of registers increases with the number of pipeline stages, which leads to high power consumption. Shifted clock technique is able to reduce the number of pre-skewing registers while preserving high-speed operation[10]. In this technique, D flip-flops (DFFs) were used to connect each row of the pipeline stages with FCW input. The shifted clocking technique uses DFFs to connect each row of pipeline stages.These registers were clocked by the pipelined pulses with one clock cycle based on the shifted clock pulses[6, 10]. These registers are clocked by one clock cycle based on the shifted clock pulses as shown in figure 2.4.

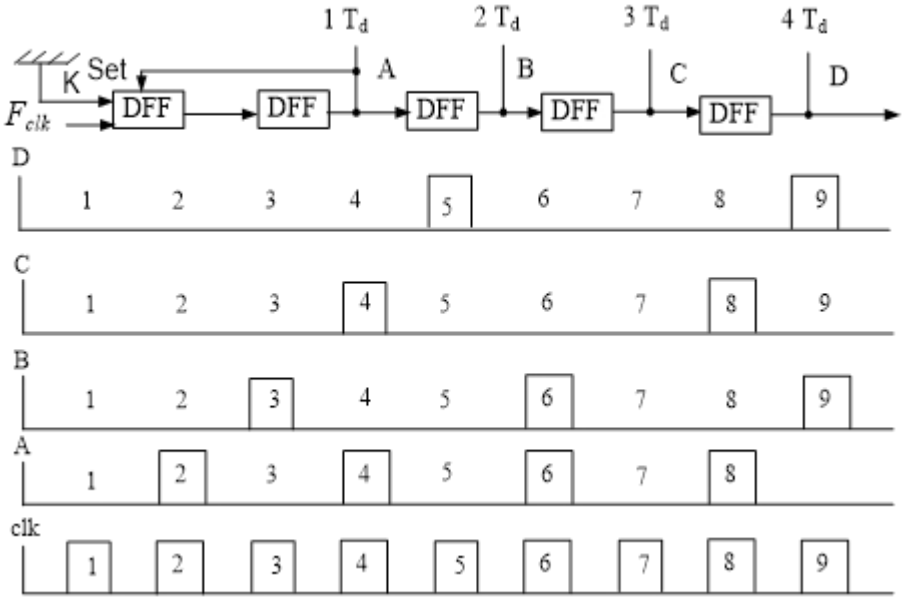


Figure 2.4 Shifted clock technique

The Shifted clocking technique is presented in Figure 2.5. It is applied to reduce the pre-skewing registers of the 4 × 5 pipelined PA design.

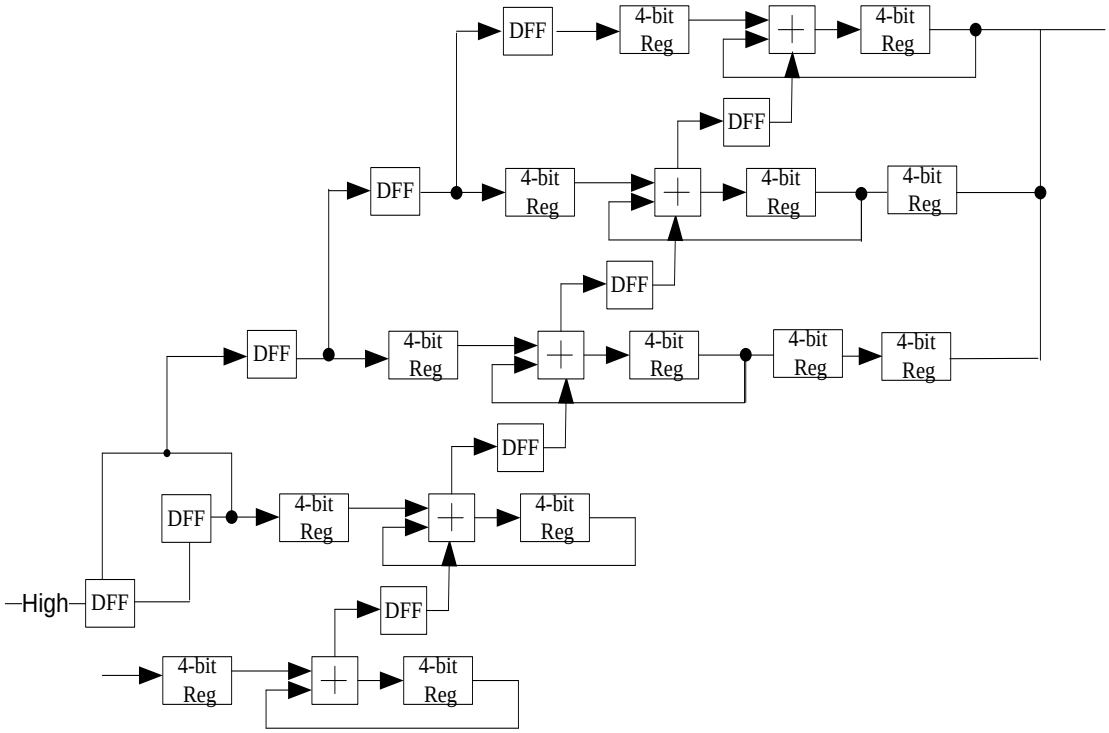


Figure 2.5 20-bit pipelined phase accumulator based on Shifted clock technique

Consider the phase accumulator input bits are N, the PA was partitioned into L stages with B DFFs in each stage. The number of the DFFs, *K* for pre-skewing registers ae given by (2-3).

$$K = (N * (L + 1))/2 \quad (2\text{-}3)$$

The conventional 20-bit pipelined PA use 60 DFFs in the pre-skewing part of the phase accumulator. By applying the shifted clock technique on the pipelined PA design, the number of DFFs reduced to 25 DFFs as is given by the equation (2-4):

$$K = N + L \quad (2\text{-}4)$$

Based on equation 3-2, the numbers of pre-skewing registers have been reduced significantly. Due that, the number of pre-skewing registers R decreases by 58.3% (Figure. 2.5).

CHAPTER THREE
RESULT AND DISCUSSION

3.1 Introduction

This chapter starts with the conventional MAT LAB simulation of phase accumulator. The simulation will be expanded to 20-bit. The simulation gives an idea about the functionality of PA operation and its output waveform shape. The 20-bit pipelined PA based RCA adder, was coded in Verilog HDL language.

The conventional ripple carry adder (RCA) has been used in the pipelining accumulator design for high speed operation.

The shifted clock technique was utilized in the proposed pipelined PAs, in order to reduce the number of pre-skewing registers. The proposed 20-bit PA designs were coded in Verilog HDL code, simulated and successfully verified in Qurtus II software.

The detail of simulation, verification and implementation of the 20-bit pipelined PA will be completed in this chapter.

3.2 Functional Verification of PA Module

A phase accumulator was modelled using the MAT LAB Simulink to allow the system parameters to be functionally verified and tested. The DDFS blocks such; PA, the ROM look-up table was architected using the MAT LAB Simulink library. The scopes monitor connects at the output of phase PA and look-up table. The DDFS system design is shown in Figure 3.1.

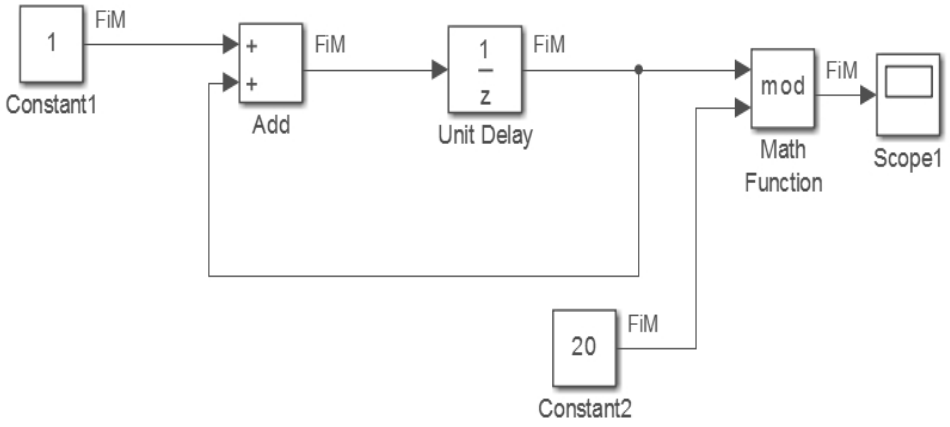


Figure 3.1 20-bit PA Architecture using MATLAB Simulink Library

The phase accumulator architect circuit builds as follow; constant one value component and the feedback of the adder output after one delay time, inputs the adder, together with the constant 2 feeds the module block to achieve 20-bit PA 32-values. The PA values determine by the constant 2 value umber.

The PA output simulation results shown in Figure 3.2 at the scope. The figure appeared that the PA sawtooth linearly increases based on the constant 2 values. Hence the constant 2 chosen to be 20, the sawtooth increase to the desired number, and reset back to zero.

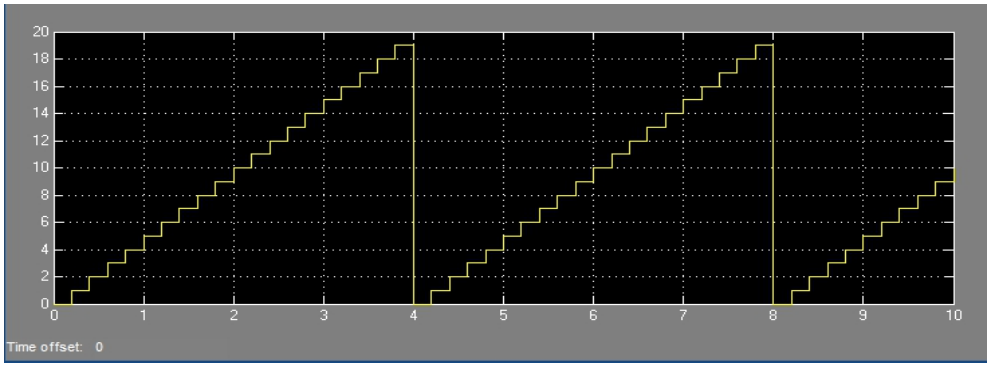


Figure 3.2 The phase accumulator output

3.3 Hardware Description Language (HDL)

Hardware description language (HDL) is a specialized computer language used to describe the structure and behavior of electronic circuits, and most commonly, digital logic circuits.

A hardware description language enables a precise, formal description of an electronic circuit that allows for the automated analysis and simulation of an electronic circuit.

Base on the explanation of the HDL, the proposed 20-bit PA design consists of 4-bit ripple carry adder, 4-bit DFF and single DFF . The Verilog HDL code of the 4-bit ripple carry adder, full adder, 4-bit DFF and single DFF are listed in the following:

3.4 4-Bit RCA Adder Verilog HDL Code

The 4-bit RCA adder has been codded Verilog HDL code. The 4-bit RCA adder and a full adder have been codded Verilog HDL code using QurtusII software as in following:

```
module rca_4
(
 input [3:0] x,
 input [3:0] y,
 input c_in,
 output [3:0]  sum,
 output c_out
 );
```

```
wire [3:1] carry;

f_adder_1 fa0(.x(x[1]),.y(y[1]),.c_in(c_in),
      .c_out(carry[1]),.sum(sum[0]));
f_adder_1 fa1(.x(x[1]),.y(y[1]),.c_in(carry[1]),
      .c_out(carry[2]),.sum(sum[1]));
f_adder_1 fa2(.x(x[2]),.y(y[2]),.c_in(carry[2]),
      .c_out(carry[3]),.sum(sum[2]));
f_adder_1 fa3(.x(x[3]),.y(y[3]),.c_in(carry[3]),
      .c_out(c_out),.sum(sum[3]));

// assign the final sum bit.
assign c_out = c_out;

endmodule
```

3.5 Verilog code of 4-bit D Flip Flop Register

The Verilog code of 4-bit Dff register is shown as in following:

```
module D_4
(D,clK,Reset,Q );
input [3:0]D;
input clK ,Reset;
output [3:0]Q;
```

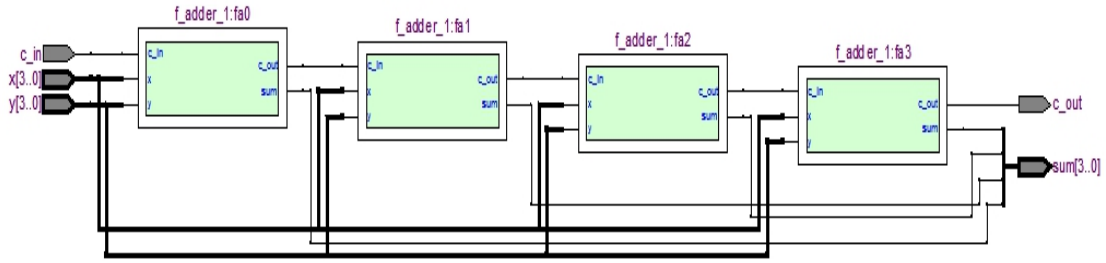
```
reg [3:0]Q;

always@( posedge clK or posedge Reset)
begin
if (Reset)
Q <= 1;
else
    Q <= D;
end
endmodule
```

The Verilog code of DFF register is shown as in following:

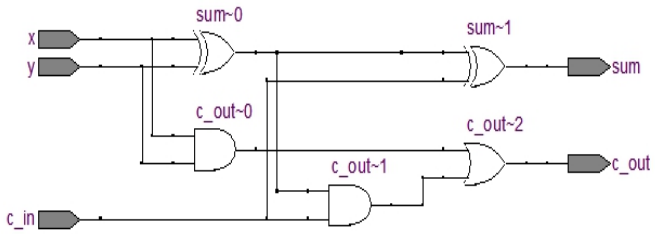
```
module  DFF_1
(DclK,Reset,Q );
input D ,clK ,Reset;
output Q;
reg Q;
always@(posedge clK or posedge Reset)
begin
if (Reset)
Q <= 1;

else
    Q <= D;
end
```

endmodule

The 4-bit ripple carry adder and the full adder code were elaborated and synthesized using ALTERA Qurtus II software. The register transfer logic (RTL) viewer of 4-bit ripple carry adder and the full adder are shown in Figure 3.3.



(a)



(b)

Figure 3.3 The RTL viewer of 4-bit ripple carry adder (a) and 1-bit full adder (b).

The register transfer logic (RTL) viewer of the pipelined phase accumulator base on ripple carry adder design in Figure 3.5 shows the four 4-bit RCA adder consisted of a four 1-bit full adder.

3.6 20-bit Pipelined Phase accumulator

A 20-bit pipelined PA based on RCA adder was used to increase the speed, whereas applying the shifted clocking technique on the proposed pipelined PA, reduce the numbers of pre-skewing registers by 59.1% (from 60 to 25). The proposed 20-bit PA design was coded in Verilog HDL code, simulated and successfully verified using ALTERA Qurtus II software.

The Verilog HDL code of the proposed 20-bit pipelined PA project shown in the following :

```
module PA_20(
Clk,
Reset,
D,
Q
);


Clk;  input
Reset;  input
[19:0] D;  input
[19:8] Q;output

[19:8] Q;   wire
internal_wire_0;   wire
[3:0] internal_wire_1;   wire
```

[3:0] internal_wire_2; wire

[3:0] internal_wire_3; wire

internal_wire_4; wire

internal_wire_35; wire

internal_wire_6; wire

[3:0] internal_wire_7; wire

[3:0] internal_wire_8; wire

[3:0] internal_wire_9; wire

internal_wire_10; wire

internal_wire_36; wire

internal_wire_14; wire

[3:0] internal_wire_15; wire

[3:0] internal_wire_37; wire

[3:0] internal_wire_17; wire

internal_wire_38; wire

internal_wire_19; wire

[3:0] internal_wire_20; wire

[3:0] internal_wire_39; wire

[3:0] internal_wire_22; wire

[3:0] internal_wire_24; wire

internal_wire_26; wire

internal_wire_29; wire

[3:0] internal_wire_30; wire

internal_wire_31; wire

[3:0] internal_wire_32; wire

```
internal_wire_33;   wire
internal_wire_34;   wire

internal_wire_10 = 1;assign


b2v_inst2(          DFF_1
.D(internal_wire_0),
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_6));


b2v_inst21(    D_4
.clK(Clk),
.Reset(Reset),
.D(D[3:0]),
.Q(internal_wire_1));


b2v_inst22(  rca_4

.x(internal_wire_1),
.y(internal_wire_2),
.c_out(internal_wire_0),
```

```
.sum(internal_wire_3));

b2v_inst23(    D_4
.clK(Clk),
.Reset(Reset),
.D(internal_wire_3),
.Q(internal_wire_2));

b2v_inst26(         DFF_1
.D(internal_wire_4),
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_35));

b2v_inst27(    D_4
.clK(internal_wire_35),
.Reset(Reset),
.D(D[7:4]),
.Q(internal_wire_7));
```

```
b2v_inst28( rca_4
.c_in(internal_wire_6),
.x(internal_wire_7),
.y(internal_wire_8),
.c_out(internal_wire_26),
.sum(internal_wire_9));

b2v_inst29(   D_4
.clK(Clk),
.Reset(Reset),
.D(internal_wire_9),
.Q(internal_wire_8));

b2v_inst3(        Trigger
.D(internal_wire_10),
.clK(Clk),
.Reset(Reset),
.set(internal_wire_35),
.Q(internal_wire_4));

b2v_inst30(        DFF_1
.D(internal_wire_35),
```

```
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_36));


b2v_inst31(    D_4
.clK(internal_wire_36),
.Reset(Reset),
.D(D[11:8]),
.Q(internal_wire_15));


b2v_inst32(  rca_4
.c_in(internal_wire_14),
.x(internal_wire_15),
.y(internal_wire_37),
.c_out(internal_wireE_33),
.sum(internal_wire_17));


b2v_inst33(    D_4
.clK(Clk),
.Reset(Reset),
.D(internal_wire_17),
.Q(internal_wire_37));
```

```
b2v_inst34(    D_4
.clK(internal_wire_38),
.Reset(Reset),
.D(D[15:12]),
.Q(internal_wire_20));


b2v_inst35(  rca_4
.c_in(internal_wire_19),
.x(internal_wire_20),
.y(internal_wire_39),
.c_out(internal_wire_34),
.sum(internal_wire_22));


b2v_inst36(    D_4
.clK(Clk),
.Reset(Reset),
.D(internal_wire_22),
.Q(internal_wire_39));


b2v_inst37(         DFF_1
```

```
.D(internal_wire_36),
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_38));

b2v_inst38(    D_4
.clK(Clk),
.Reset(Reset),
.D(internal_wire_24),
.Q(Q[11:8]));

b2v_inst39(    D_4
.clK(Clk),
.Reset(Reset),
.D(internal_wire_37),
.Q(internal_wire_24));

b2v_inst4(         DFF_1
.D(internal_wire_26),
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_14));
```

```
b2v_inst40(    D_4
.clK(Clk),
.Reset(Reset),
.D(internal_wire_39),
.Q(Q[15:12]));


b2v_inst41(         DFF_1
.D(internal_wire_38),
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_29));


b2v_inst42(    D_4
.clK(internal_wire_29),
.Reset(Reset),
.D(D[19:16]),
.Q(internal_wire_32));

b2v_inst43(    D_4
.clK(Clk),
.Reset(Reset),
```

```
.D(internal_wire_30),
.Q(Q[19:16]));

b2v_inst44( rca_4
.c_in(internal_wireRE_31),
.x(internal_wire_32),
.y(Q[19:16]),
.sum(internal_wire_30));

b2v_inst5(        DFF_1
.D(internal_wire_33),
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_19));

b2v_inst6(        DFF_1
.D(internal_wire_34),
.clK(Clk),
.Reset(Reset),
.Q(internal_wire_31));

Q = Q;assign

endmodule
```

The 20-bit PA code were elaborated and synthesized using ALTERA Qurtus II software, The register transfer logic (RTL) viewer of the proposed design is shown in the Figure 3.4

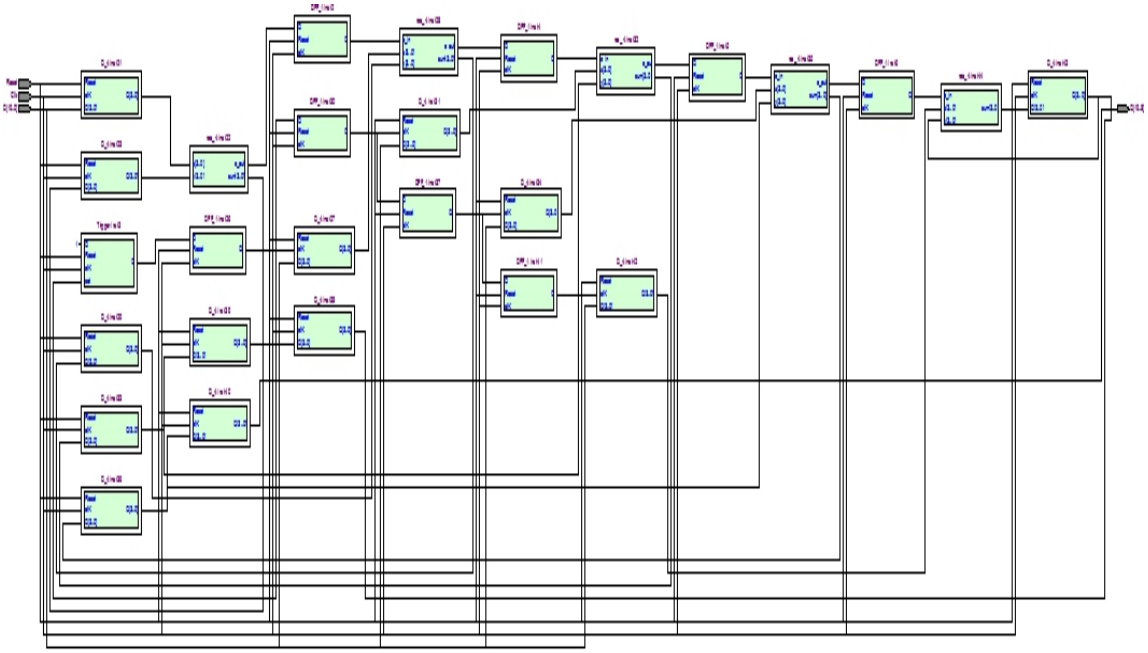


Figure 3.4 RTL viewers of the 20-bit pipelined phase accumulator based on RCA adder.

3.7 20-Bit Pipelined PA Simulation Based On Shifted Clock Technique

The proposed 20-bit PA design was synthesized, elaborated and compiled in ALTERA Quartus II software system to functionally verify the system design. The gate level simulation of the proposed design in Figure 3.5 shows the PA output.

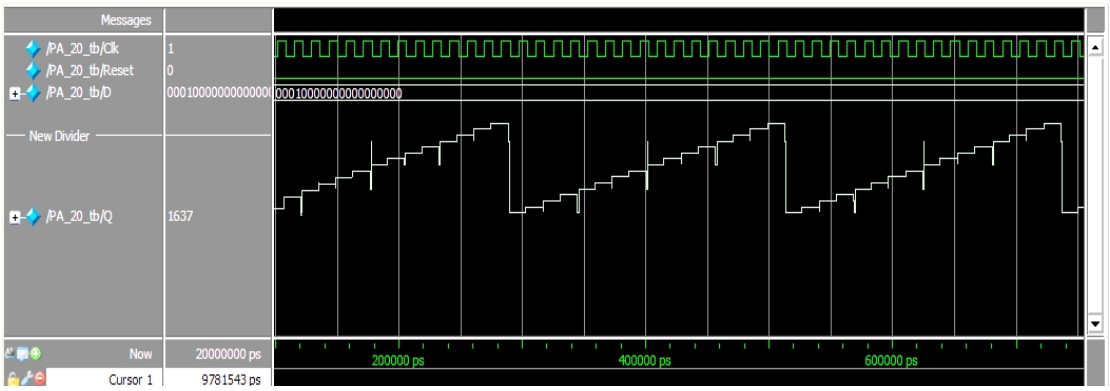


Figure 3.5 The gate level simulation of the pipelined PA show Phase Accumulator

This gate level simulation result in figure demonstrates that the phase out (sawtooth shape) steadily increases with every clock pulse to the extreme limit 2^N -1 (for the proposed design, the limit is (2^{19} -1), and then resets back to zero.

CHAPTER FOUR
CONCLUSION AND FUTURE WORKS

4.1 Conclusion

This project proposed the architectures of high speed pipelined phase accumulator. The project has been designed using HDL and then simulated on Modelsim and Quartus II software.

The shifted clocking technique uses DFFs to connect each row of pipeline stages The Shifted clocking technique is applied to reduce the pre-skewing registers of the 4 × 5 pipelined PA design.

The conventional 20-bit pipelined PA use a 60 DFFs in the pre-skewing part of the phase accumulator. By applying the shifted clock technique on the pipelined PA design, the number of DFFs reduced to 25 DFFs. The numbers of pre-skewing registers have been significantly decreases by 58.3% ratio.

A 20-bit PA had been designed, and required 56 logic elements, 52 registers to operate 597.73 MHz frequency. The 20-bit PA was coded in Verilog HDL and successfully simulated in ModelSim with ALTERA Quartus II software of the proposed design.

4.1 Future works

It is suggested to implement the proposed pipelined phase accumulator design on the field programmable gate array (FPGA), to compare the real time result with the achieved gate level simulation result, to prove the validity of the design.

REFERENCES

[1] S. H. Akhurwy, "AN AREA EFFICIENT READ ONLY MEMORY (ROM) DESIGN ARCHITECTURE FOR HIGH SPEED DIRECT DIGITAL FREQUENCY SYNTHESIZER," PhD, Department of Electrical, Electronics & System Engineering, UNIVERSITI KEBANGSAAN MALAYSIA, Malaysia, 2015.

[2] D .B. Ken Gentile, Ted Harris (1999). *A Technical Tutorial on Digital Signal Synthesis* , http://www.analog.com/UploadedFiles/Tutorials/450968421DDS_Tutorial_rev12-2-99.pdf

[3] M. A.-E.-B. H. El-Rewini, <*Pipelining Design Techniques.pdf*>. Hoboken, New Jersey & canada: A JOHN WILEY & SONS, INC PUBLICATION 2005.

[4] S. H. Ibrahim, S. H. M. Ali, and M. S. Islam, "12-BIT HIGH SPEED DIRECT DIGITAL FREQUENCY SYNTHESIZER BASED ON PIPELINING PHASE ACCUMULATOR DESIGN," *Journal of Asian Scientific Research*, vol ,2 .pp. 667-672, 2012.

[5] Y. S. Kim, j. Lee, Y. Hong, J. E. Kim, and K. H. Baek, "Low-power pipelined phase accumulator with sequential clock gating for DDFSs," *Electronics Letters*, vol. 49, pp. 1445-1446, 2013.

[6] Y. Ching, W. Jun, and C. Hsuan, "A 5.3-GHz 32-bit accumulator designed for direct digital frequency synthesizer. ," *Chin Sci Bull,*, 2012.

[7] U. o. California. (2009). *Adder* .

[8] B. S. Jensen, M. M. Khafaji, T. K. Johansen, V. Krozer, and J. C. Scheytt, "Twelve-bit 20-GHz reduced size pipeline accumulator in 0.25 µm SiGe:C technology for direct digital synthesiser applications," *Circuits, Devices & Systems, IET*, vol. 6, pp. 19-27, 2012.

[9] G. Xueyang, F. F. Dai, J. D. Irwin, and R. C. Jaeger, "An 11-Bit 8.6 GHz Direct Digital Synthesizer MMIC With 10-Bit Segmented Sine-

Weighted DAC," *Solid-State Circuits, IEEE Journal of*, vol. 45, pp. 300-313, 2010.

[10] S. H. Ibrahim, "Hardware Implementation of 32-Bit High-Speed Direct Digital Frequency Synthesizer," *The Scientific World Journal*, vol. Volume 2014, p. 10, 2014.