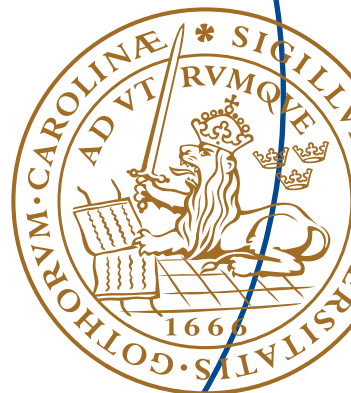


Master's Thesis

Hardware Implementation of a 32-point Radix-2 FFT Architecture

Ying Gao



Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, July 2015.



LUND UNIVERSITY

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY

MASTER OF SCIENCE THESIS

Hardware Implementation of a 32-point Radix-2 FFT Architecture

Author:
Ying Gao

Supervisor:
Rakesh Gangarajaiah
Erik Hertz
Peter Nilsson

Lund July 28, 2015

The Department of Electrical and Information Technology
Lund University
Box 118, S-221 00 LUND
SWEDEN

This thesis is set in Computer Modern 10pt,
with the L^AT_EX Documentation System

© Ying Gao 2015

Abstract

The *Fast Fourier Transform* (FFT) algorithm has been widely used in the Digital Signal Processing industry as a rudimentary operation to select the specific frequency components of a signal, which has been involved with other time domain signals. In order to fulfill the requirements of executing precise calculations and less power & area consumption, an algorithm with less number of adders and multipliers is used.

In this thesis, a radix-2 32-point FFT algorithm, which is using *Decimation-In-Frequency* (DIF) , is implemented in VHDL. In addition, the implementation is designed for a 65nm CMOS process. The ASIC verification process is tested and implemented, by using Synthesis, Post-synthesis Simulation, Place & Route, Post-layout Simulation, and Prime Time. Results regarding area, throughput, and power consumption are presented.

Acknowledgements

Firstly, I would like to thank to my supervisor, Peter Nilsson, who has offered great help in the fully understanding of the 32-point FFT algorithms. Furthermore, I would like to thank Erik Hertz, who has given his opinions regarding with the building up of the 32-point FFT architecture. Then, I would like to thank to my supervisor, Rakesh Gangarajaiah who always patiently provided the detailed technical supports, concerning with the digital implementation. Finally, my appreciation would extend to my friends and family members who always on my side from the very beginning , and my System-on-Chip classmates for all the generous help.

Ying Gao
Lund, June 2015

Popular Scientific Essay

This thesis report is about the hardware implementation of 32-point Radix-2 FFT Architecture.

The modeling part is done in MATLAB. The algorithm for the realization of the FFT in MATLAB modeling part is using the Cooley - Tukey algorithm, which is famous for the radix-2 butterfly. By using the radix-2, the computation complexity is decreasing with fewer numbers of adders and multipliers. Therefore, The area consumption could be saved.

The programming of the design is using VHDL. Since the DIF algorithm is being chosen for this thesis, the input signals are in the positive sequence order and the output signals are half even-indexed, a half odd-indexed time samples. In the hardware programming part, except for the five stages of radix-2 butterflies, an additional block is added to the end of the architecture, which is called-"bit reverse", in order to reverse the order of the output signals into sequence order. Therefore, the results of the 32-point Radix-2 FFT could be used in the next phase.

The hardware utilization is been analysis in the Xilinx ISE Design Suite 14.2 by choosing the FPGA board-Xilinx Vertex 5 (XC5VLX110T).

The ASIC verification is done in a $65nm$ CMOS Technology, including Synthesis in Design Vision, Post-synthesis in Modelsim, Place & Route in Encounter, Post-layout Simulation in Modelsim, as well as Prime Time. All the measurements all focus on the trade-off between speed, and area consumption. This thesis design is aiming for low-area with proper performance.

Contents

| | |
|-----------------------------------|-------------|
| Abstract | iii |
| Acknowledgements | iv |
| Popular | vii |
| List of Tables | xi |
| List of Figures | xiii |
| List of Acronyms | xv |
| 1 Introduction | 1 |
| 1.1 Thesis Overview | 1 |
| 1.2 Thesis Organization | 2 |
| 2 Algorithms | 3 |
| 2.1 The DFT Algorithm | 3 |
| 2.2 The FFT Algorithm | 4 |
| 2.2.1 The 4-point FFT | 5 |
| 2.2.2 The 32-point FFT | 8 |

| | | |
|----------|-------------------------------------|-----------|
| 3 | Matlab Modeling | 11 |
| 4 | Hardware Implementation | 17 |
| 5 | ASIC Verification | 23 |
| 5.1 | Synthesis | 24 |
| 5.2 | Place & Route | 26 |
| 5.3 | Prime Time | 27 |
| 6 | Conclusion & Future Work | 31 |
| 6.1 | Conclusion | 31 |
| 6.2 | Future Work | 32 |
| A | Appendix 1 | 33 |
| | Reference | 33 |

List of Tables

| | | |
|-----|--|----|
| 4.1 | Device utilization summary | 19 |
| 5.1 | Timing and maximum speed constraint and 1.20V supply voltage | 25 |
| 5.2 | Timing and minimum area constraint and 1.20V supply voltage | 25 |
| 5.3 | Area at maximum speed constraint and 1.20V supply voltage | 25 |
| 5.4 | Area at minimum area constraint and 1.20V supply voltage . | 25 |
| 5.5 | Prime time power report I | 28 |
| 5.6 | Prime time power report II | 28 |
| 5.7 | Power consumption at 10MHz and 1.20V supply voltage with maximum speed constraint | 28 |
| 5.8 | Power consumption at maximum frequency and 1.20V supply voltage with maximum speed constraint | 29 |
| 5.9 | Slacks for setup time and hold time | 30 |

List of Figures

| | | |
|-----|---|----|
| 2.1 | The positions of twiddle factors in (a) radix-2 DIF butterfly and (b) radix-2 DIT butterfly | 5 |
| 2.2 | A flow graph of the complex valued 4-point radix-2 DIF FFT Algorithm | 6 |
| 2.3 | The real part of 4-point radix-2 DIF FFT architecture | 7 |
| 2.4 | The imaginary part of 4-point radix-2 DIF FFT architecture | 8 |
| 2.5 | The five stages of the 32-point DIF FFT architecture | 9 |
| 2.6 | The detailed-first stage architecture | 10 |
| 3.1 | Two input signals for 32-point DIF FFT | 12 |
| 3.2 | Two 32-point DIF FFT modeling results in full precision . . . | 13 |
| 3.3 | Comparison between truncated and un-truncated results . . . | 15 |
| 3.4 | Number of bits | 16 |
| 4.1 | The schematic top level of the 32-point FFT hardware architecture | 18 |
| 4.2 | The detailed schematic of the 32-point FFT hardware architecture | 19 |
| 4.3 | Two 32-point DIF FFT hardware implementation input signals | 20 |
| 4.4 | Two 32-point DIF FFT hardware implementation results . . | 21 |

| | | |
|-----|---|----|
| 5.1 | The ASIC verification flow | 23 |
| 5.2 | The synthesis data flow | 24 |
| 5.3 | 32-point DIF FFT chip design outlook | 26 |
| 5.4 | The prime time flow | 27 |
| 5.5 | Power consumption varies with frequency | 30 |

List of Acronyms

| | |
|----------------|--|
| FFT | Fast Fourier Transform |
| DFT | Discrete Fourier Transform |
| DIF | Decimation In Frequency |
| DIT | Decimation In Time |
| ABS | ABSolute value |
| ASIC | Application Specific Integrated Circuit |
| P&R | Place-and-Route |
| RTL | Register Transfer Level |
| SDF | Standard Delay Format |
| VHDL | Very High Speed Integrated Circuit Hardware Description Language |
| WNS | Worst Negative Slack |
| TNS | Total Negative Slack |
| SDF | Synopsys Delay Format |

FPGA Field-programmable Gate Arrays

CHAPTER 1

Introduction

1.1 Thesis Overview

This thesis report illustrates the hardware implementation of 32-point *Decimation In Frequency Fast Fourier Transform* and the ASIC verification flow. The later part has been emphasized in the paper.

The radix-2 butterfly algorithm is being used for the realization of the 32-point DIF FFT. The algorithm is tested in the Matlab modeling part first. The input signals for the design have already been settled to 16 bits and the rest of the signals in the design are fixed-point format. The programming part is done by using VHDL. The Xilinx ISE Design Suite 14.2 is being used in the analysis of the design utilization part. The ASIC flow, including Synthesis, Place & Route, Prime Time are being measured in the last part of the thesis.

1.2 Thesis Organization

This thesis report consists of seven chapters.

Chapter 1 focuses on the brief introduction of the main idea of the 32-point DIF FFT algorithm and hardware implementation methods. In addition, the organization of thesis report is introduced.

Chapter 2 gives detailed information about the DFT and FFT algorithms, the specific structure that has been used in this thesis, like the radix-2 butterfly structure. The five-stage hardware structure of the realization of the 32-point DIF FFT.

Chapter 3 illustrates the MATLAB modeling of the 32-point DIF FFT. Including the specific setting for the input signal, the error analysis of the result of the modeling, and the truncation method in the structure is included, as well.

Chapter 4 demonstrates the results of the VHDL programming of the 32-point DIF FFT structure. The simulation result will be compared to the Matlab modeling result.

Chapter 5 lists the ASIC Verification process of the hardware-mapped 32-point DIF FFT, including the *Synthesis*, *Post – synthesisSimulation*, *Place&Route*, *Post – layoutSimulation* and *PrimeTime*. The tables regarding the area, throughput, power consumption are presented.

Chapter 6 declares the conclusion of the whole process of the Thesis Project, as well as the future work, which could be refined in the future.

The reference of the thesis report is added in the end of the report.

CHAPTER 2

Algorithms

Chapter 2 is mainly about the algorithm of the DFT and FFT, along with the explanations in the form of figures and equations, which will correspond to the Chapter 3, the Matlab modeling part. In addition, the detailed information about the 4-point DIF FFT and 32-point DIF FFT architecture is introduced.

2.1 The DFT Algorithm

The *DFT* is short for *Discrete Fourier Transform*, which is one of the most crucial algorithms that has been used in digital signal processing and image processing industries.

The DFT algorithm is defined in equation (2.1), where n is an element belong to a matrix row; k represents row, which equals to 0 to $N - 1$ [2].

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{kn} \\ W_N^{kn} &= e^{\frac{-j2\pi kn}{N}} \end{aligned} \tag{2.1}$$

The magnitude and phase of the DFT algorithm are described in equation (2.2) as below:

$$\begin{aligned} \text{Mag}(X(k)) &= \sqrt{X_{Re}(k)^2 + X_{Im}(k)^2} \\ \varphi(X(k)) &= \arctan \frac{X_{Im}(k)}{X_{Re}(k)} \end{aligned} \quad (2.2)$$

In order to reduce the computation complexity of DFT algorithm, some changes have been added to the algorithm, in terms of convenience and efficiency.

2.2 The FFT Algorithm

As a fast computation algorithm, compared to DFT, the Fast Fourier Transform(FFT) Cooley - Tukey algorithm[3] is famous for decomposing the DFT computing module into small calculation blocks, which is called radix-2. By using that, the arithmetical complexity will be decreased from $O(N^2)$ to $O(N \log_2 N)$, which will increase the computation speed and the total computational cost will be greatly reduced.

Before using the radix-2 algorithm, the hardware realization of the 32-point FFT is parallel-in parallel-out, which is inappropriate for the implementation, because of the large amount of the usage of the adders and the multipliers. The number of the input ports for the whole architecture is more than 32 ports, i.e. for the chip manufacturing, the pins of the chip would increase at the same time.

Typically, the multiplication coefficients in Cooley - Tukey algorithms are called *twiddle factors*. For the FFT algorithm that has been used in this thesis project A radix-2 DIF butterfly configuration is used. Another configuration corresponding to the previous one is radix-2 DIT butterfly, the difference between these two algorithms is the location of the twiddle factors. Figure 2.1 illustrates the different position of the twiddle factors respectively.

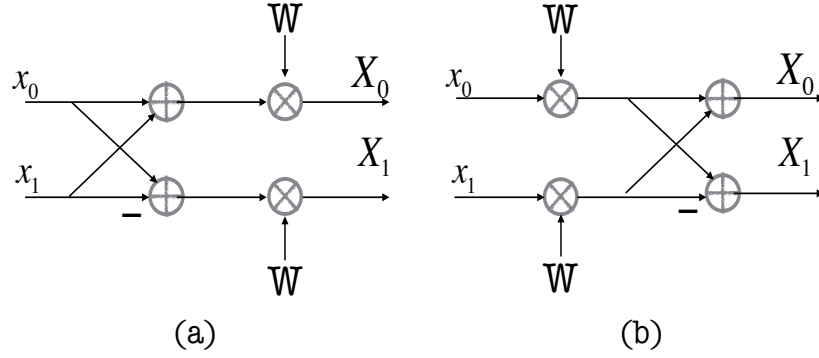


Fig. 2.1: The positions of twiddle factors in (a) radix-2 DIF butterfly and (b) radix-2 DIT butterfly

2.2.1 The 4-point FFT

The definition of the 4 - point FFT is shown in equation (2.3). In this equation, the input signal has been divided into two parts, the real part and the imaginary part[2]. During the hardware implementation, the imaginary part and the real part would all be expressed in the binary format.

$$\begin{aligned}
 X_{Re}(k) &= \sum_{n=0}^3 \left[X_{Re}(n) \cos \frac{2\pi kn}{4} + j X_{Im}(n) \sin \frac{2\pi kn}{4} \right] \\
 X_{Im}(k) &= - \sum_{n=0}^3 \left[X_{Re}(n) \sin \frac{2\pi kn}{4} - j X_{Im}(n) \cos \frac{2\pi kn}{4} \right]
 \end{aligned} \tag{2.3}$$

Figure 2.2 demonstrates the 4 - point DIF FFT data-flow graph. The input signals are in sequence order, on the contrary, the output signals are bit-reversed. During the procedure of implementation in hardware, the output signals are reorganized to bit-sequence order.

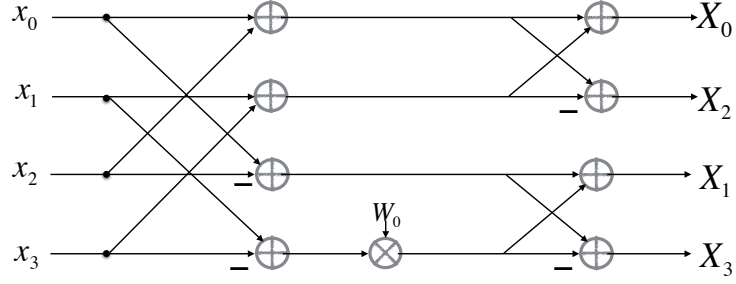


Fig. 2.2: A flow graph of the complex valued 4-point radix-2 DIF FFT Algorithm

The output values of X_0, X_1, X_2, X_3 , which can be seen in the figure 2.2 are being calculated in the equation (2.4).

$$\begin{aligned}
 X_0(k) &= x_0 + x_2 + x_1 + x_3; \\
 X_1(k) &= x_0 - x_2 + j(x_1 - x_3); \\
 X_2(k) &= x_0 + x_2 - x_1 - x_3; \\
 X_3(k) &= x_0 - x_2 - j(x_1 - x_3);
 \end{aligned} \tag{2.4}$$

Figure 2.3 illustrates the real part of the complex-valued 4-point FFT architecture. The output values of X_0, X_1, X_2, X_3 are listed in the equation (2.5).

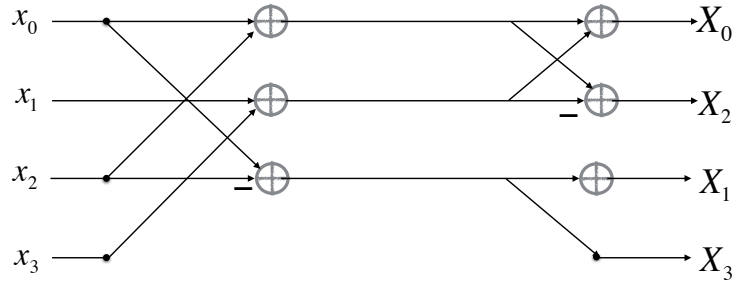


Fig. 2.3: The real part of 4-point radix-2 DIF FFT architecture

$$\begin{aligned}
 X_0(k) &= x_0 + x_2 + x_1 + x_3 \\
 X_1(k) &= x_0 - x_2 \\
 X_2(k) &= x_0 + x_2 - x_1 - x_3 \\
 X_3(k) &= x_0 - x_2
 \end{aligned} \tag{2.5}$$

Figure 2.4 illustrates the imaginary part of the complex-valued 4-point FFT architecture. The output values of X_0, X_1, X_2, X_3 are listed in the equation (2.6).

$$\begin{aligned}
 X_0(k) &= 0 \\
 X_1(k) &= j(x_1 - x_3) \\
 X_2(k) &= 0 \\
 X_3(k) &= -j(x_1 - x_3)
 \end{aligned} \tag{2.6}$$

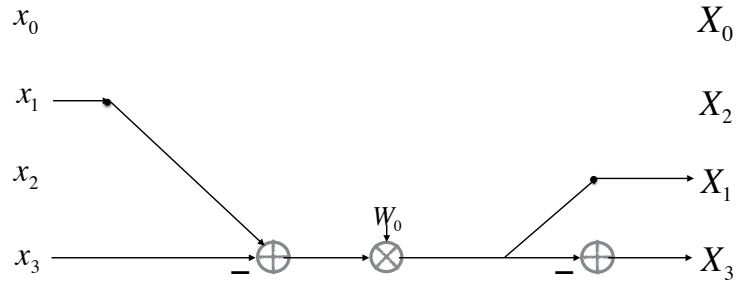


Fig. 2.4: The imaginary part of 4-point radix-2 DIF FFT architecture

2.2.2 The 32-point FFT

Figure 2.5 demonstrates the 32-point DIF FFT hardware architecture, which include five stages. In each stage, it contains a radix-2 butterfly architecture, several registers, whose number depending on the numbers of the input signals. Between every second stage, the twiddle factors are multiplexed to the output, which come from the previous stage.

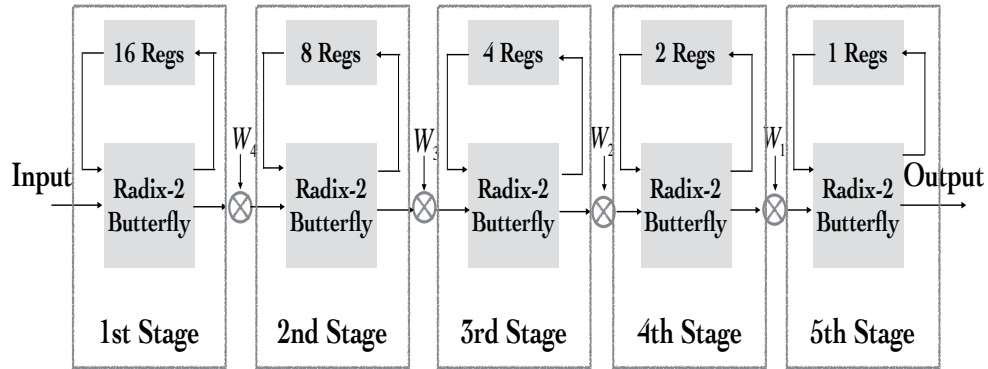


Fig. 2.5: The five stages of the 32-point DIF FFT architecture

Figure 2.6 describes the 1st stage of the Figure 2.5. Take the 1st stage as an example, it processes two phases of the calculations. In the first phase, the initial 16 input signals, which is the first half of the input signals, will be transferred and stored in the register 0 to register 15. In the second phase, the rest 16 input signals will be transferred into the first stage. The 1st input signal will be executed with the 17th input signal in the radix-2 butterfly, using either an addition or a subtraction. Next, the subtraction result, will be multiplied with the twiddle factor. The addition results will be stored, waiting for to be subtracted in the next stage.

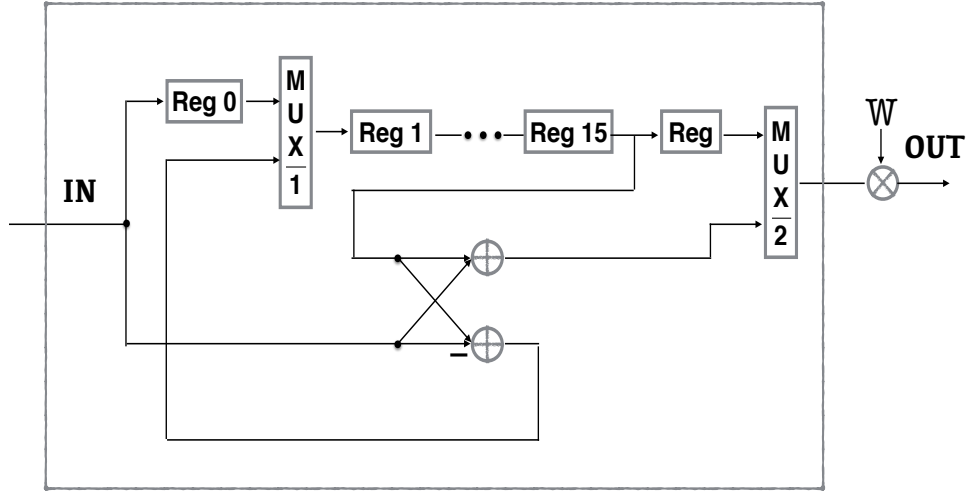


Fig. 2.6: The detailed-first stage architecture

By using this radix-2 butterfly architecture, the number of adders and multipliers will be reduced efficiently. For example, without using the radix-2 algorithm, the number of execution elements for the first stage would be 32 adders and 16 multipliers. By using the radix-2 method, the consumption number of the adders is 2, for the multiplier is one. As for the number of the registers, it is the same, which is 16 registers, to store the first 16 input signals. Since the area consumption are mainly depending on the number of the adders and the multipliers, area could be saved by using the radix-2 algorithm.

CHAPTER 3

Matlab Modeling

The chapter 3 illustrates the the modeling of the 32-point DIF FFT. The modeling is based on the hardware architecture that is shown in chapter 2 Figure 2.5.

First of all, the input signal should be processed in a specific form. The Figure 3.1 depicts the input signal that has been set to be four periods per frame. The 32 sampling points are shown with the dots.

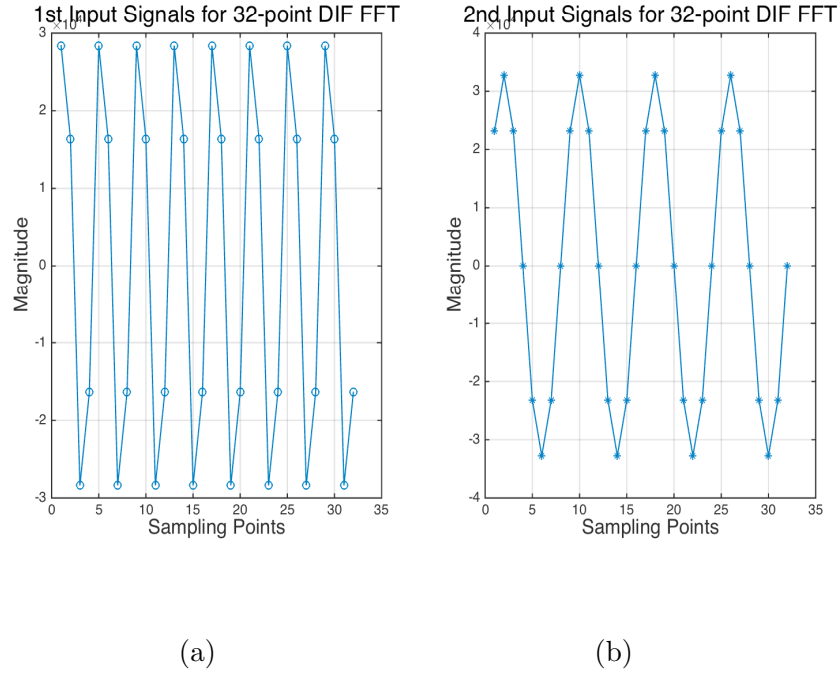


Fig. 3.1: Two input signals for 32-point DIF FFT

As it can be seen in the Figure 3.1, the 32 sampling points are divided evenly regarding with the X-axis. In the Figure 3.1 (a) shows the input signal with 8 periods per frame. Figure (b) contains 4 periods per frame. The reason for the specific setting for input signals is that little peaks on the waveform of the output performance could be avoided. Since these little peaks could make a bad influence on both the system truncation results and the whole performance of the modeling.

Figure 3.2 illustrates the two performances of the 32-point DIF FFT modeling which corresponding to the Figure 3.1 in full precision. Both Figure (a) and (b) are folded graphs.

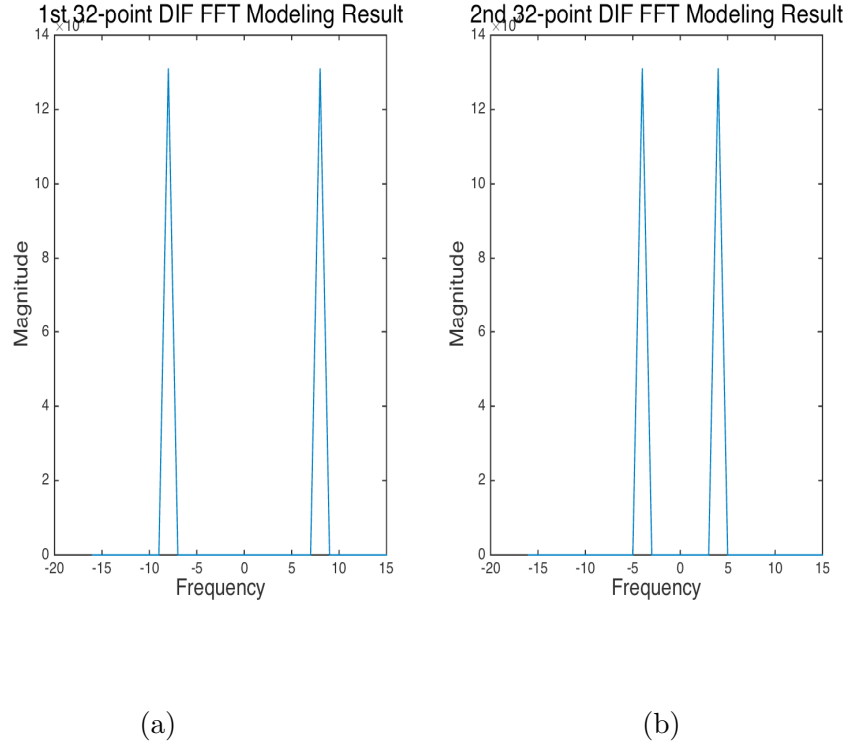


Fig. 3.2: Two 32-point DIF FFT modeling results in full precision

In hardware realization part, the truncated word lengths in the architecture are crucial. For the reason that the selection of the word length would have an impact on the area consumption, as well as the precision performance. If the word lengths of the inner signals of the architecture have been increased, the performance would indeed be better than before, the area is larger correspondingly. The excellent performance means that the result of the modeling is more precise, the quantization error is lower, which is good for the modeling. On the contrary, the bigger area, the more money could be consumed to manufacture this chip. In reality, the trade-off between the precision and the area should be considered. To achieve an appropriate performance with the reasonable word length and quantization noise, more importantly, the smallest area consumption. Therefore, some simulations should be done to set the word lengths for the output signal and the inner signals in the 32-point DIF FFT.

In this design, the word lengths of the input signals have already been settled, which is 16 bits. The word lengths of the rest signals should be evaluated are: the twiddle factors on each stage, the multiplication results that executing with the twiddle factors, the final output of the 32-point DIF FFT.

Next, the performances of floating-point and fixed-point should be gauged. For the reason that the trade-off between the speed performance and the area consumption should be considered, the lower area consuming has been emphasized in this thesis project. Because of the limitation resource in FPGA board, during the procedure of realization 32-point DIF FFT, the balance between the input & output signals truncation bits on each stages and area consumption should be taken into consideration.

On the one hand, if the input & output signals truncation bits on each stages have been decreased, the quantization errors for every stage are accumulated to a larger error, the design precision could not approach to an appropriate level. On the other hand, if the truncation bits have been saved more, the quantization error is shrinking, on the contrary, the area for the design required to be bigger than before, which is not good for the manufacturing of the chip, more money should be paid as well.

After several tests have be made in MATLAB, with the consideration between the performance and the area, the proper word lengths for the inner signals haven be settled. After the truncation, the word length of the twiddle factors for all five stages have been set to 18 bits, after the multiplication operation with the twiddle factors, the outputs of the multipliers have been set to 19 bits, the output signal word length of the 32-point DIF FFT has been settled to 16 bits.

The un-truncated result and the truncated result are showing in the Figure 3.3.

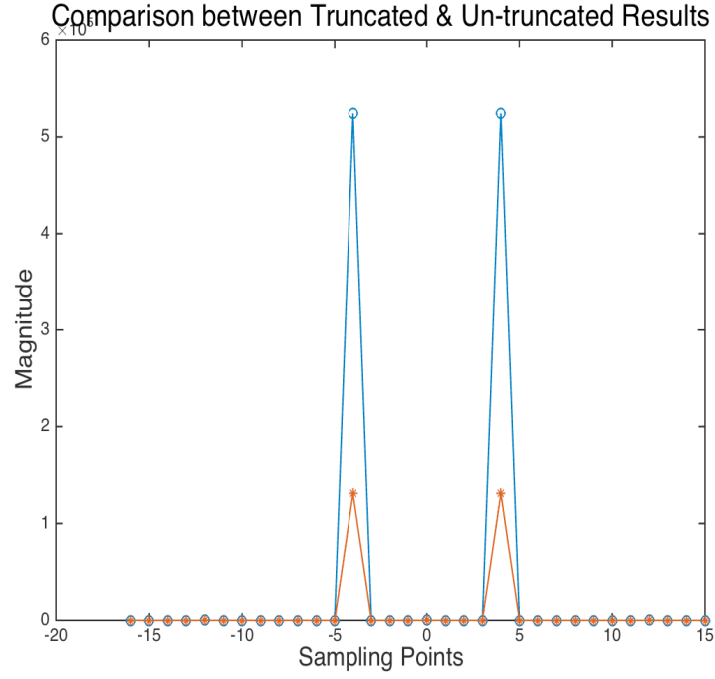


Fig. 3.3: Comparison between truncated and un-truncated results

The difference between the truncated result and the un-truncated result should be analyzed. The error function verification should be used at this point. By using the equation (3.1), the error number of bits could be seen in Figure 3.4.

$$\begin{aligned}
 err_lin_m &= untruncated_out - truncated_out \\
 err_abs_m &= abs(err_lin_m) \\
 err_log_m &= 20 \log_{10} (err_log_m / 2^{15}) \\
 bit_log_m &= err_log_m / 20 / \log_{10}(2)
 \end{aligned} \tag{3.1}$$

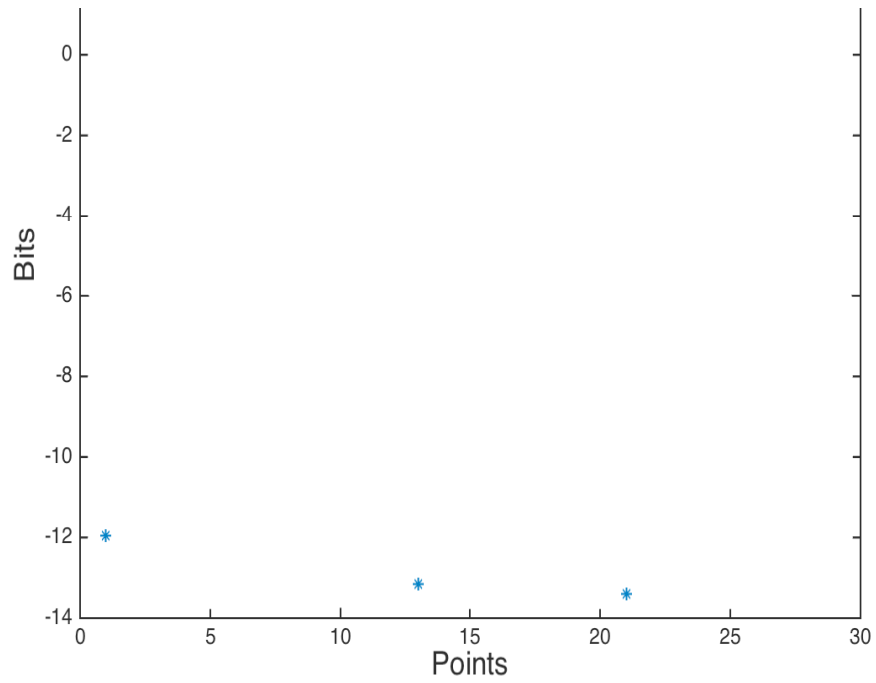


Fig. 3.4: Number of bits

In Figure 3.4, the maximum bits are approaching 14, which could be accepted.

CHAPTER 4

Hardware Implementation

In the hardware implementation part, the whole 32-point FFT is implemented in VHDL. For this design, the twiddle factors have been stored in the registers in the system instead of using ROM. Figure ?? illustrates the schematic top level of the 32-point DIF FFT hardware architecture.

The two input signals that shown in the figure 4.1 are i_data_i and i_data_q , which represent the real part and imaginary part respectively.

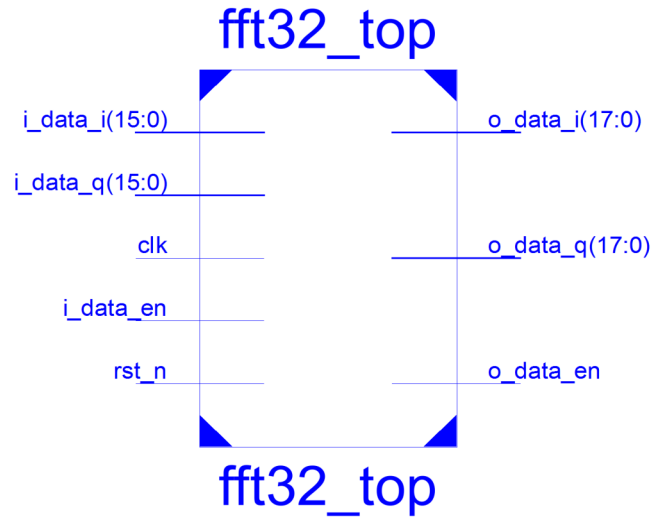


Fig. 4.1: The schematic top level of the 32-point FFT hardware architecture

Figure 4.2 shows the expanded top architecture. The blocks named *butterfly_16*, *butterfly_8*, *butterfly_4*, *butterfly_2*, *butterfly_1* are the five stages of the 32-point DIF FFT. The last block *order_adjust* is to adjust the sequence of the 32 outputs of the FFT.

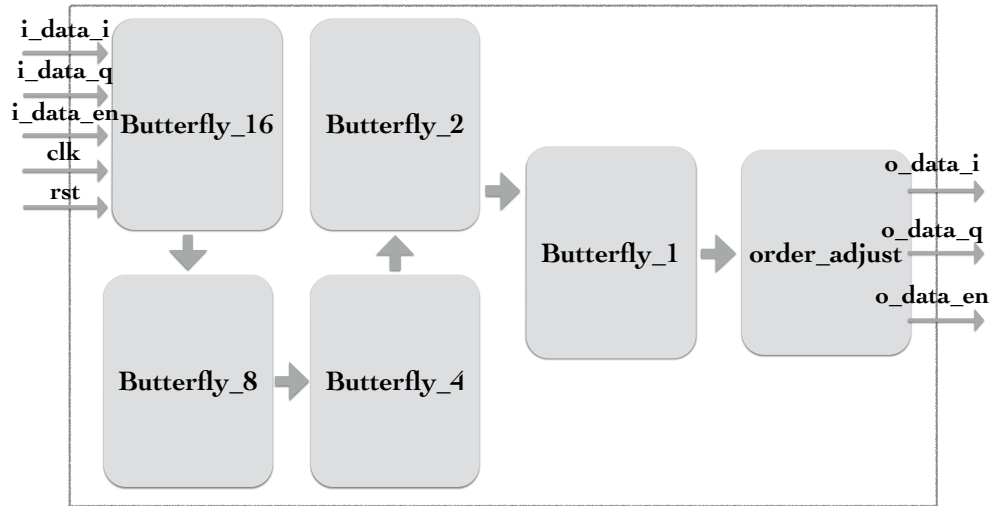


Fig. 4.2: The detailed schematic of the 32-point FFT hardware architecture

Implemented the design on the FPGA board-Xilinx Virtex 5 (XC5VLX110T). The device utilization summary could be drawn from the ISE tool, which illustrates in Table 4.1:

Table 4.1: Device utilization summary

| Slice Logic Utilization | Used | Available | Utilization |
|---------------------------|-------|-----------|-------------|
| Number of Slice Registers | 3,164 | 69,120 | 4% |
| Number of Slice LUTs | 2,101 | 69,102 | 3 % |
| Number of Occupied Slices | 1,098 | 17,280 | 6% |
| Number of bonded IOBs | 72 | 640 | 11% |
| Number of DSP48Es | 24 | 64 | 37% |

In the Table 4.1 , the utilization of the the DSP48E is 37%, which mainly consumed by the adders and the multipliers.

Figure 4.3 illustrates two 32-point DIF FFT hardware implementation

input signals. The input signal which shown in (a) has 6 periods per frame, in (b), it has 4 periods per frame. The sampling points are illustrated by dots.

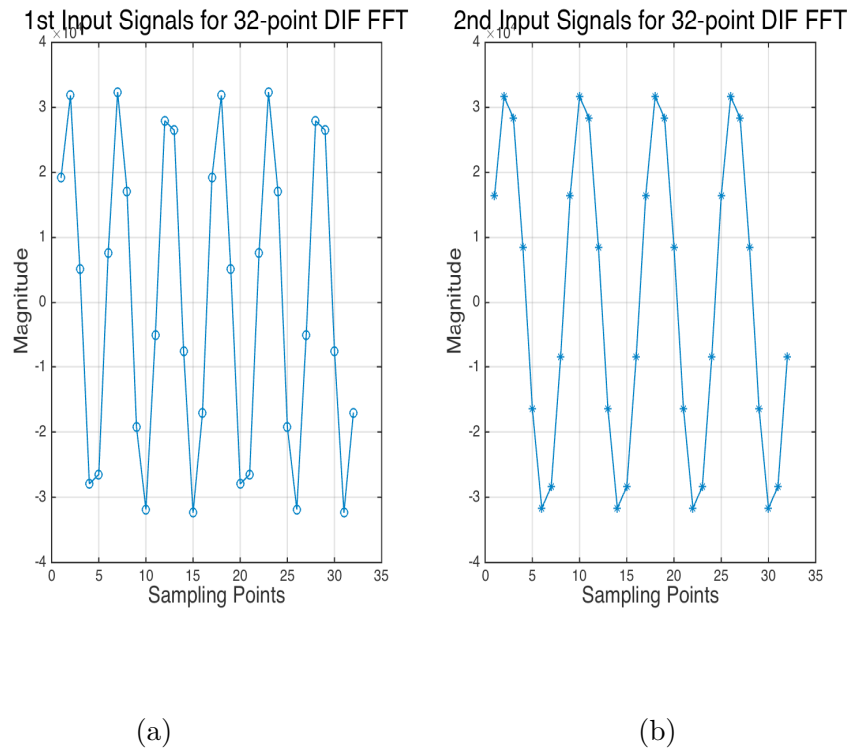


Fig. 4.3: Two 32-point DIF FFT hardware implementation input signals

Figure 4.4 illustrates two 32-point DIF FFT hardware implementation results.

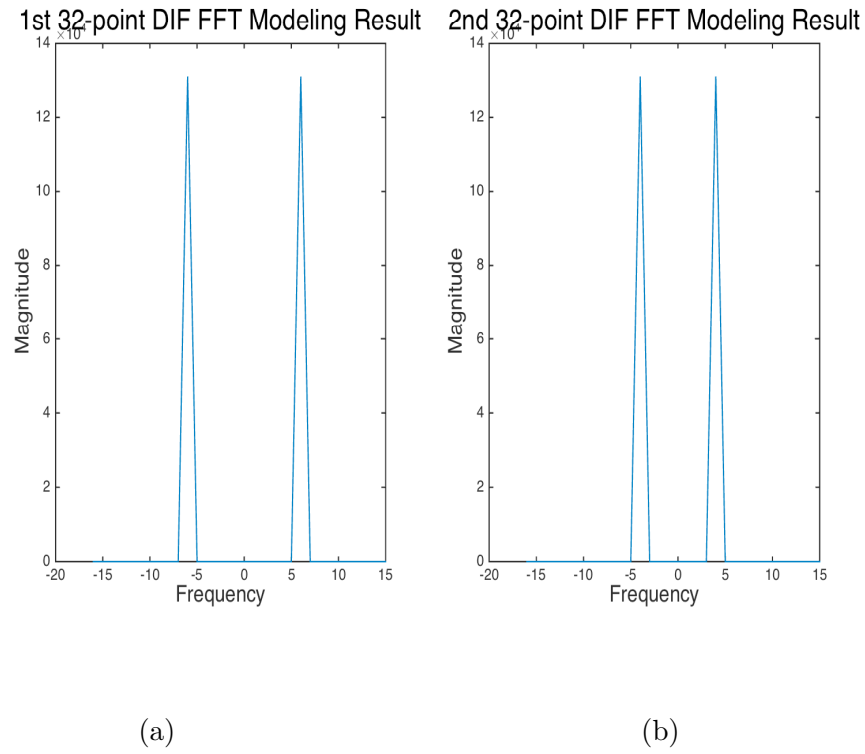


Fig. 4.4: Two 32-point DIF FFT hardware implementation results

Implement the same inputs to the 32-point DIF FFT Matlab model, the results are the same. It proves that the hardware implementation is acceptable.

ASIC Verification

Figure 5.1 demonstrates the data flow of the ASIC verification, which will be executed in $63nm$ CMOS process.

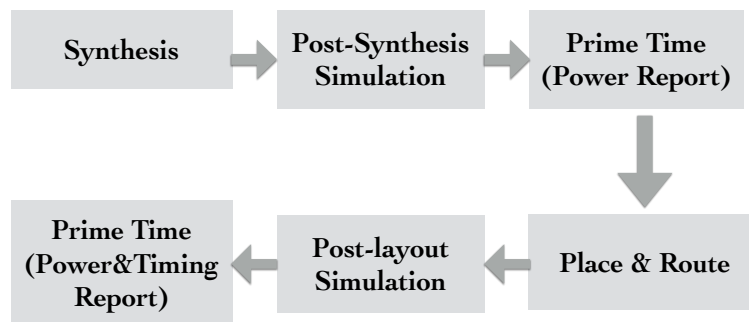


Fig. 5.1: The ASIC verification flow

5.1 Synthesis

The first step that shown in the figure 5.1 is *Synthesis*, which is execute in Design Vision. In figure 5.2, the executing steps of the synthesis are illustrating as follows:

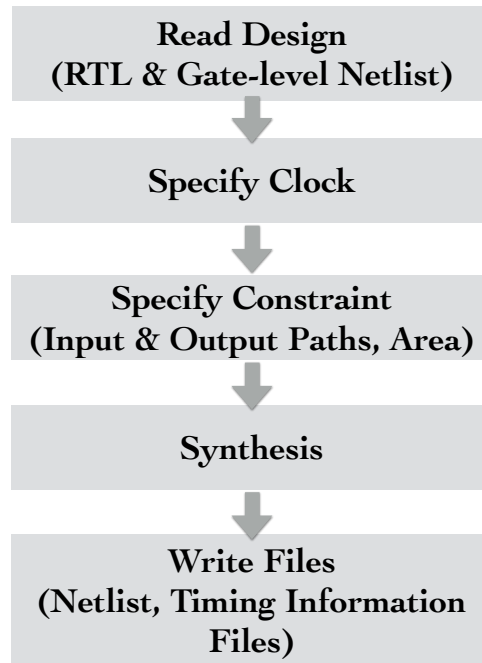


Fig. 5.2: The synthesis data flow

All the VHDL programming files is read into the Design Vision at the first step, then the clock constraint should be set to the design, which specify the clock period, the clock skew as well. The next step is Specify Constraints, the propagation delay of external & external logic(input path & output path) should be specified. For the chip design and manufacturing, the area parameter of the chip design should be saved, since the bigger the area, more money has to be consumed.

Table 5.1 illustrates the frequency and clock period with the specific parameters, which under maximum speed and 1.20V supply voltage. Table 5.2 illustrates the frequency and clock period with the specific parameters,

which under minimum area and 1.20V supply voltage. Table 5.3 demonstrates the area with the specific parameters, which under maximum speed and 1.20V supply voltage. Table 5.4 demonstrates the area with the specific parameters, which under minimum area and 1.20V supply voltage.

Table 5.1: Timing and maximum speed constraint and 1.20V supply voltage

| | Unit | LPHVT |
|---------|------|-------|
| Voltage | V | 1.20 |
| Speed | MHz | 434 |
| Time | ns | 2.3 |

Table 5.2: Timing and minimum area constraint and 1.20V supply voltage

| | Unit | LPHVT |
|---------|------|-------|
| Voltage | V | 1.20 |
| Speed | MHz | 138 |
| Time | ns | 7.2 |

Table 5.3: Area at maximum speed constraint and 1.20V supply voltage

| | Unit | LPHVT |
|---------|--------|-------|
| Voltage | V | 1.20 |
| Area | mm^2 | 0.133 |

Table 5.4: Area at minimum area constraint and 1.20V supply voltage

| | Unit | LPHVT |
|---------|--------|-------|
| Voltage | V | 1.20 |
| Area | mm^2 | 0.117 |

In the end of the synthesis, the nettles(.v file), two files including timing information(.sdf file & .sdc file)should be written from the design vision. These three files should be converted to *Modelsim* to run the *Post-synthesis*, which would verify again of all the parameter that you have been set to the Design Vision are correct.

5.2 Place & Route

Figure 5.3 illustrates 32-point DIF FFT chip design outlook, which has been done in the Cadence SoC Encounter. The total size of the chip is $480*400\mu m^2$ without the pads. The overall cell placement density achieves 75.1%, which is quite compact. The cell replacement density could be increased if needed. However, others requirements need to be achieved as well, such as: the setup violation and the hold violation parameters, the 75.1% density has been settled.

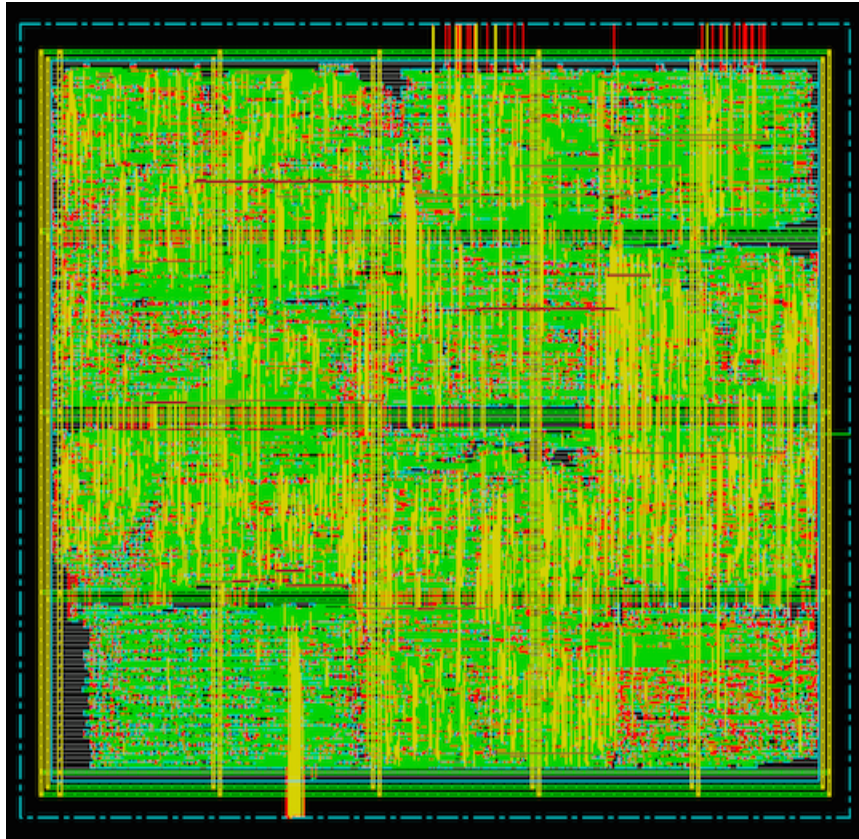


Fig. 5.3: 32-point DIF FFT chip design outlook

Once the layout has been settled, it is time to measure the setup viola-

tion and the hold violation, the parameter-*Violation Path* for both of them should be 0, the WNS and TNS should be 0 as well, which stands for the Worst Negative Slack and Total Negative Slack separately.

After all the constrains have been satisfied, the SDF(Synopsys Delay Format) file and the design Netlist should be written from the Encounter, waiting to be used in the Post-layout Simulation.

The input files for the post-layout simulation are .v file and .sdf file, they are all generated from the Place & Route.

5.3 Prime Time

Figure 5.4 describes the input files and output reports of the *Prime Time*.

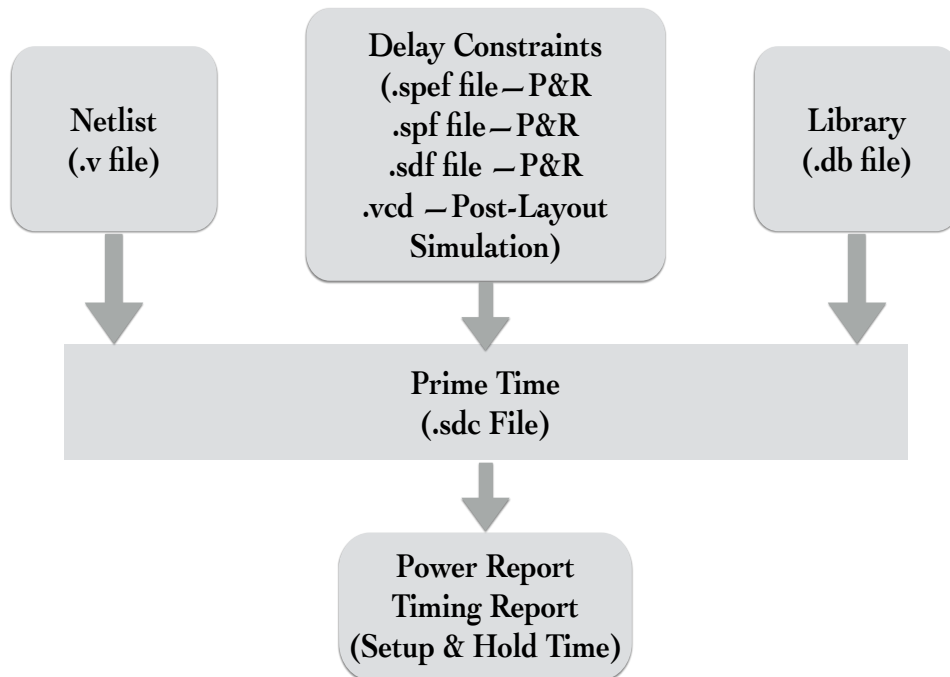


Fig. 5.4: The prime time flow

Since it is good to see whether there is a difference about the power reports between the *Pre-Place & Route* and *After-Place & Route*. The input

files should be slightly different for these two case.

The power reports before the Place & Route are illustrated in the Table 5.5:

Table 5.5: Prime time power report I

| Power Consumption | Frequency(MHz) | LPHVT(W) |
|---------------------|----------------|-----------|
| Net Switching Power | 10 | $21.7e-3$ |
| Cell Internal Power | 10 | $15.0e-3$ |
| Cell Leakage Power | 10 | $60.1e-6$ |
| Total Power | 10 | $3.7e-2$ |

The power reports after the Place & Route are illustrated in the Table 5.6 :

Table 5.6: Prime time power report II

| Power Consumption | Frequency(MHz) | LPHVT(W) |
|---------------------|----------------|-----------|
| Net Switching Power | 10 | $52.1e-3$ |
| Cell Internal Power | 10 | $34.6e-3$ |
| Cell Leakage Power | 10 | $70.1e-6$ |
| Total Power | 10 | $8.7e-2$ |

Table 5.7 illustrates with three power parameters, which under the situation that the frequency equals to 10 MHz, the supply voltage is 1.20V. Table 5.8 shows with three power parameters, which under the situation that the frequency reaches to the maximum value, the supply voltage is 1.20V.

Table 5.7: Power consumption at 10MHz and 1.20V supply voltage with maximum speed constraint

| Power Consumption | Frequency(MHz) | LPHVT(W) |
|---------------------|----------------|-----------|
| Net Switching Power | 10 | $15.7e-3$ |
| Cell Internal Power | 10 | $12.0e-3$ |
| Cell Leakage Power | 10 | $50.1e-6$ |
| Total Power | 10 | $2.8e-2$ |

Table 5.8: Power consumption at maximum frequency and 1.20V supply voltage with maximum speed constraint

| Power Consumption | Frequency(MHz) | LPHVT(W) |
|---------------------|----------------|-----------|
| Net Switching Power | 104 | $26.1e-2$ |
| Cell Internal Power | 104 | $18.4e-2$ |
| Cell Leakage Power | 104 | $70.3e-4$ |
| Total Power | 104 | $44.6e-2$ |

As shown before, the power which consumed by the register and combinational logic are decreasing, on the contrary, the clock network power consumption is increasing. Because in the Place & Route part, the constraints about the clock are quite tight, after the Place & Route, the clock tree in the design has been synthesized. The violation delays have been fixed both in the setup time and hold time.

Figure 5.5 illustrates the power consumption varies with different frequencies from 10Hz to 1GHz. The power consumption increased steadily until the frequency reached to 104MHz, then the power consumption keeps stable afterwards.

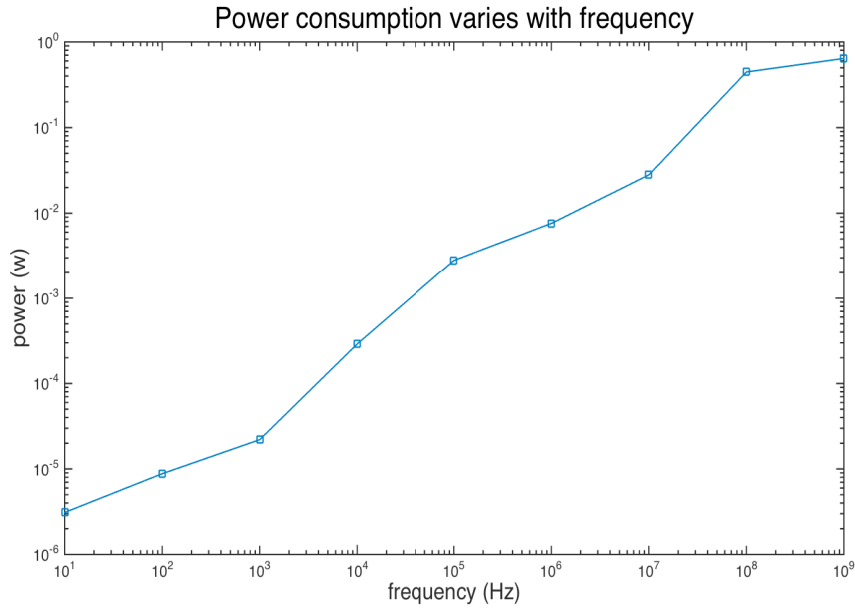


Fig. 5.5: Power consumption varies with frequency

The timing reports for the setup time and hold time are shown as follow in Table 5.9:

Table 5.9: Slacks for setup time and hold time

| Timing Group | Statistics(<i>ns</i>) |
|------------------------------|-------------------------|
| Critical Path for Setup Time | 3.12 |
| Critical Path for Hold Time | 0.85 |

When the report timing has been executed, the *Prime Time* would default to the longest path, which is critical path in this design. As it can be seen in the Table 5.9, the critical path for the setup time and hold time are both positive.

Conclusion & Future Work

6.1 Conclusion

This thesis is about the radix-2 32-point DIF FFT which has been implemented in $65nm$ CMOS technology.

Based on the radix-2 algorithm, it presents the superiority of low area consumption and longer throughput. More specifically, the usage of the adders and multipliers are shrinking significantly.

Since the hardware architecture of the design includes 5 stages with their twiddle factors separately. For each stage, one multiplier and two adders are being consumed. The number of the registers are the half of the input signals for each stage. The area depletion would decrease because of the less adders and the multipliers are being used in the implementation. For the proper trade-off between the accuracy and the area, the word length of the twiddle factors for all five stages have been truncated to 18 bits, the word length of the final output of the design has been truncated to 16 bits. After being synthesized, the clock frequency of the design is measured, which is $1MHz$, the critical path is $7.84ns$. After the Place& Route, the die size of the design is $480 * 400um^2$. By going through the Prime Time, the total

power of the design is $0.0323mW$. The critical paths for the setup time and hold time are $3.12ns$ and $0.85ns$ separately.

6.2 Future Work

By improving the algorithm of the design, some parameters could be improved in the future work, such as: chip area, critical path delay, power consumption. The *Harmonized Parabolic Synthesis Methodology* could be implemented in this design. Furthermore, with the increasing number of the points for the FFT, the ROM could be used in the design to store the more complexed twiddle factors instead of using the registers. The twiddle factors could be processed to the specific form to delete the similar forms. The logic for how to selecting the specific twiddle factors could be improved in a more intelligent way. The truncation for the inter signals for the architecture could be settled in a more detailed way to achieve the best precision criteria.

In the next stage, the design could be implemented in the FPGA, for the further usage of the other industries, such as: the medical industry, the entertainment industry, etc.

APPENDIX A

Appendix 1

[1] Asmita Haveliya, "Design and Simulation of 32-Point FFT Using Radix-2 Algorithm for FPGA Implementation," Amity University, page 167, 2012.

[2] John G. Proakis and Dimitris G. Manolakis, "Introduction to Digital Signal Processing," MacMillian, page 683 and 696, 1988.

[3] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comp., vol. 19, pp, 297-301, April 1965.



LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2015-461

<http://www.eit.lth.se>