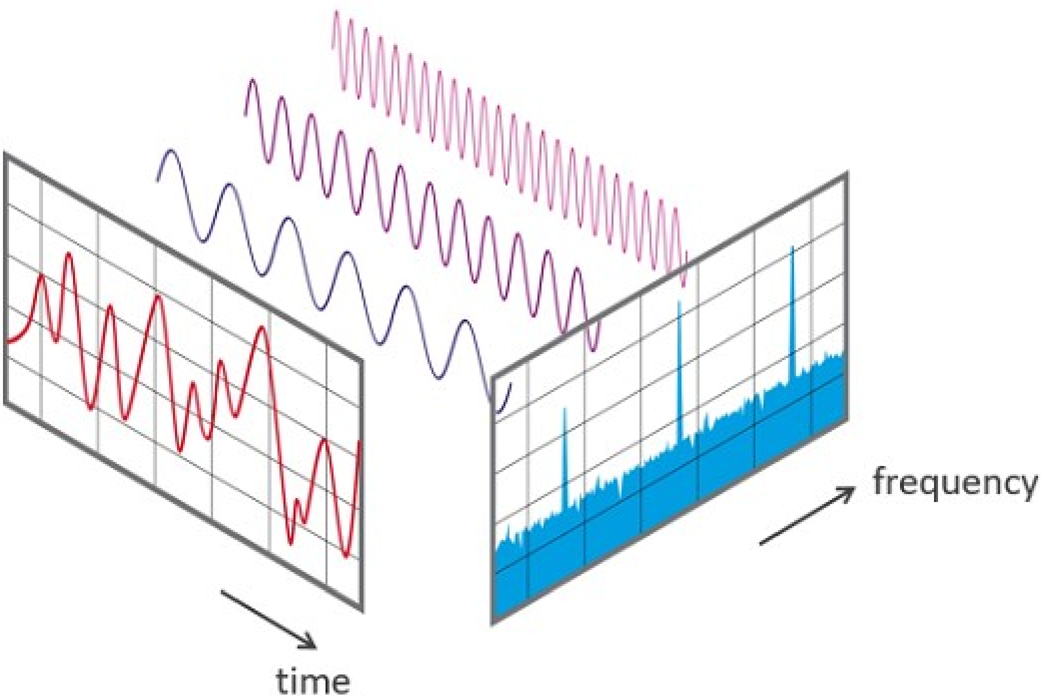
2025 Digital IC Design

Homework 3: Fast Fourier Transform

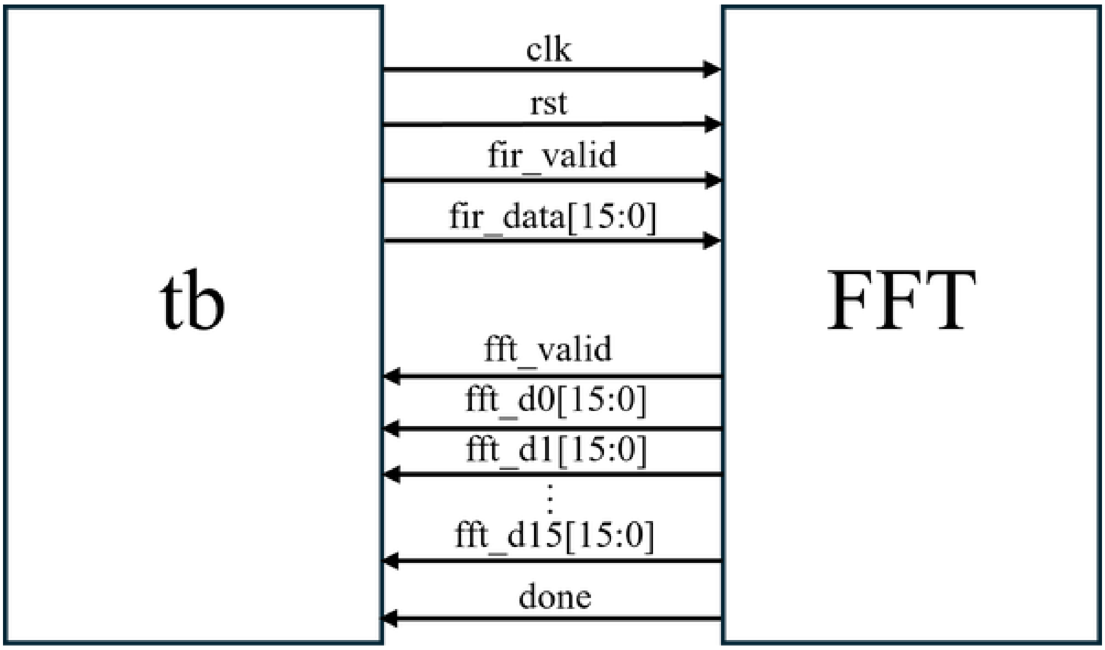
1. Introduction

The goal of this assignment is to design a circuit that converts time-domain signals into frequency-domain signals. The input is a time-domain signal that has passed through an FIR filter. After being processed by multiple butterfly units, it is converted into a frequency-domain signal.



2. Specification

2.1. System Block Diagram

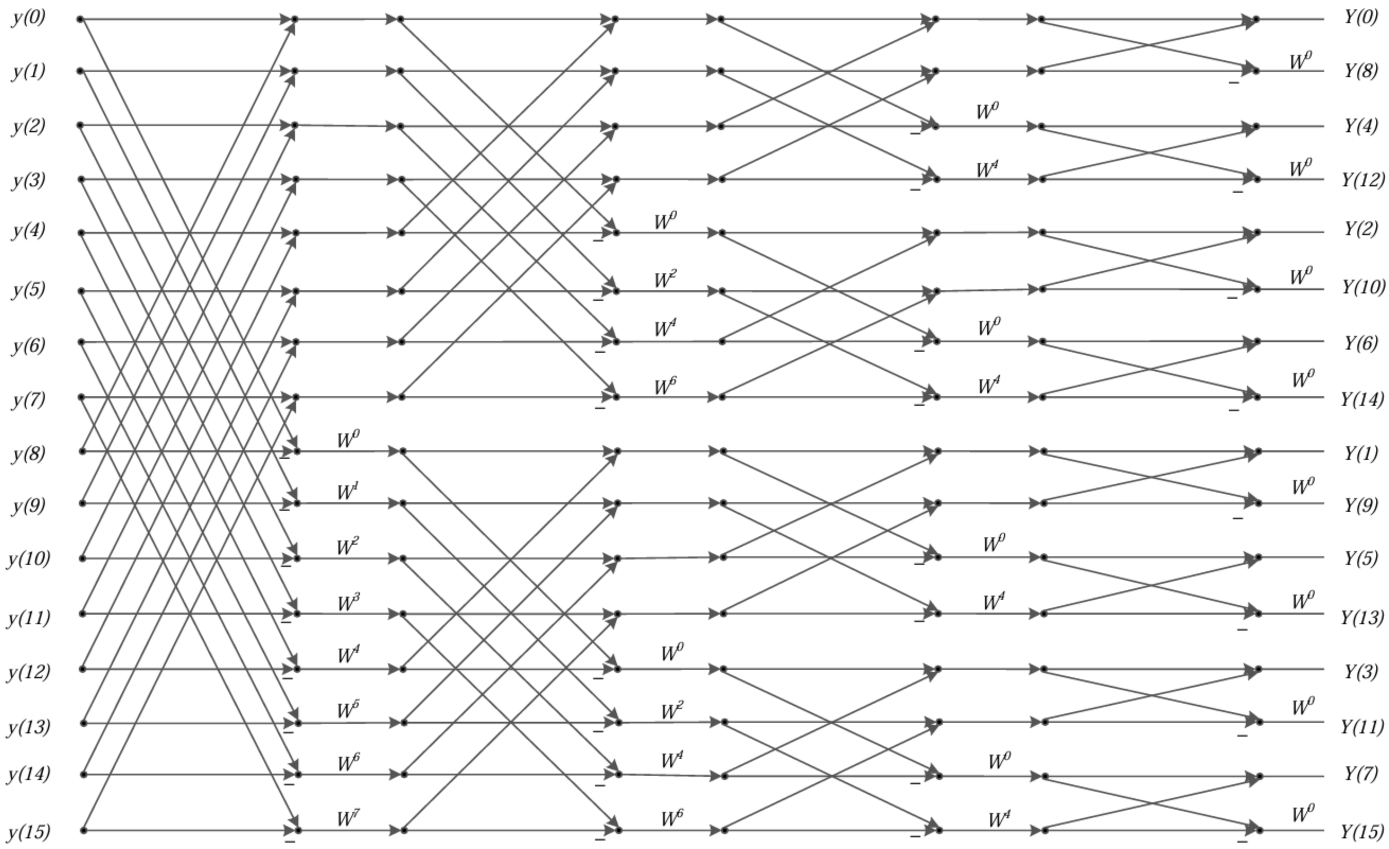


2.2. System I/O Interface

Signal Name	I/O	Width	Description
clk	Input	1	System clock signal. System should be triggered by the positive edge of clock.
rst	Input	1	System reset signal. Active high, asynchronous reset.
fir_valid	Input	1	When this signal is high, indicates **fir_data** input is valid.
fir_data	Input	16	16-bit FIR filter data input bus.
fft_valid	Output	1	The FFT data valid signal. Should be high for two consecutive cycles each time.
fft_d0~fftd15	Output	16	The FFT output data buses.
done	Output	1	When the controller completes writing to **IRAM**, it sets **done** to **high** to indicate completion.

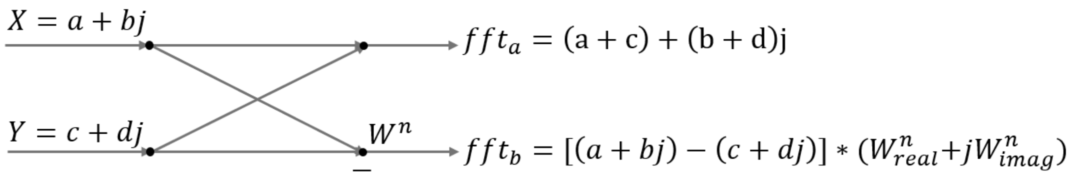
2.3. Functional Description

The Fast Fourier Transform (FFT) used in this homework requires the implementation of a 16-point FFT. The hardware architecture is shown in the below. This FFT circuit is designed to convert time-domain signals into frequency-domain signals.



2.3.1 Butterfly Unit Description

The butterfly unit is the fundamental unit for performing FFT. A 16-point FFT can be constructed using multiple butterfly units. The hardware architecture of a butterfly unit is shown below. In the lower path, a minus sign (−) indicates the operation of subtracting Y data from X data. W^n represents the coefficient of the FFT, which consists of a real part (W^n_{real}) and an imaginary part (W^n_{imag}). When computing fft_b, complex arithmetic operations must be performed, requiring separate recording and computation of the real and imaginary components during the process.



$$fft_b = [(a + bj) - (c + dj)] * (W^n_{real} + jW^n_{imag})$$

After cross-multiplication, the result can be obtained as follows:

$$fft_b = (a - c)W^n_{real} + j(a - c)W^n_{imag} + j(b - d)W^n_{real} + (b - d)W^n_{imag}$$

It can be simplified as follows:

$$fft_b = (a - c)W^n_{real} + (b - d)W^n_{imag} + j[(a - c)W^n_{imag} + (b - d)W^n_{real}]$$

Finally, we can get:

$fft_a\ real\ part$	$a + c$
$fft_a\ imaginary\ part$	$b + d$
$fft_b\ real\ part$	$(a - c)W^n_{real} + (d - b)W^n_{imag}$
$fft_b\ imaginary\ part$	$(a - c)W^n_{imag} + (b - d)W^n_{real}$

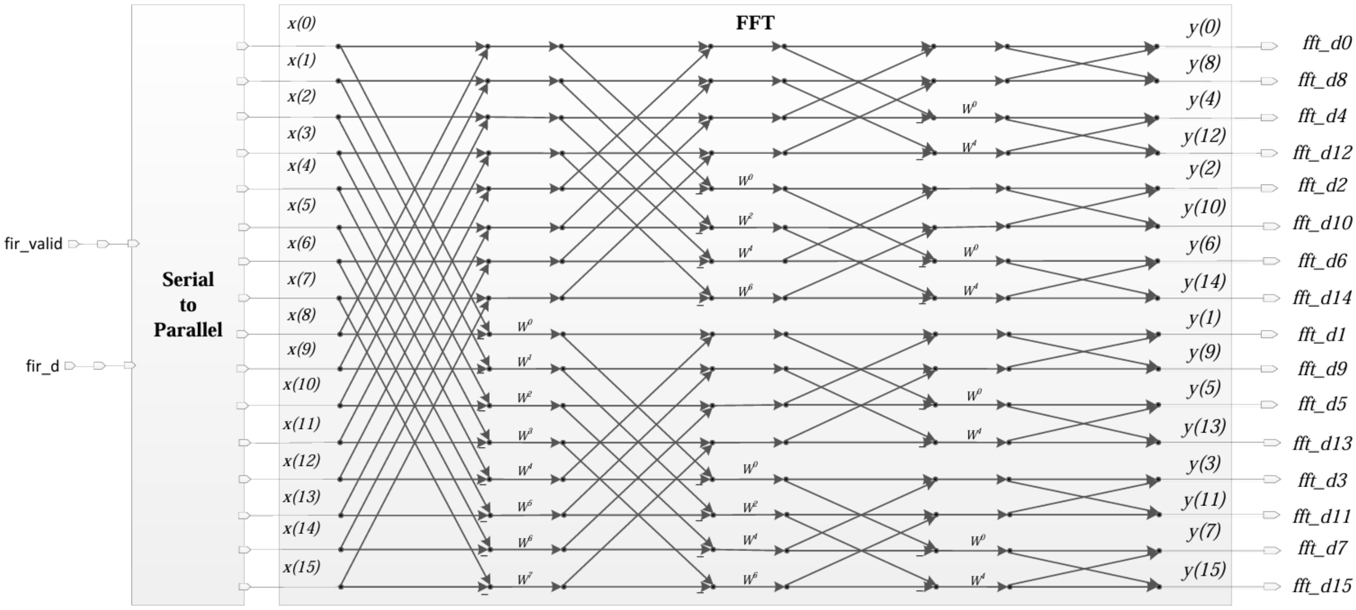
Following is the coefficient table for the 16-point FFT in hexadecimal format

	$real\ part$	$imaginary\ part$		$real\ part$	$imaginary\ part$
W_8^0	32'h00010000	32'h00000000	W_8^4	32'h00000000	32'hFFFF0000
W_8^1	32'h0000EC83	32'hFFFF9E09	W_8^5	32'hFFFF9E09	32'hFFFF137D
W_8^2	32'h0000B504	32'hFFFF4AFC	W_8^6	32'hFFFF4AFC	32'hFFFF4AFC
W_8^3	32'h000061F7	32'hFFFF137D	W_8^7	32'hFFFF137D	32'hFFFF9E09

	Sign bit	Integer part	Fractional part
W_k^n	1 bit	7 bit	8 bit

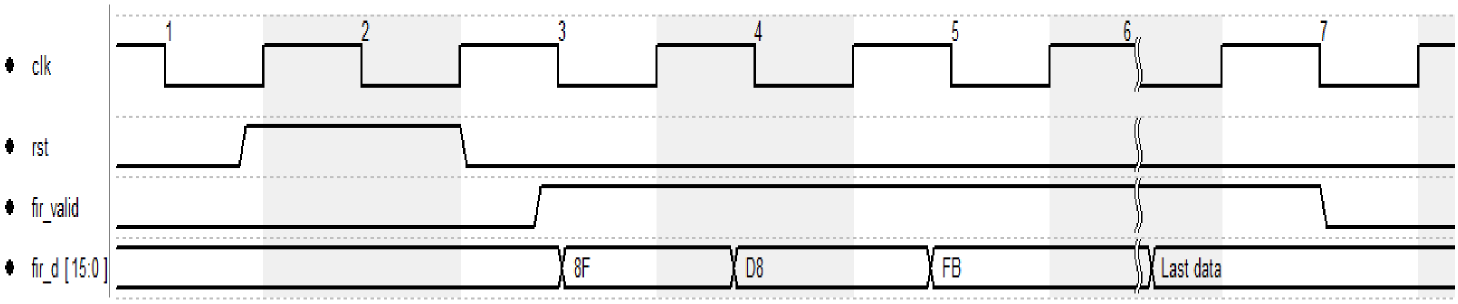
2.3.2 Serial to Parallel

Before performing FFT calculations, the serial data must first be converted into parallel signals for 16-point processing. Every 16 fir_d input values can be used to compute one set of FFT result. This conversion ensures that the output data meets the timing specifications for FFT output and matches the timing for the FIR filter input.



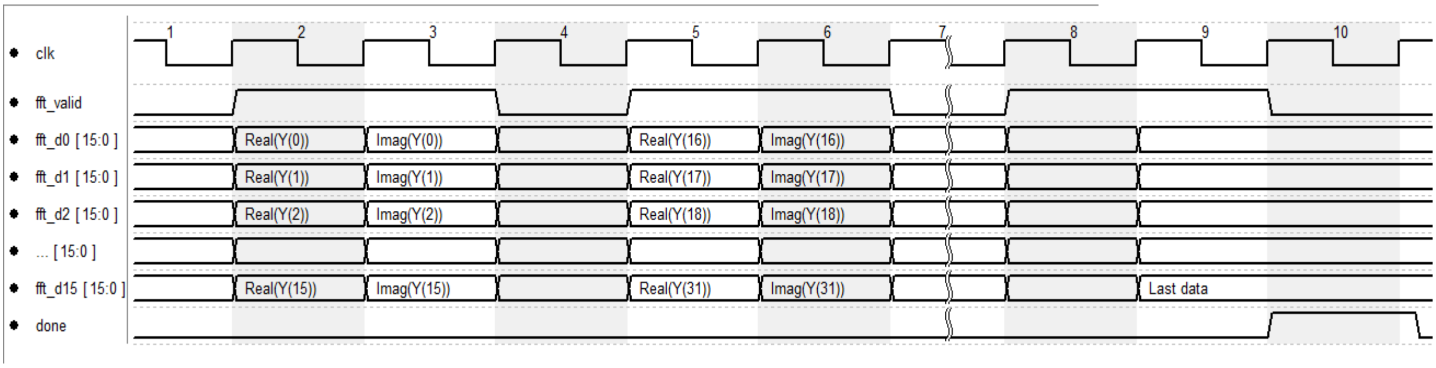
2.3.3 Input and Output Timing Specification

When the host sets the **fir_valid** signal to high, the **fir_d** bus from the host will send a time-domain data on each clock cycle, as shown in the timing specification below. The data format of the **fir_d** signal from the host is 16 bits, consisting of 1 bit for the **sign bit**, 7 bits for the **integer part**, and 8 bits for the **fractional part**.



	Sign bit	Integer part	Fractional part
fir_d	1 bit	7 bit	8 bit

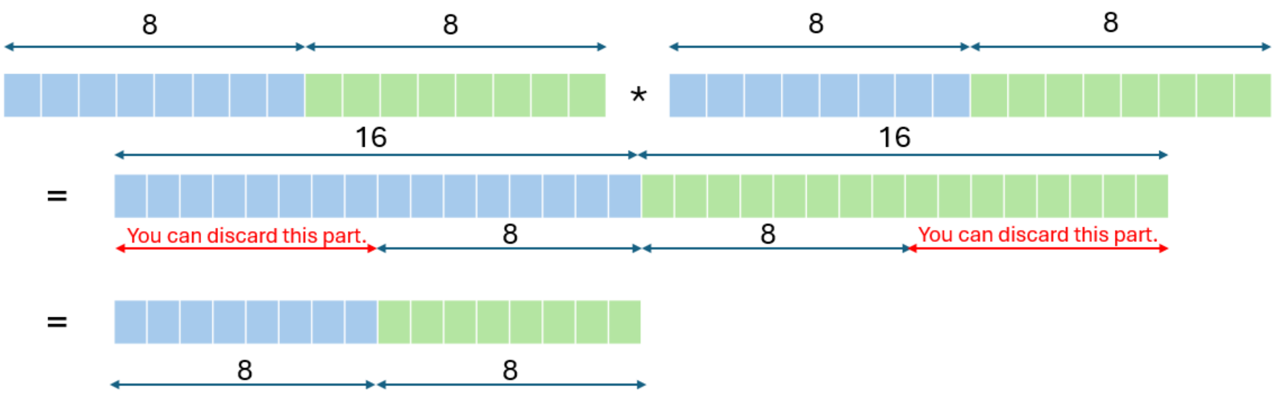
After the data undergoes FFT processing, if the **fft_valid** signal is set to high, it indicates that **fft_d0 to fft_d15** will start transmitting data to the host. The testbench will simultaneously perform data comparison. Notice that during each data transmission, the **fft_valid** signal must be high for two cycles: the **first cycle** is used to transmit the **real part**, and the **second cycle** is used to transmit the **imaginary part**. The data timing specification is shown in the figure below. The data format of the **fft_d0 to fft_d15** signal is 16 bits, consisting of 1 bit for the **sign bit**, 7 bits for the **integer part**, and 8 bits for the **fractional part**.

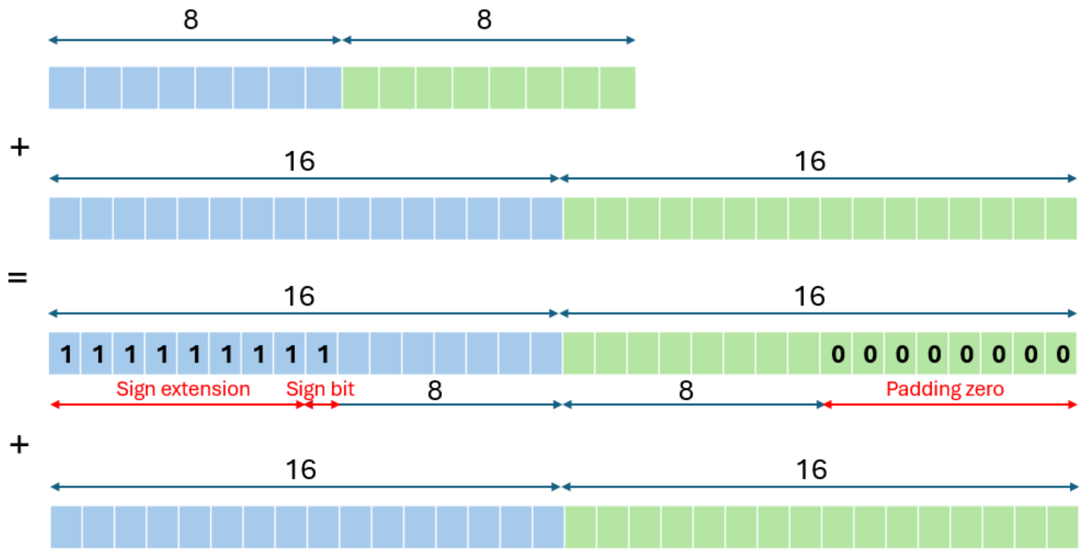


fft_d0 ~ fft_d15	Sign bit	Integer part	Fractional part
	1 bit	7 bit	8 bit

2.3.4 Fixed Point Operation

When performing fixed-point addition and subtraction, ensure that the **decimal points are aligned**. If necessary, you must append zeros at the LSB and perform **sign extension** at the MSB. When performing fixed-point multiplication, an 8-bit integer and an 8-bit fractional part will result in a 16-bit integer and a 16-bit fractional part. When outputting to **fft_dx**, you can discard the first 8 bits and the last 8 bits. Although this will result in a loss of precision, some tolerance will be given during testbench verification.





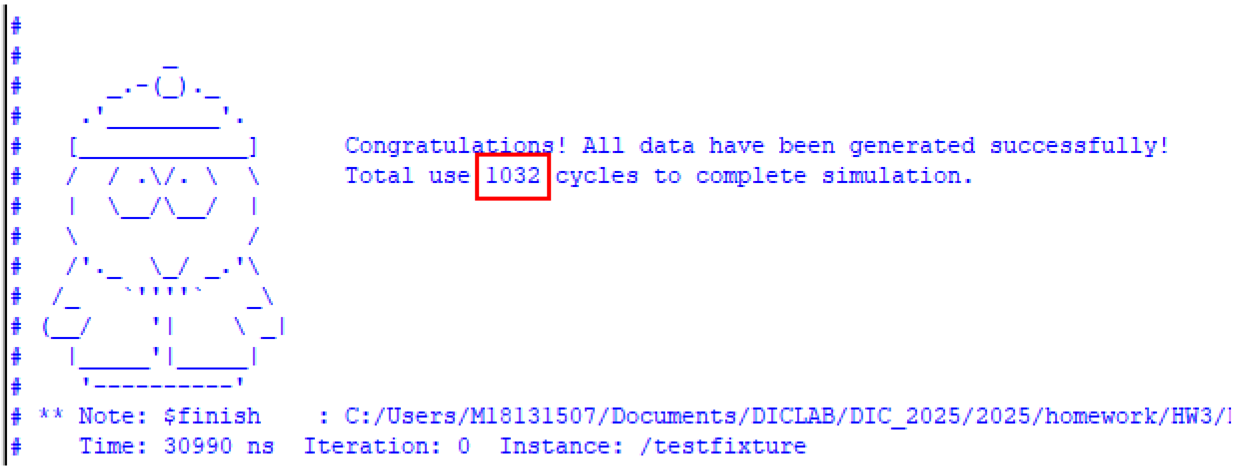
2.4 File Description

File Name	Description
FFT.v	The module of FFT.
testfixture.v	Testbench file.
PatternX_FIR.dat	FIR input data.
GoldenX_FFT_real.dat	Golden data X for real part of FFT result
GoldenX_FFT_imag.dat	Golden data X for imaginary part of FFT result
Real_Value_Ref.dat	Real coefficients of W_k^n.
Imag_Value_Ref.dat	Imaginary coefficients of W_k^n.

3. Scoring:

3.1. Functional Simulation [40%]

While there are two test patterns in this homework, the grading rule for this homework is that twenty points can be obtained by passing one test pattern. After the simulation, the result will be shown in ModelSim terminal as shown in the below figure. Please don't design specifically for the test pattern. Otherwise, you will get 0 point.



3.2. Pre-Layout Simulation [40%]

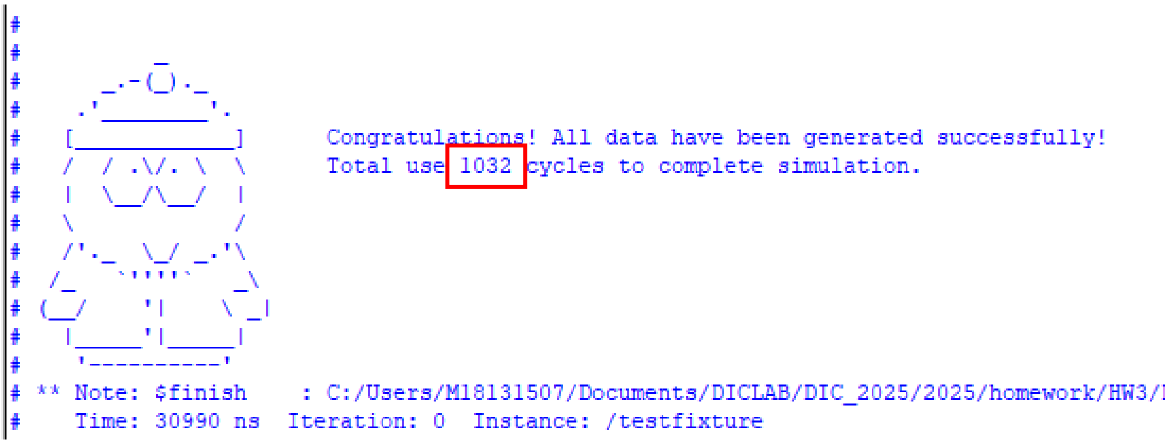
3.2.1. Synthesis

Your code should be synthesizable. After it is synthesized in Quartus, files named FFT.vo and FFT_v.sdo will be obtained.

DEVICE：Cyclone IV E - EP4CE55F23A7

3.2.2. Simulation

All of the results should be generated correctly using FFT.vo and FFT_v.sdo, and you will get the following message in ModelSim simulation.



3.3. Performance [20%]

The performance is scored by the total logic elements, total memory bit, and embedded multiplier 9-bit element your design used in gate-level simulation and the simulation time your design takes.

Flow Summary

Flow Status	Successful - Wed Apr 09 14:02:05 2025
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	FFT
Top-level Entity Name	FFT
Family	Cyclone IV E
Device	EP4CE55F23A7
Timing Models	Final
Total logic elements	3,029 / 55,856 (5 %)
Total registers	1741
Total pins	277 / 325 (85 %)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	62 / 308 (20 %)
Total PLLs	0 / 4 (0 %)

The performance score will be decided by your ranking in all received homework. Only designs that passed gate-level simulation and meet resource limitations will be considered in the ranking. Otherwise, you can't get performance score. Notice that 5 points will be reduced if the simulation clock period is larger than 30 ns.

The scoring standard: (The smaller, the better)

Scoring = Area cost * Timing cost

Area cost = Total logic elements + Total registers + Total memory bits + 9 * Embedded Multiplier 9-bit elements

Timing cost = Total cycle used * Clock period

4. Submission

4.1. File Submission

You should classify your files into two directories and compress them to .zip format. The naming rule is StudentID.zip. If your file is not named according to the naming rule, you will lose five points.

	RTL
*.v	All of your Verilog RTL code
	Pre-Layout
*.vo	Gate-Level netlist generated by Quartus
*.sdo	SDF timing information generated by Quartus
	Documentary
Report.pdf	The report file of your design (in pdf)

- StudentID.zip
  - StudentID
    - RTL
      - *.v (All of your Verilog RTL code)
    - Pre-Layout
      - *.vo, *.sdo
    - Documentary
      - Report.pdf

4.2. Report File

Please follow the spec of report. You are asked to describe how the circuit is designed as detailed as possible.

4.3. Notes

a. Please submit your .zip file to folder HW3 in moodle.
   Deadline: 2025/05/04 23:55
b. Late submissions will result in a penalty of 5 points per day.

5. Appendix

This is the result of the first set of **fir_data** after passing through each stage of butterfly unit.

Stage1		Stage2	
Real	Imag	Real	Imag
32'hFFFFDA00	32'h00000000	32'hFFFFFE00	32'h00000000
32'hFFFFEA00	32'h00000000	32'h00001900	32'h00000000
32'hFFFFFE00	32'h00000000	32'h00002F00	32'h00000000
32'h00001300	32'h00000000	32'h00003F00	32'h00000000
32'h00002400	32'h00000000	32'hFFFFB600	32'h00000000
32'h00002F00	32'h00000000	32'hFFFFCF36	32'h000030CA
32'h00003100	32'h00000000	32'h00000000	32'h00003300
32'h00002C00	32'h00000000	32'h000011AD	32'h000011AD
32'h00014400	32'h00000000	32'h00014400	32'hFFFEA000
32'h0001A370	32'hFFFF5244	32'h00016079	32'hFFFEB097
32'h0001645F	32'hFFE9BA1	32'h00017A4A	32'hFFFEB18C
32'h0000B379	32'hFFFE4EB5	32'h000189CF	32'hFFFEA77C
32'h00000000	32'hFFFEA000	32'h00014400	32'h00016000
32'hFFFFBD09	32'hFFFF5E53	32'h00014F67	32'hFFFE9F8B
32'h000015EB	32'h000015EB	32'hFFFE85B6	32'hFFFEB18C
32'h0000D656	32'h000058C7	32'hFFFEA780	32'h000189CD

Stage3		Stage4	
Real	Imag	Real	Imag
32’h00002D00	32’h00000000	32’h00008500	32’h00000000
32’h00005800	32’h00000000	32’hFFFFD500	32’h00000000
32’hFFFFCF00	32’h00000000	32’hFFFFCF00	32’h00002600
32’h00000000	32’h00002600	32’hFFFFCF00	32’hFFFFDA00
32’hFFFFB600	32’h00003300	32’hFFFF96E3	32’h00007577
32’hFFFFE0E3	32’h00004277	32’hFFFFD51D	32’h FFFFF089
32’hFFFFB600	32’hFFFFCD00	32’hFFFFD51D	32’h00000F77
32’h00001F1D	32’h00004277	32’hFFFF96E3	32’hFFFF8A89
32’h0002BE4A	32’hFFFD518C	32’h0005A892	32’hFFFAA99F
32’h0002EA48	32’hFFFD5813	32’hFFFFD402	32’hFFFFF979
32’hFFFFC9B6	32’hFFFFEE74	32’hFFFFD2D1	32’h000017CA
32’h0000091B	32’h00002956	32’hFFFFC09B	32’hFFFFC51E
32’hFFFFC9B6	32’h0000118C	32’hFFFFC09D	32’h00003AE4
32’hFFFFF6E7	32’h00002958	32’hFFFFD2CF	32’hFFFFE834
32’h0002BE4A	32’h0002AE74	32’hFFFFD408	32’h0000068D
32’hFFFD15BE	32’hFFFD5819	32’h0005A88C	32’h0005565B