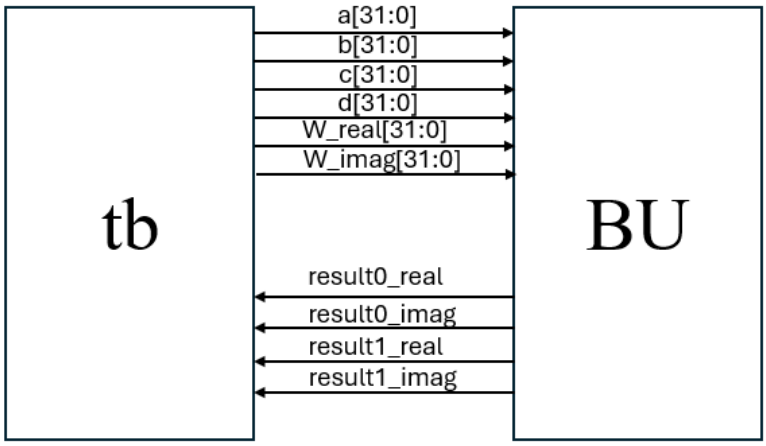
2025 Digital IC Design

Final Exam

Q1: Butterfly unit

The goal of this assignment is to design a circuit that uses butterfly units to convert a time-domain signal into a frequency-domain signal. The input, filtered by an FIR filter, is processed through multiple butterfly units to perform the FFT.

1.1 Block Overview



1.2 System I/O Interface

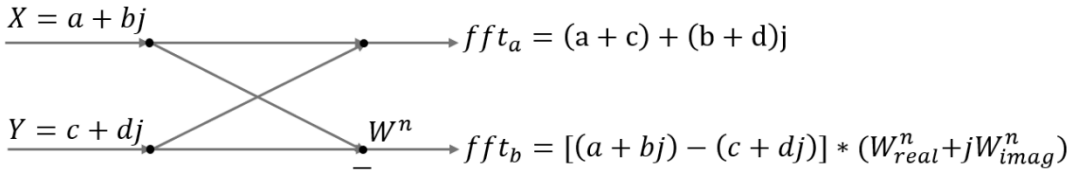
Signal	I/O	Width	Description
a	Input	32	The **real part** of the **first** input sample
b	Input	32	The **imaginary part** of the **first** input sample
c	Input	32	The **real part** of the **second** input sample
d	Input	32	The **imaginary part** of the **second** input sample
W_real	Input	32	The **real part** of the twiddle factor
W_imag	Input	32	The **imaginary part** of the twiddle factor
Result0_real	Output	32	The **real part** of the **first** output sample
Result0_imag	Output	32	The **imaginary part** of the **first** output sample
Result1_real	Output	32	The **real part** of the **second** output sample
Result1_imag	Output	32	The **imaginary part** of the **second** output sample

Every I/O Port	Sign bit	integer part	fractional part
	1 bit	15 bit	16 bit

1.3 Function Description

1.3.1 Butterfly Unit

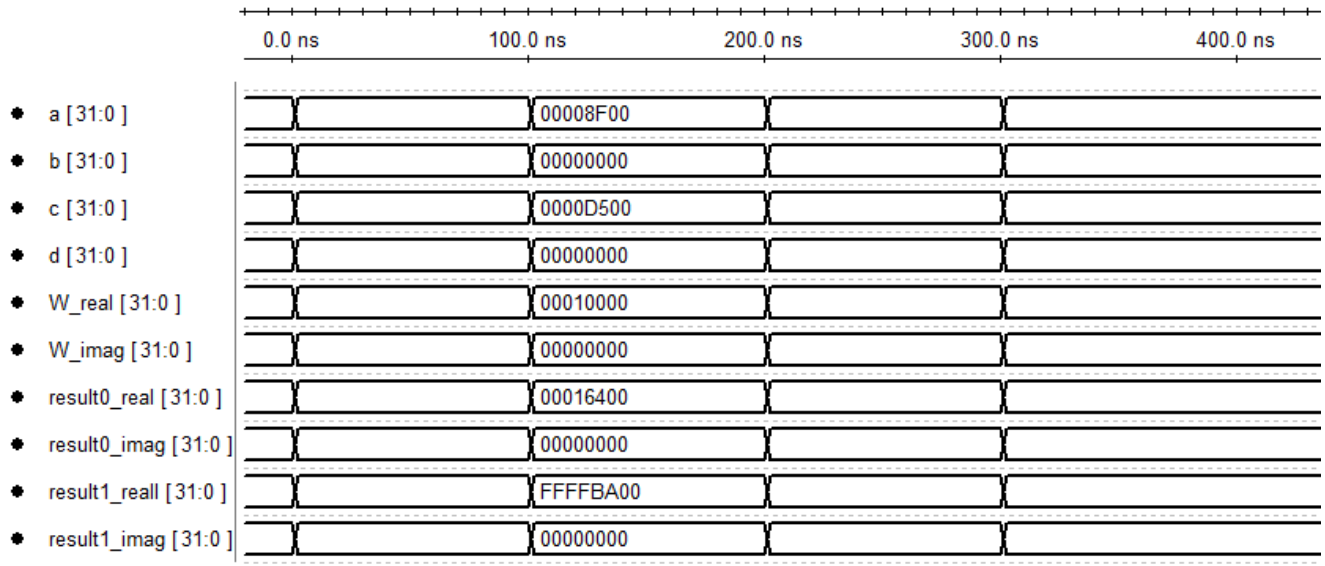
The butterfly unit is the fundamental unit for performing FFT. A 32-point FFT can be constructed using multiple butterfly units. The hardware architecture of a butterfly unit is shown below. In the lower path, a minus sign (−) indicates the operation of subtracting Y data from X data. W^n represents the coefficient of the FFT, which consists of a real part (W^n_{real}) and an imaginary part (W^n_{imag}). When computing fftb, complex arithmetic operations must be performed, requiring separate recording and computation of the real and imaginary components during the process.



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

1.3.2 Input and Output Timing Specification

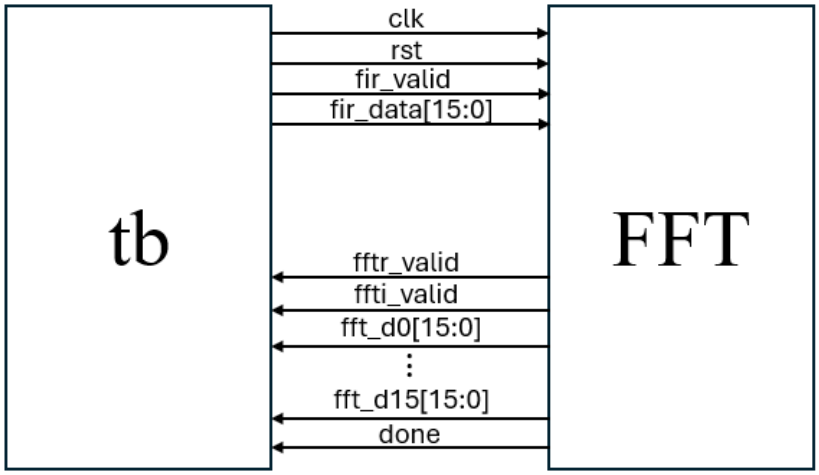
The Butterfly Unit (BU) is purely combinational logic. It produces an output immediately upon receiving the input, and the testbench verifies the result right away.



Q2: 32-points FFT

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.

2.1 Block Overview

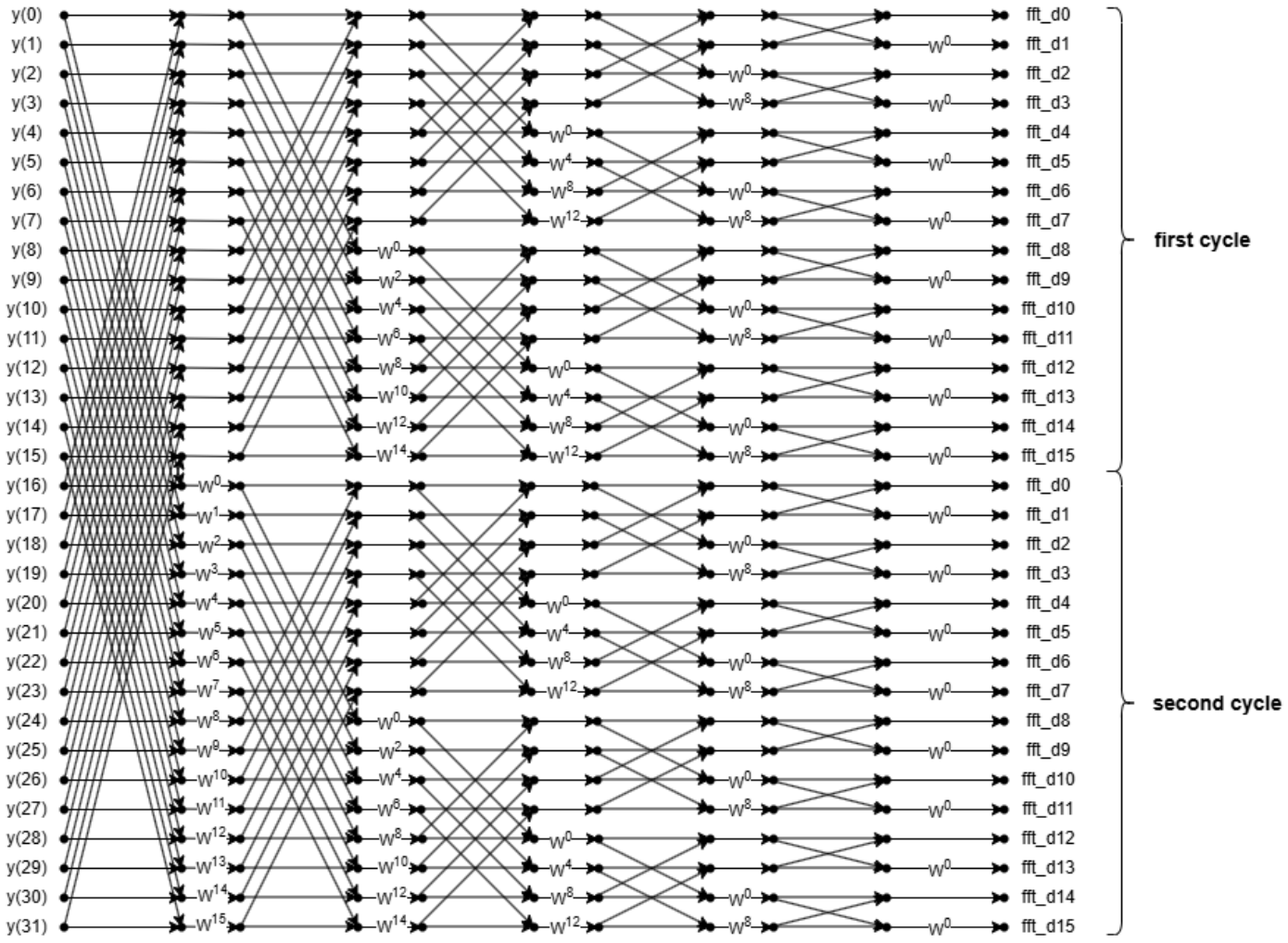


2.2 System I/O Interface

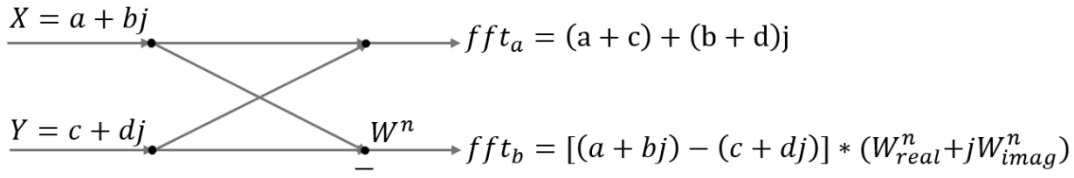
Signal	I/O	Width	Description
clk	Input	1	System clock signal. System should be triggered by the positive edge of clock.
rst	Input	1	System reset signal. Active high, asynchronous reset.
fir_valid	Input	1	When this signal is high, indicates fir_data input is valid.
fir_d	Input	16	16-bit FIR filter data input bus.
fftr_valid	Output	1	The FFT **real part** data valid signal. Should be high for two consecutive cycles each time.
ffti_valid	Output	1	The FFT **imaginary part** data valid signal. Should be high for two consecutive cycles each time.
fft_d0~fft_d15	Output	16	The FFT output data buses.
done	Output	1	When the FFT module completes the task, it sets **done** to **high** to indicate completion.

2.3 Function Description

The Fast Fourier Transform (FFT) used in this assignment requires the implementation of a **32-point** FFT. The hardware architecture is shown in the below. This FFT circuit is designed to convert time-domain signals into frequency-domain signals. Note that the output format this time differs from the homework. **Bit reversal is not required**. Please refer to the diagram below.



2.3.1 Butterfly Unit



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

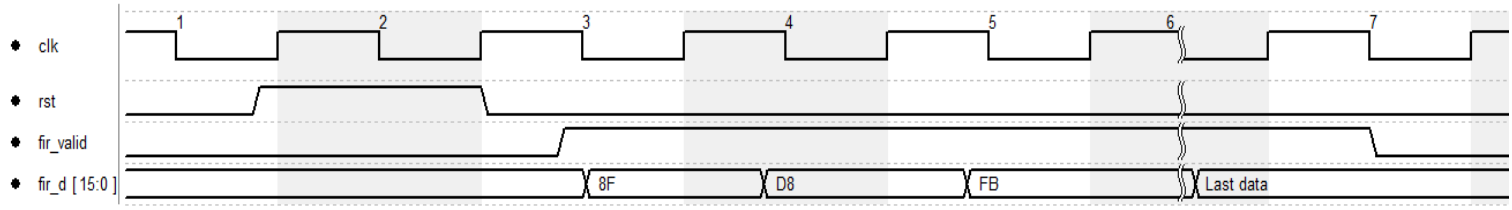
Following is the coefficient table for the 16-point FFT in hexadecimal format

	real part	*imaginary part*		*real part*	*imaginary part*
W_{16}^0	32'h00010000	32'h00000000	W_{16}^8	32'h00000000	32'hFFFF0000
W_{16}^1	32'h0000FB15	32'hFFFFCE0F	W_{16}^9	32'hFFFFCE0F	32'hFFFF04EB
W_{16}^2	32'h0000EC83	32'hFFFF9E09	W_{16}^{10}	32'hFFFF9E09	32'hFFFF137D
W_{16}^3	32'h0000D4DB	32'hFFFF71C6	W_{16}^{11}	32'hFFFF71C6	32'hFFFF2B25
W_{16}^4	32'h0000B504	32'hFFFF4AFC	W_{16}^{12}	32'hFFFF4AFC	32'hFFFF4AFC
W_{16}^5	32'h00008E3A	32'hFFFF2B25	W_{16}^{13}	32'hFFFF2B25	32'hFFFF71C6
W_{16}^6	32'h000061F7	32'hFFFF137D	W_{16}^{14}	32'hFFFF137D	32'hFFFF9E09
W_{16}^7	32'h000031F1	32'hFFFF04EB	W_{16}^{15}	32'hFFFF04EB	32'hFFFFCE0F

W_k^n	Sign bit	integer part	fractional part
	15 bit	1 bit	16 bit

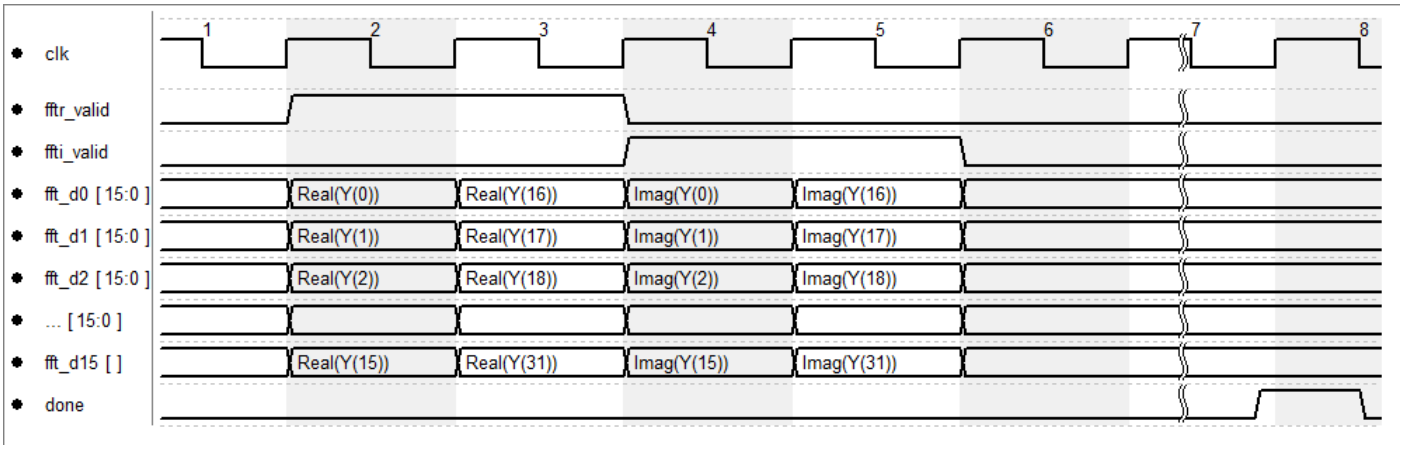
2.3.2 Input and Output Timing Specification

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**.



fir_d	Sign bit	integer part	fractional part
	1 bit	7 bit	8 bit

After the data undergoes FFT processing, if the **fftr_valid or ffti_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 **fftr_valid or ffti_valid** signal must be high for two cycles: during the first cycle, they should be used to transmit data from **index 0 to 15**, and during the second cycle, from **index 16 to 31**. 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

3. File Description

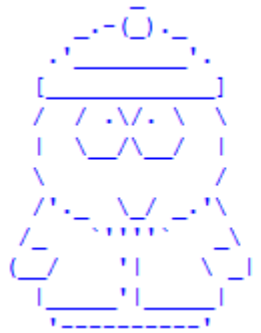
File Name	Description
FFT.v	The module of FFT.
testfixture.sv	Testbench file
GoldenX_FIR.dat	FIR input data
GoldenX_FFT_real.dat	Golden data X for real part of FFT result
GoldenX_FFT_imag.dat	Golden data X for imaginary part of FFT result
GoldenX_BU_real.dat	Golden data X for real part of BU result
GoldenX_BU_imag.dat	Golden data X for imaginary part of BU result
Real_Value_Ref.dat	Real coefficients of FFT
Imag_Value_Ref.dat	Imaginary coefficients of FFT
Output_Ref.dat	Reference values at each stage for the first set of P1

4. Scoring

4.1 Butterfly Unit [40%]/ 32-points FFT [60%]

The scoring is fully based on the functional simulation results in ModelSim. There is no necessary to synthesis the Verilog codes. Please don't design specifically for the test pattern. Otherwise, you will get 0 point.

There are two test patterns in this question, testbench will verify if the outputs are correctly generated. If all the results are correct, you will get the following message in ModelSim simulation.



```
Congratulations! All data have been generated successfully!
Total use 1036 cycles to complete simulation.
```

5. Submission

5.1 Submitted files

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. Please submit your .zip file to folder "Meeting Room X Submission" in Moodle according to your meeting room number X. You can only submit BU.v and FFT.v, so if the design include submodules please copy them into the top module.

StudentID.zip	
StudentID	
BU.v	Verilog code of BU
FFT.v	Verilog code of FFT

5.2 Notes

Deadline: 2025/06/09 12:00.

No late submissions will be accepted, please pay attention to the deadline.