

**E0 217 Project Report (Aug 2023)****Project Title:** Design, Implementation and Simulation of an 8-point FFT Circuit in 45nm CMOS**TABLE OF CONTENTS:**

<b>S.NO</b>	<b>TITLE</b>	<b>PAGE NUMBER</b>
1	Introduction about group members and individual Contributions	2
2	Number of clock cycles for FFT computation	2
3	Area of synthesized design	3
4	Analysis of max clock frequency	3
5	Power Consumption analysis	5
6	Energy Consumption analysis	6
7	Error Computation	6
8	Modification of FFT circuit to compute inverse FFT	8
9	Design Details and explanation of implementation	8
10	Architectural Trade-offs	11
11	References	11

**E0 217 Project Report (Aug 2023)****Names and IISc Email IDs of Group Members:**

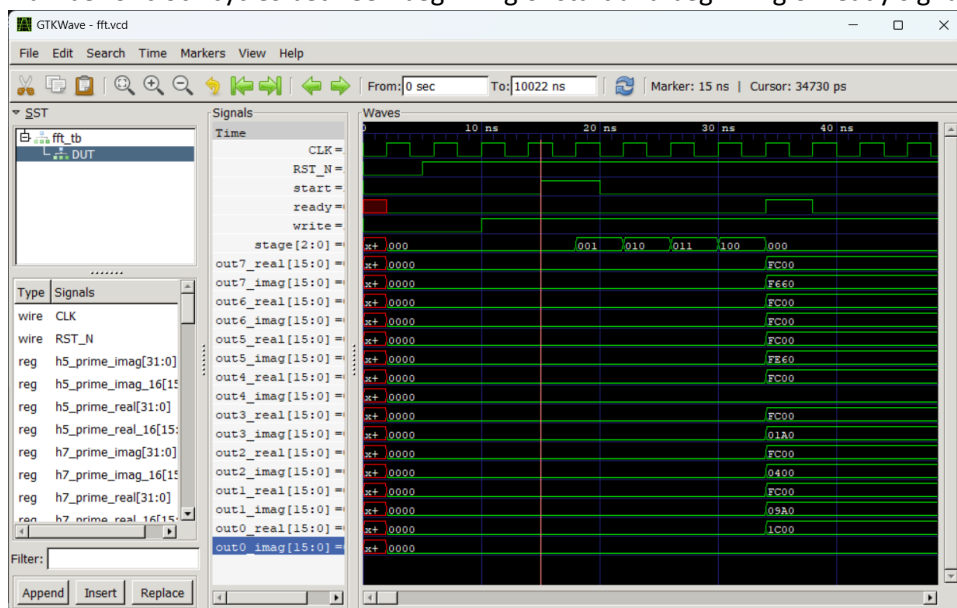
	Name	SR No.	IISc Email ID
1.	Guhan Rajasekar	22410	guhanr@iisc.ac.in
2.	Ujjwal Chaudhary	22577	ujjwalc@iisc.ac.in

**Individual Contributions**

	Name	Contributions
1.	Guhan Rajasekar	<ul style="list-style-type: none"> <li>• Implementation of FFT code (without control signals)</li> <li>• Error computation between obtained and desired results</li> <li>• Report compilation</li> </ul>
2.	Ujjwal Chaudhary	<ul style="list-style-type: none"> <li>• Optimization of code (changing combinational blocks to sequential blocks) and making the code compatible with CLK signal and control signals like RST, WRITE, START, READY</li> <li>• Creation of test bench and simulation of design in GTK Wave.</li> <li>• Performed area, power and speed analysis.</li> </ul>

**Number of clock cycles per FFT computation:**

- Number of clock cycles between beginning of start and beginning of ready signal = **4.75**



Area of synthesized design:

```

ujjwal@ujjwal: ~/Desktop/ESDCS/ESDCS_project
19. Printing statistics.
=== fft ===
Number of wires:          9468
Number of wire bits:      9468
Number of public wires:   1736
Number of public wire bits: 1736
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          7732
AND2_X1                   350
AND3_X1                   42
AOI211_X1                 88
AOI21_X1                 990
AOI221_X1                 1
AOI22_X1                 5
DFF_X1                   1156
INV_X1                   268
MUX2_X1                   736
NAND2_X1                 717
NAND3_X1                 196
NAND4_X1                 8
NOR2_X1                  515
NOR3_X1                  48
NOR4_X1                   4
OAI211_X1                 7
OAI21_X1                 544
OAI22_X1                 130
OR2_X1                   108
OR3_X1                   67
OR4_X1                   2
XNOR2_X1                 873
XOR2_X1                  877

Chip area for module 'fft': 13367.298000

```

- Area of synthesized design is **13367.298  $\mu\text{m}^2$** .

Max clock frequency supported by the synthesized design:

- Typical corner, 25°C and 1V

```

ujjwal@ujjwal: ~/Desktop/ESDCS/ESDCS_project
OpenSTA> report_checks
Startpoint: _14567_ (rising edge-triggered flip-flop clocked by CLK)
Endpoint: _15401_ (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: max

Delay    Time    Description
-----
0.00     0.00    clock CLK (rise edge)
0.00     0.00    clock network delay (ideal)
0.00     0.00    ^ _14567_/CK (DFF_X1)
0.09     0.09    v _14567_/Q (DFF_X1)
0.05     0.14    ^ _07894_/ZN (NOR2_X1)
0.03     0.17    v _07895_/ZN (NAND2_X1)
4.32     4.49    ^ _07896_/ZN (NOR2_X1)
0.04     4.53    v _08158_/ZN (OAI21_X1)
0.17     4.69    ^ _08159_/ZN (OAI22_X1)
0.00     4.69    ^ _15401_/D (DFF_X1)
0.00     4.69    data arrival time

5.00     5.00    clock CLK (rise edge)
0.00     5.00    clock network delay (ideal)
0.00     5.00    clock reconvergence pessimism
5.00     5.00    ^ _15401_/CK (DFF_X1)
-0.05    4.95    library setup time
4.95     4.95    data required time

4.95     4.95    data required time
-4.69    -4.69    data arrival time

0.26     0.26    slack (MET)

```

- OpenSTA> S
- This result is for typical corner, 25°C and 1V.
- Input clock signal period is 5ns.
- Here the slack period is 0.26ns.
- Supported clock period is 5ns – 0.26ns = **4.74ns**
- Operating frequency is **210.97 MHz**.

- **SS corner, 25°C, 1V**

```

ujjwal@ujjwal: ~/Desktop/ES...
create_clock -name CLK -period 20 {CLK}

set_power_activity -input -activity 0.5

set_power_activity -global -activity 0.5
OpenSTA> report_checks
Startpoint: _14567_ (rising edge-triggered flip-flop clocked by CLK)
Endpoint: _15401_ (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: max

Delay    Time    Description
-----
0.00     0.00    clock CLK (rise edge)
0.00     0.00    clock network delay (ideal)
0.00     0.00    ^ _14567_/CK (DFF_X1)
0.26     0.26    v _14567_/Q (DFF_X1)
0.20     0.46    ^ _07894_/ZN (NOR2_X1)
0.09     0.55    v _07895_/ZN (NAND2_X1)
16.41    16.96    ^ _07896_/ZN (NOR2_X1)
1.36     18.32    v _08158_/ZN (OAI21_X1)
0.85     19.17    ^ _08159_/ZN (OAI22_X1)
0.00     19.17    ^ _15401_/D (DFF_X1)
         19.17    data arrival time

20.00    20.00    clock CLK (rise edge)
0.00     20.00    clock network delay (ideal)
0.00     20.00    clock reconvergence pessimism
         20.00    ^ _15401_/CK (DFF_X1)
-0.16    19.84    library setup time
         19.84    data required time

         19.84    data required time
        -19.17    data arrival time

         0.66    slack (MET)

```

- This is for SS corner, 25°C and 1V.
- Input clock period is 20ns. (because this case does not support clock signal of 5ns)
- Slack period is 0.66 ns.
- Supported clock signal period is 20ns – 0.66ns = **19.34ns**
- Operating frequency = **51.706MHz**.

- **FF corner, 25°C, 1V**

```

ujjwal@ujjwal: ~/Desktop/ESDCS/E...
link_design fft

create_clock -name CLK -period 5 {CLK}

set_power_activity -input -activity 0.5

set_power_activity -global -activity 0.5
OpenSTA> report_checks
Startpoint: _14567_ (rising edge-triggered flip-flop clocked by CLK)
Endpoint: _15210_ (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: max

Delay    Time    Description
-----
0.00     0.00    clock CLK (rise edge)
0.00     0.00    clock network delay (ideal)
0.00     0.00    ^ _14567_/CK (DFF_X1)
0.06     0.06    v _14567_/Q (DFF_X1)
0.03     0.08    ^ _07894_/ZN (NOR2_X1)
0.01     0.10    v _07895_/ZN (NAND2_X1)
2.23     2.33    ^ _07896_/ZN (NOR2_X1)
-0.01     2.32    v _09189_/ZN (NAND2_X1)
0.11     2.43    ^ _09190_/ZN (OAI22_X1)
0.00     2.43    ^ _15210_/D (DFF_X1)
         2.43    data arrival time

5.00     5.00    clock CLK (rise edge)
0.00     5.00    clock network delay (ideal)
0.00     5.00    clock reconvergence pessimism
         5.00    ^ _15210_/CK (DFF_X1)
-0.03     4.97    library setup time
         4.97    data required time

         4.97    data required time
         2.43    data arrival time

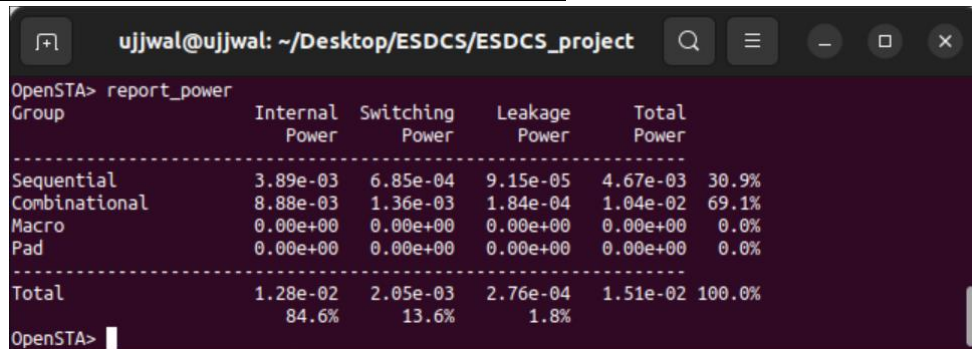
         2.54    slack (MET)

```

- Input clock signal period = 5ns.
- Slack period = 2.54 ns.
- Clock period supported by the design = **2.46ns**.
- Max operating frequency at FF corner = **406.504MHz**

**Power Consumption of the synthesized design:**

- **Power Consumption in typical corner, 25°C and 1V:**



```

ujjwal@ujjwal: ~/Desktop/ESDCS/ESDCS_project
OpenSTA> report_power

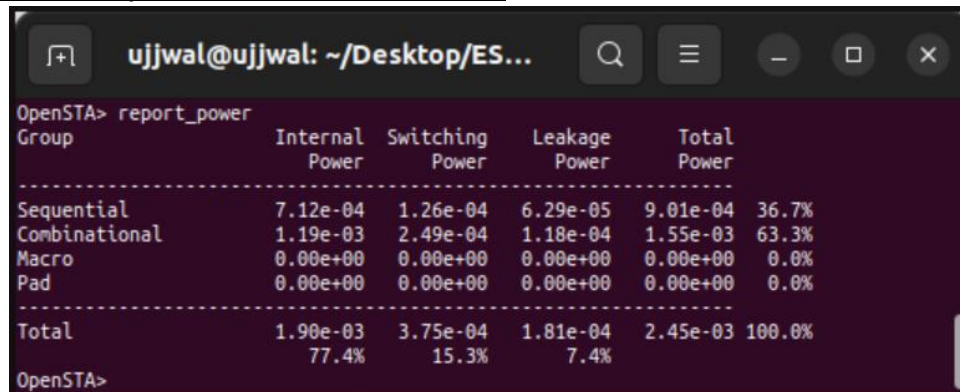
```

Group	Internal Power	Switching Power	Leakage Power	Total Power	
Sequential	3.89e-03	6.85e-04	9.15e-05	4.67e-03	30.9%
Combinational	8.88e-03	1.36e-03	1.84e-04	1.04e-02	69.1%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
<b>Total</b>	<b>1.28e-02</b>	<b>2.05e-03</b>	<b>2.76e-04</b>	<b>1.51e-02</b>	<b>100.0%</b>
	84.6%	13.6%	1.8%		

OpenSTA>

- 
- The total power consumed at typical corner, 25°C and 1V is **15.1mW**.

- **Power Consumption in SS corner, 25°C and 1V.**



```

ujjwal@ujjwal: ~/Desktop/ES...
OpenSTA> report_power

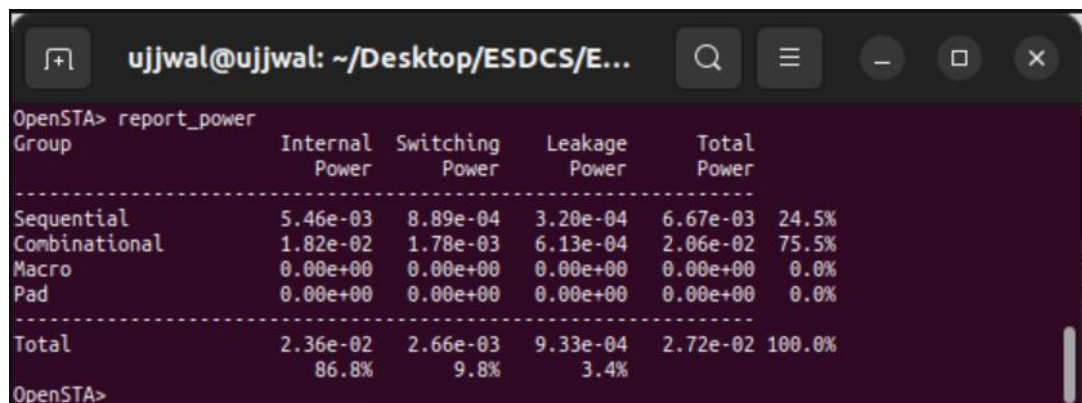
```

Group	Internal Power	Switching Power	Leakage Power	Total Power	
Sequential	7.12e-04	1.26e-04	6.29e-05	9.01e-04	36.7%
Combinational	1.19e-03	2.49e-04	1.18e-04	1.55e-03	63.3%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
<b>Total</b>	<b>1.90e-03</b>	<b>3.75e-04</b>	<b>1.81e-04</b>	<b>2.45e-03</b>	<b>100.0%</b>
	77.4%	15.3%	7.4%		

OpenSTA>

- 
- Power consumed at SS corner, 25°C and 1V is **2.45mW**

- **Power Consumption in FF corner, 25°C and 1V.**



```

ujjwal@ujjwal: ~/Desktop/ESDCS/E...
OpenSTA> report_power

```

Group	Internal Power	Switching Power	Leakage Power	Total Power	
Sequential	5.46e-03	8.89e-04	3.20e-04	6.67e-03	24.5%
Combinational	1.82e-02	1.78e-03	6.13e-04	2.06e-02	75.5%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
<b>Total</b>	<b>2.36e-02</b>	<b>2.66e-03</b>	<b>9.33e-04</b>	<b>2.72e-02</b>	<b>100.0%</b>
	86.8%	9.8%	3.4%		

OpenSTA>

- 
- Total power consumption at FF corner, 25°C and 1V is **27.2mW**.

**Energy Consumption of synthesized design:**

- Typical, 25°C, 1V, T=5ns (Image attached above)
- Typical, 25°C, 1V, T=10ns

```
ujjwal@ujjwal: ~/Desktop/ESDCS/ESDCS_project
OpenSTA> create_clock -name CLK -period 10 {CLK}
OpenSTA> report_power
```

Group	Internal Power	Switching Power	Leakage Power	Total Power	
Sequential	1.95e-03	3.43e-04	9.15e-05	2.38e-03	31.0%
Combinational	4.44e-03	6.81e-04	1.84e-04	5.31e-03	69.0%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	6.39e-03	1.02e-03	2.76e-04	7.68e-03	100.0%
	83.1%	13.3%	3.6%		

- Typical, 25°C, 1V, T=20ns

```
ujjwal@ujjwal: ~/Desktop/ESDCS/ESDCS_project
OpenSTA> create_clock -name CLK -period 20 {CLK}
OpenSTA> report_power
```

Group	Internal Power	Switching Power	Leakage Power	Total Power	
Sequential	9.73e-04	1.71e-04	9.15e-05	1.24e-03	31.0%
Combinational	2.22e-03	3.40e-04	1.84e-04	2.74e-03	69.0%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Total	3.19e-03	5.12e-04	2.76e-04	3.98e-03	100.0%
	80.2%	12.9%	6.9%		

Corner Case	Power Consumption (mW)	Frequency (MHz)	Energy Consumption (pJ)
Typical, 25°C, 1V	15.1	200	75.5
Typical, 25°C, 1V	7.68	100	76.8
Typical, 25°C, 1V	3.98	50	79.6

**Dependence of Energy Consumption of the synthesized design on the clock frequency:**

- From above mentioned table, we conclude that **energy consumption reduces as clock frequency increases.**

**Maximum absolute error when computing FFT of {0, 1, 2, 3, 4, 5, 6, 7}**

- Let the 8-point FFT result be denoted as X(0), X(1), X(2), X(3), X(4), X(5), X(6) and X(7)
- Here a **decimal point has been added in the obtained results to indicate there are 8 bits for the fractional part, 7 bits for the decimal part and MSB is the sign bit.** This holds true for both real and imaginary parts.
- In the following calculations,  $\sqrt{2}$  is considered as **1.414**.

**1. X(0) :**

- Expected =  $28 + j0$
- Obtained =  $(1c.00 + j00.00)_H = 28 + j0$
- Absolute Error = 0

**2. X(1)**

- Expected =  $-4.0 + j 9.656$
- Obtained =  $(fc.00 + j09.a0)_H = -4 + j9.625$
- Absolute Error =  $|(-4.0 + j 9.656) - (-4 + j9.625)| = 0.031$

**3. X(2)**

- Expected =  $-4 + j4$
- Obtained =  $(fc.00 + j04.00)_H = (-4 + j4)$
- Absolute Error = 0

**4. X(3)**

- Expected =  $-4 + j1.656$
- Obtained =  $(fc.00 + j01.a0)_H = -4 + j1.625$
- Absolute Error =  $|(-4 + j1.656) - (-4 + j1.625)| = 0.031$

**5. X(4)**

- Expected =  $-4 + j0$
- Obtained =  $(fc.00 + j00.00)_H = -4 + j0$
- Absolute Error = 0

**6. X(5)**

- Expected =  $-4 - j1.656$
- Obtained =  $(fc.00 + jfe.60)_H = (-4 - j1.402)$
- Absolute Error =  $|(-4 - j1.656) - (-4 - j1.402)| = 0.254$

**7. X(6)**

- Expected =  $-4.0 - j4.0$
- Obtained =  $(fc.00 + jfc.00)_H = -4 - j4$
- Absolute Error = 0

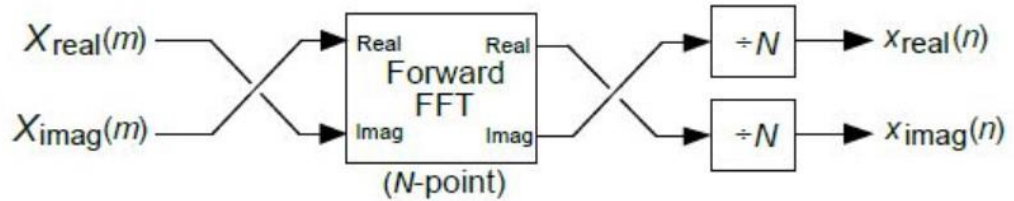
**8. X(7)**

- Expected =  $-4 - j9.656$
- Obtained =  $(fc.00 + jf6.60)_H = -4 - j9.625$
- Absolute Error =  $|(-4 - j9.656) - (-4 - j9.625)| = 0.031$



### Modifying the fft circuit to compute inverse fft

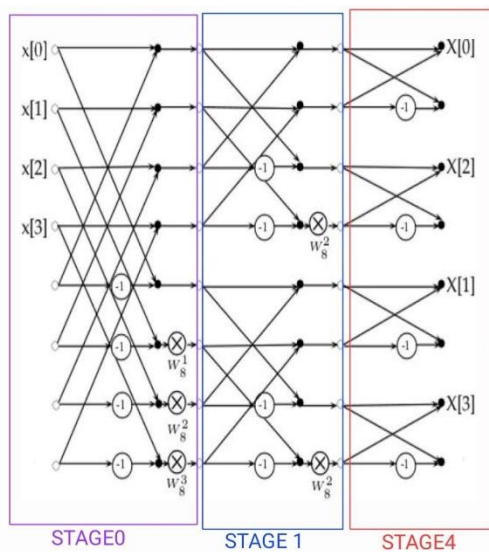
- Inverse fft can be computed using Data Swapping technique, as shown below:



- 
- The real part of FFT input is fed as imaginary input to the circuit and the imaginary part of FFT is fed as real input to the circuit.
- The real portion of the output gives imaginary part of expected time domain sequence scaled by N and the imaginary part of the output gives the real part of the time domain sequence scaled by N. To get the desired outputs, we divide them by N.
- So, in addition to the existing hardware circuitry, extra circuitry will be required to scale by factor of (1/N) to get the desired time domain inverse FFT sequence.

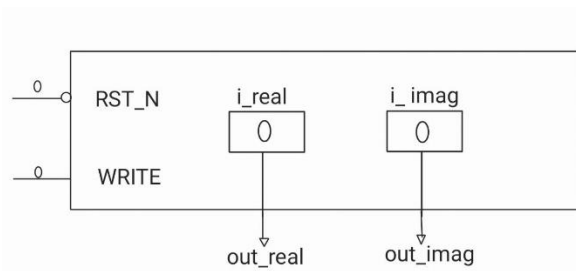
### Design Details:

- The entire FFT computation has been divided into five stages as follows:

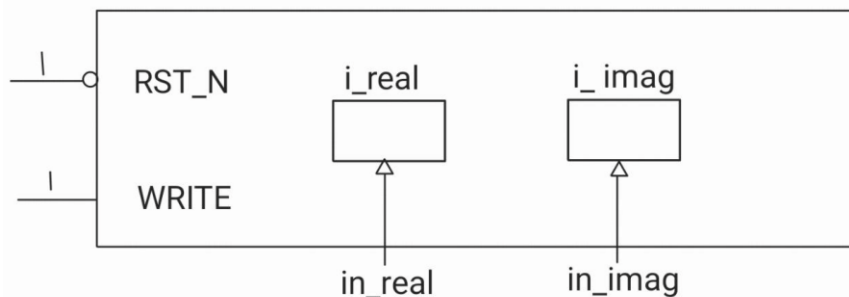


- 
- Stage 2 and Stage 3 are used for intermediate variable computations in the code.
- But the overall FFT computation can be broadly classified into stages 0, 1 and 4.
- If the RST\_N signal is low, the value 0 is written in the registers i0\_real, i0\_imag, i1\_real, i1\_imag, ..... , i7\_real, i7\_imag.
- The output wires out0\_real, out0\_imag, out1\_real, out1\_imag, ..... out7\_real, out7\_imag are continuously driven by the data present in the registers as shown below.

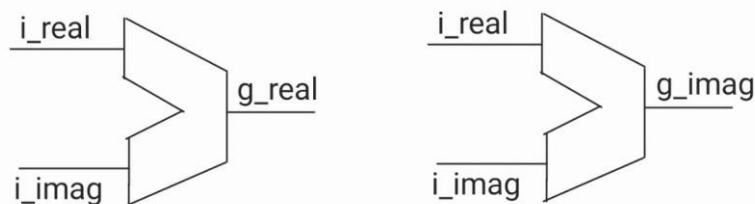




- 
- In the above block, although only two registers are shown, there are total of 16 registers ( 8 for the real part and 8 for the imaginary part ) and each register holds 16 bit data.
- **Storing Inputs in registers:**
  - The circuit stores inputs in registers when **write** signal is high.
  - When write signal is high, the registers **i0\_real**, **i0\_imag**, **i1\_real**, **i1\_imag**, ..., **i7\_real**, **i7\_imag** will store the value held by the continuously driven wires **in0\_real**, **in0\_imag**, **in1\_real**, **in1\_imag**, ..., **in7\_real**, **in7\_imag** respectively as shown below:



- 
- Here as well only two signals **in\_real** and **in\_imag** are shown for brevity of block diagram.
- But there are 16 wires ( 8 for real and 8 for imaginary parts) that are continuously driven.
- Computation commences when **start** signal is high.
- The value stored by 2-bit register variable named **stage** indicates the stage of FFT computation.
- **Stage 0 computations:**
  - The contents in **i0\_real**, **i0\_imag**, ..., **i7\_real**, **i7\_imag** act as inputs for stage 0 computations.
  - The registers **g\_real[0]**, **g\_imag[0]**, **g\_real[1]**, **g\_imag[1]**, ..., **g\_real[7]**, **g\_imag[7]** store the results of stage 0.

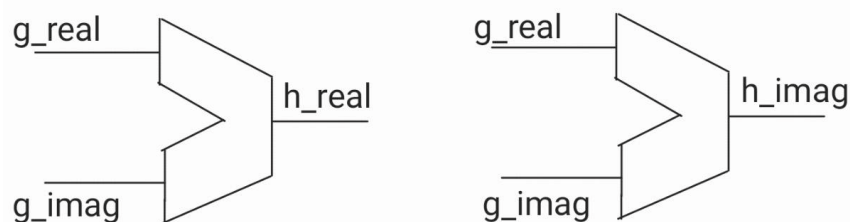


➤

- Once again for brevity, not all the registers of stage 0 are shown.
- There are 16 registers. 8 registers hold the real part of stage 0 results and 8 registers hold the imaginary part of stage 0 results .
- Stage 0 computations is based on the formula obtained from the butterfly diagram.
- Once stage 0 computations are done, the content of **stage register** is incremented by 1 to go to the next stage.

- **Stage 1 computations**

- For stage 1 computations, the contents of the registers  $g\_real[0]$ ,  $g\_imag[0]$ ,  $g\_real[1]$ ,  $g\_imag[1]$ , ...,  $g\_real[7]$ ,  $g\_imag[7]$  will be the inputs and the results are stored in the registers



- 
- The computations are based on the equations obtained from the butterfly diagram.

- **Stage 2 computations**

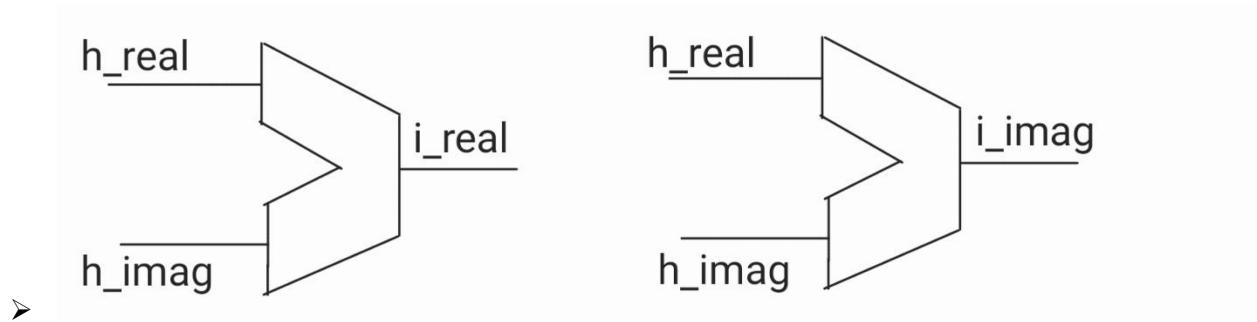
- In this stage , the registers temp0, temp1 ,temp2 and temp3 are populated as follows:
  - $temp0 = h\_real[5] + h\_imag[5]$
  - $temp1 = h\_imag[5] - h\_real[5]$
  - $temp2 = h\_imag[7] - h\_real[7]$
  - $temp3 = -(h\_imag[7] + h\_real[7])$
- These calculations aid in the computation of the final result.
- Once the above calculations are done, contents of the stage register is incremented by 1 and we go to the next stage.

- **Stage 3 computations**

- In this stage, the contents of temp0, temp1, temp2, temp3 are multiplied with 0.707 ( hardcoded in binary in the code) .
- Then content of the stage register is incremented and we go to the final stage.

- **Stage 4 computations:**

- Here the contents of  $h\_real[0]$ ,  $h\_imag[0]$ ,  $h\_real[1]$ ,  $h\_imag[1]$ , ...,  $h\_real[7]$ ,  $h\_imag[7]$  act as inputs and the final output is stored in the registers  $i\_real[0]$ ,  $i\_imag[0]$ ,  $i\_real[1]$ ,  $i\_imag[1]$ , ...,  $i\_real[7]$ ,  $i\_imag[7]$ .



- The wires out0\_real , out0\_imag, out1\_real, out1\_imag,.....out7\_real,out7\_imag are continuously driven by the contents in these registers. Hence at any point of time, the output at that instant is present in these output wires.

### Architectural Trade-offs

- In this FFT implementation, focus is low latency.
- Hence the inbuilt multiplication function has been used with the ' \* ' symbol to get faster multiplication results that are required at the intermediate stages.
- This comes at the cost of more area and power consumption.
- As an addition, pipelining can be implemented if we want to improve throughput.

### References

- [1] Wikipedia – Fast Fourier Transform
- [2] NPTEL – Hardware Modelling using Verilog by Prof. Indranil Sen Gupta
- [3] “The Fast Fourier Transform (FFT): Most Ingenious Algorithm Ever?” Youtube.com by Reducible
- [4] “DSP Tricks: Computing inverse FFTs using the forward FFT” – embedded.com