8 Point FFT Hardware Accelerator

Sanidhya Saxena, DESE, Indian Institute of Science, Bangalore-560012

I Introduction

The Fast Fourier Transform (FFT) is an efficient algorithm to compute the Discrete Fourier Transform (DFT) and its inverse. It plays a crucial role in various digital signal processing (DSP) applications such as image processing, telecommunications, and audio signal analysis. While software implementations of FFT are common, hardware accelerators are often employed in applications requiring real-time processing and high throughput.

II Specification

- Each input element x_m (for m=0,1,...,7) is a complex number with real and imaginary parts each represented as a signed 16-bit fixed point quantity with 1 sign bit, 7 bits for decimal part and 8 bits for fractional part.
- Each output element X_k (for k=0,1,...,7) is a complex number with real and imaginary parts each represented as a signed 16-bit fixed-point quantity with 1 sign bit, 7 bits for decimal part and 8 bits for fractional part.

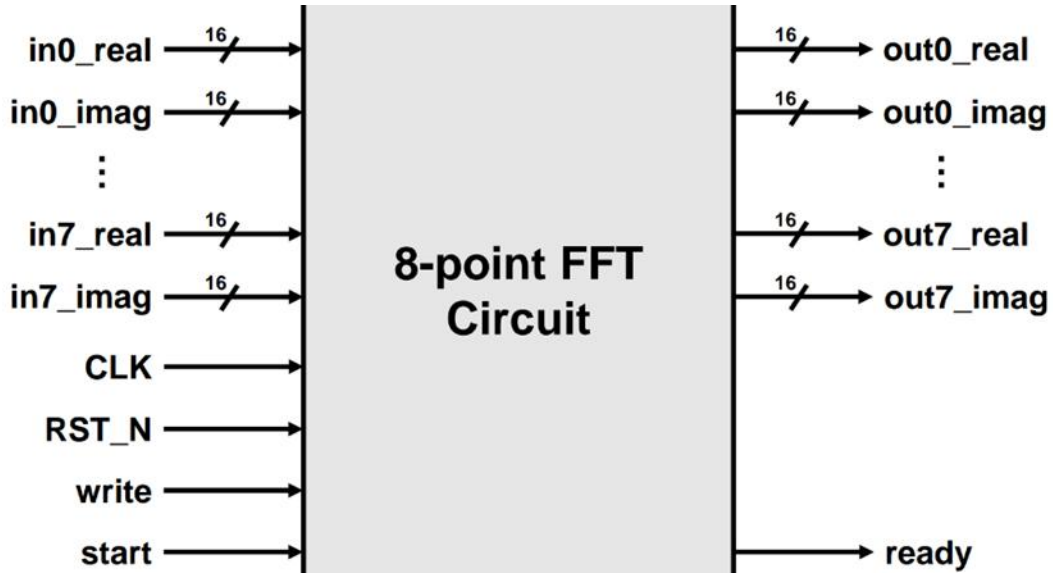

Figure 1: 8-point FFT Top level IP

III Architecture

The architecture of an 8-point FFT hardware accelerator is designed to efficiently perform the Fast Fourier Transform on a sequence of 8 complex input samples. The architecture typically includes several key components that work together to achieve high-speed, low-latency, and power-efficient FFT computation.

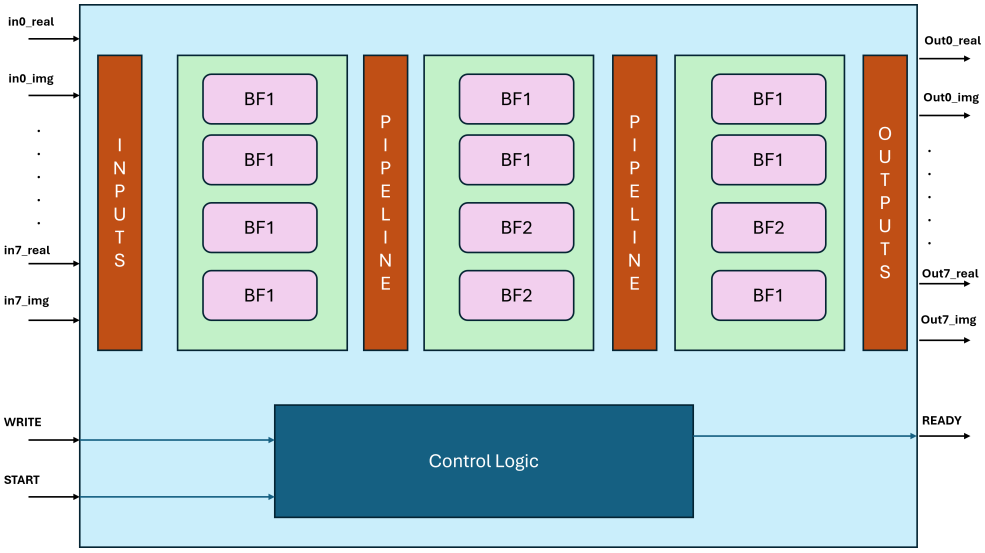

Figure 2: 8 point FFT Microarchitecture

This is the high-level micro architecture of the FFT block. It consists of FFT stages which is where all the combinational logic related to computation is there. We have a control block for generating READY signal when we are done with the computation. Active High reset strategy is used for out design. The designed in throughout pipelined for a higher throughput and for high operating clock frequencies.

Each of the FFT stage (1,2,3) comprises of the basic butterfly block which does the addition and subtraction of the 2 complex numbers. 8-point DFT is broken into 4 point and then each 4 point is broken into 2 points.

Parallel topology is used for computation so as to reduce the critical path and stall time of the system. Combinational loops are avoided as much as possible and each block computation is made parallel.

3.1 Butterfly Processing Unit

The core of the 8-point FFT hardware accelerator is the Butterfly Processing Units. These units are responsible for executing the fundamental butterfly operations, which are the basic building blocks of the FFT algorithm. Each butterfly operation involves a pair of complex multiply-accumulate (MAC) operations, which combine two input values to produce two output values.

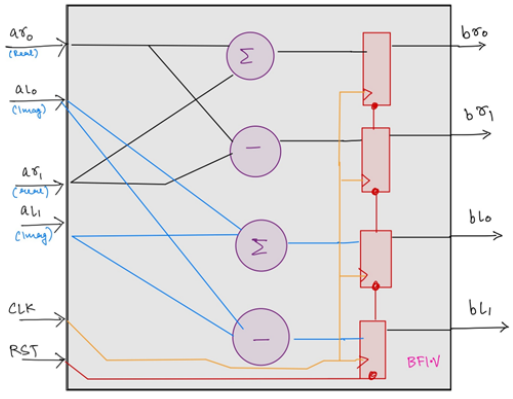

Figure 3: BF1.v

This is a basic structure of the butterfly block used in side the FFT_stage block. This is just a simple addition and subtraction block operated on 2 complex number.

It computes:$(Ar_0 + jAi_0) + (Ar_1 + jAi_1) = (Br_0 + jBi_0)$ and output is generated after one-clock cycle (Pipelined).

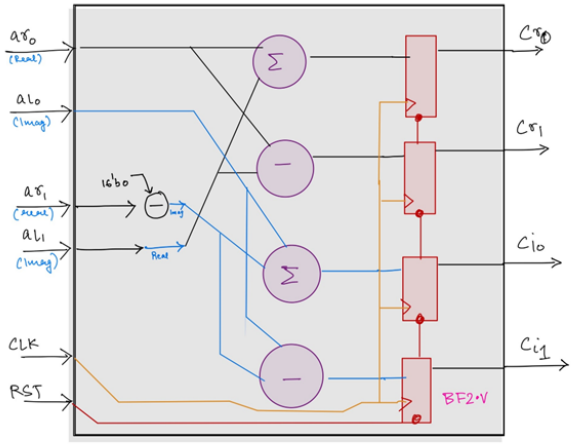

Figure 4: BF2.v

This block works same as the BF1 block, with an additional functionality of j multiplication. Any Complex number multiplied with -j, is simply just interchanging the real and imaginary part and adding a negative sign to the real part. Hence the multiplication ofW_8^2 and W_8^6 is done using this block only. In our twiddle factor is not stored in any memory, on the fly computation is done.

In an 8-point FFT, there are three stages of butterfly operations, with each stage involving a different level of computation:

- Stage 1: Has 4 instances of BF1.v, since only addition and subtraction of input is required in this stage in bit reversal form.
- Stage 2: Has 2 instances of BF1.v, and 2 instances of BF2.v.
- Stage 3: Has 3 instances of BF1.v and 1 instance of BF2.v

Output of the BF2.v in the stage2 is fed to the Constant multiplier block for multiplying it with $W_8^1, W_8^3, W_8^5, W_8^7$

3.2 Twiddle Factor Multiplication Unit

Twiddle factors are complex exponential coefficients used in the FFT computation. The twiddle factor multiplication unit is responsible for multiplying the intermediate results from the butterfly units by the appropriate twiddle factors. This unit typically consists of complex multipliers that are optimized for high-speed and low-power operation.

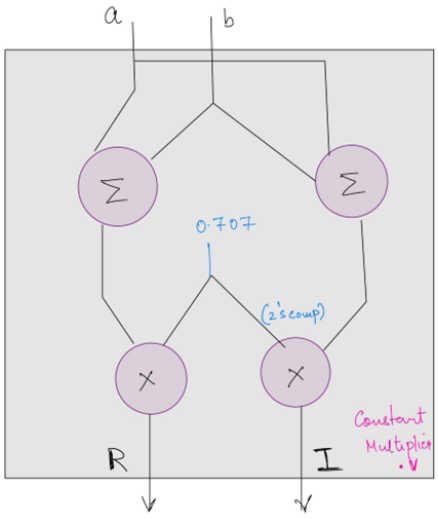

Figure 5: Complex_Multiplier.v

This is the complex multiplication block for multiplying the complex number. Multiplication of 2 complex number could be expressed as

$$(a + jb)(0.707 - j0.707) = 0.707(a + b) - j0.707(a - b) \tag{1}$$

$$(a + jb)(-0.707 - j0.707) = -0.707(a - b) - j0.707(a + b) \tag{2}$$

Hence in order to reduce the number of multiplications we created a constant multiplier block. Here we just generate (a+b) and (a-b) and multiply the result with the 0.707 in order to get the resultant complex number.

1 is solved in **complex_mult.v** and 2 is solved in **complex_mult1.v.**

3.3 Control Unit

The control unit orchestrates the entire FFT computation process. It manages the sequencing of butterfly operations, twiddle factor multiplication, and data reordering. The control unit also ensures that the data flows smoothly through the various stages of the accelerator, coordinating the timing and synchronization of each operation.

This control logic is implemented at the top only. With WRITE signal we initialise our inputs present on the IO pads or from TB in this case.

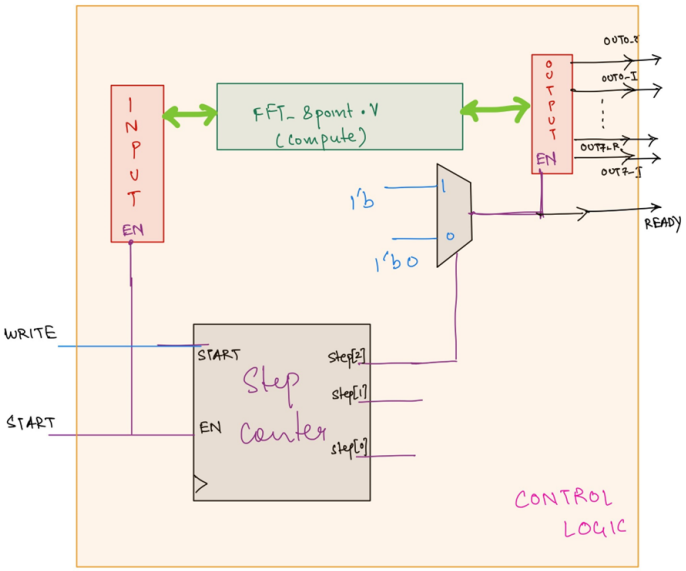

Figure 6: Control.v

Here we have the step counter which measures the number of clock cycle used for the computation of the 8 Point FFT. In our design FFT computation is done in 5 clock cycles.

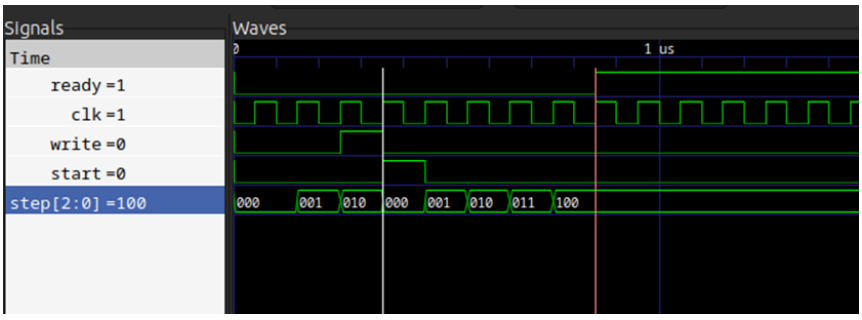

Figure 7: Control Logic Waveforms

When the counter reaches 3'b100, the ready signal goes high in next clock cycle indicating that the FFT computation is done and output can be loaded on to the bus. Start signal, starts the computation of step signal using the counter.

This is how the computation of the 8 point FFT is done within 5 clock cycles with an operating speed of 609Mhz @ 1v and 25°C

IV Simulation Waveforms

All simulations are done on GTK Wave

Figure 8: Inputs

Inputs are loaded at start write signal and computation is started at START signal. It takes 5 clock cycle to compute the FFT, and output is stored when READY signal is high.

Figure 9: Outputs

V Performance Matrix

5.1 Testbench Output

```
root@santanya-VirtualBox:/home/santanya/ESDCS/ESDCS_Projec
VCD info: dumpfile FFT_tb.vcd opened for output.
START OF SIMULATION (Time: 150 ns)
X0:  28.000000+j0.000000
X1:  -4.000000+j9.625000
X2:  -4.000000+j4.000000
X3:  -4.000000+j1.656250
X4:  -4.000000+j0.000000
X5:  -4.000000+j-1.625000
X6:  -4.000000+j-4.000000
X7:  -4.000000+j-9.656250
```

Figure 10: Testbench Output

5.2 Area

```
Number of wire bits:          20733
Number of public wires:       11396
Number of public wire bits:   11396
Number of memories:               0
Number of memory bits:            0
Number of processes:              0
Number of cells:              10280
  AND2_X1                       404
  AND3_X1                       260
  AND4_X1                         1
  AOI211_X1                      14
  AOI21_X1                      651
  AOI221_X1                       7
  AOI22_X1                        6
  DFF_X1                       1088
  INV_X1                       1650
  MUX2_X1                       345
  NAND2_X1                      866
  NAND3_X1                      105
  NAND4_X1                        2
  NOR2_X1                      1766
  NOR3_X1                        20
  NOR4_X1                        12
  OAI211_X1                      78
  OAI21_X1                      628
  OAI221_X1                      27
  OAI222_X1                       1
  OAI22_X1                       19
  OAI33_X1                        4
  OR2_X1                        158
  OR3_X1                        191
  XNOR2_X1                     1214
  XOR2_X1                       763

Chip area for top module '\FFT_8point': 14627.074000
```

Figure 11: Chip Area

Area: 14267.07um^2

5.3 Operating Frequency

‘

```
   0.02    1.49 v stg3/m4/_345_/ZN (AOI21_X1)
   0.05    1.54 ^ stg3/m4/_348_/ZN (AOI21_X1)
   0.04    1.58 ^ stg3/m4/_350_/ZN (XNOR2_X1)
   0.01    1.59 v stg3/m4/reg1/d16/_3_/ZN (INV_X1)
   0.02    1.61 ^ stg3/m4/reg1/d16/_4_/ZN (NOR2_X1)
   0.00    1.61 ^ stg3/m4/reg1/d16/_5_/D (DFF_X1)
           1.61   data arrival time

   1.64    1.64   clock CLK (rise edge)
   0.00    1.64   clock network delay (ideal)
   0.00    1.64   clock reconvergence pessimism
           1.64 ^ stg3/m4/reg1/d16/_5_/CK (DFF_X1)
  -0.03    1.61   library setup time
           1.61   data required time
---------------------------------------------------------
           1.61   data required time
          -1.61   data arrival time
---------------------------------------------------------
           0.00   slack (VIOLATED)


OpenSTA> create_clock -name CLK -period 1.64 {clk}
```

Figure 12: Operating Frequency

Operating Frequency: 609.75Mhz

5.4 Average Power

```
OpenSTA> create_clock -name CLK -period 1.64 {clk}
OpenSTA> set_power_activity -input -activity 0.5
OpenSTA> set_power_activity -global -activity 0.5
OpenSTA> report_power
Group                  Internal  Switching    Leakage      Total
                          Power      Power      Power      Power
----------------------------------------------------------------
Sequential             1.10e-02   1.33e-03   8.61e-05   1.24e-02  37.1%
Combinational          1.64e-02   4.51e-03   2.25e-04   2.11e-02  62.9%
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                  2.74e-02   5.84e-03   3.11e-04   3.36e-02 100.0%
                          81.7%      17.4%       0.9%
```

Figure 13: Average Power

AVERAGE POWER: 33.6mW

VI Conclusion

The architecture of an 8-point FFT hardware accelerator is designed to balance speed, efficiency, and power consumption. By integrating key components like Butterfly Processing Units, Twiddle Factor Multiplication Units, a Data Reordering Module, a Control Unit, and an optimized Input/Output Interface, the accelerator efficiently performs the FFT computation. The architecture's pipelined data flow further enhances performance, making it suitable for real-time DSP applications.