Synthesizable 2-Clock Verilog FIR Filter

Isha Garg
Electrical Engineering
Columbia University
New York, USA
ig2451@columbia.edu

Mor Shimshi
Electrical Engineering
Columbia University
New York, USA
ms6350@columbia.edu

Yunze Yang
Electrical Engineering
Columbia University
New York, USA
yy3181@columbia.ed

***Abstract*—This report presents a hardware design of a synthesizable 2-Clock 64-tap 16-bit FIR Filter in Verilog.**

I. Introduction

Finite Impulse Response (FIR) Filters are digital filters used in Digital Signal Processing. An FIR filter produces a weighted sum of the current and past inputs. It is implemented using the following equation:

$$y[n] = \sum_{i=0}^{N} b_i * x[n-i] \tag{1}$$

where x[n] is the input signal, y[n] is the output signal and b_i is the filter coefficient. This is also called discrete convolution. We learned the details of FIR filter calculations from [1].

II. Design

A. ALU

The ALU performs signed multiplication and addition. It takes two 16-bit signed numbers (current data point and coefficient) and one 41-bit signed number (accumulator) as inputs. The current data and the coefficient are are multiplied together, and the product is added with the accumulator. The output of the ALU is a 41-bit signed number.

A 41-bit signed number is used for the accumulator to remove any possibility of overflow in the non-fractional component of the fixed point sum. 41 bits is used as an 8, 8 signed fixed point number multiplied by another 8,8 fixed point number results in a 17, 16 number. The repeated addition of 64 17, 16 numbers results in a 25, 16 number.

B. FIFO

The asynchronous FIFO uses two clocks, one for writing (10kHz), and one for reading (2MHz), and has a depth of 64. The FIFO is used in place of the data memory. At the rising edge of the write clock, the data from outside the core is shifted into the FIFO. At each rising edge of the read clock, the data at the requested index is returned. Thus, in order to properly operate, the FIFO read clock must be at least 64 times faster than the write clock, as if the clock were not faster data would be overwritten before it is read.

Internally, the FIFO consists of 64 16-bit registers that are connected in a shift register configuration. On each rising edge of the write clock, new data is shifted into the first register and all data is shifted down one position. This means that the newest data is always retrieved by the core first (index 0).

We got the FIFO design's inspiration from [2].

C. CMEM

The CMEM contains the coefficients of the FIR filter. It is a single port SRAM with 64 16-bit words. The CMEM was generated by the memory compiler. Since all of the 64 values stored in the CMEM need to be used when calculating $y[n]$, the FIR coefficients are loaded into the CMEM at the beginning of the testbench, before the FIR core calculation.

D. FIR Filter Core

The core of the FIR filter uses all of the sub-modules described in the subsections above. The core instantiates all of the sub-modules and uses a finite state machine (FSM) to control the inputs and outputs of sub-modules and the FIR calculation. The algorithmic state machine of the FIR core is shown in Fig. 1.

1) Compilation: The core design was compiled using the Design Compiler (DC) software. Instead of splitting up the compilation into the individual modules, the entire design was compiled into one netlist. This meant that the DC software could optimize the design for the core alongside optimizing all of the sub-modules for the required timing.

III. Verification

To verify the functionality of ALU, we first implemented it in MATLAB and then compared the results of MATLAB and Verilog. We generated random 16-bit numbers, performed the computation, saved the output in an output file and then compared the data in the output file with Verilog's result inside a Verilog testbench.

To verify the functionality of CMEM, we performed a write operation at each memory address with a random 16 bit number and then performed a read operation at every address and compared the read and write numbers.

To check the FIFO, we write a new value to it and test if it has been written at the next cycle.

To test the core, we first perform the operation in MATLAB and save the result in a file. Then in Verilog, we first load every CMEM address with the coefficients. We send in the input signal's value, one at a time, and then compare the output with MATLAB's result.

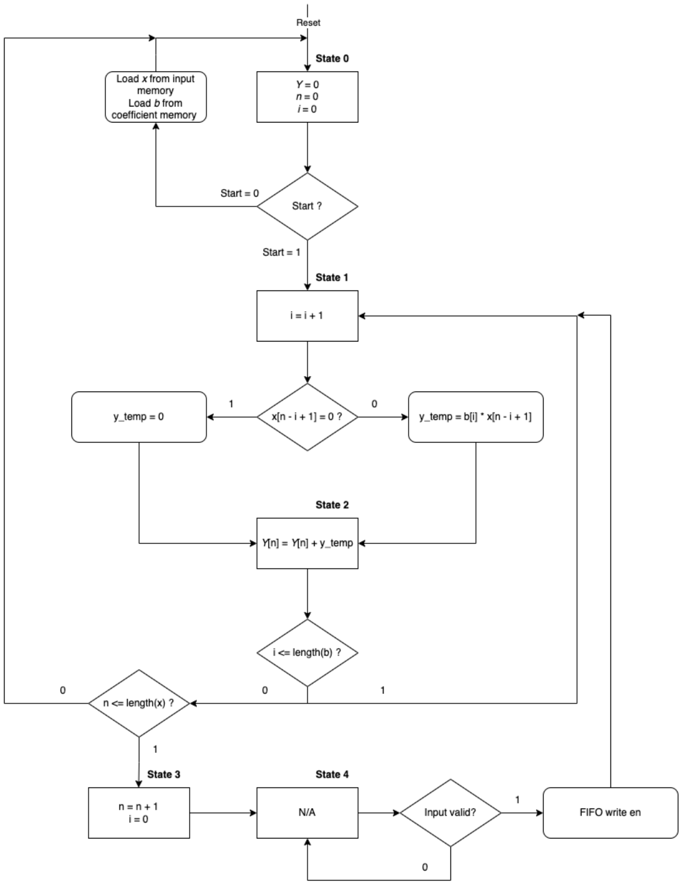
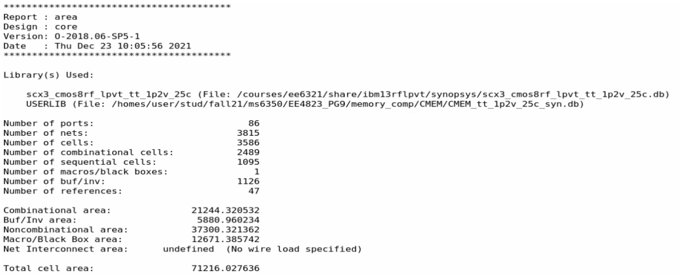


Fig. 1. ASM of FIR core

IV. Design Compiler (DC) Results

After compiling the FIR filter core using DC, we got the synthesized design with total cell area 71216. The report for total cell area is shown below in Fig. 2.



```
****************************************
Report : area
Design : core
Version: O-2018.06-SP5-1
Date   : Thu Dec 23 10:05:56 2021
****************************************

Library(s) Used:

    scx3_cmos8rf_lpvt_tt_1p2v_25c (File: /courses/ee6321/share/ibm13rflpvt/synopsys/scx3_cmos8rf_lpvt_tt_1p2v_25c.db)
    USERLIB (File: /homes/user/stud/fall21/ms6350/EE4823_PG9/memory_comp/CMEM/CMEM_tt_1p2v_25c_syn.db)

Number of ports:                           86
Number of nets:                          3815
Number of cells:                         3586
Number of combinational cells:           2489
Number of sequential cells:              1095
Number of macros/black boxes:               1
Number of buf/inv:                       1126
Number of references:                      47

Combinational area:              21244.320532
Buf/Inv area:                     5880.960234
Noncombinational area:           37300.321362
Macro/Black Box area:            12671.385742
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                 71216.027636
```

Fig. 2. Area Report of FIR Core

The simulation waveform of the synthesized FIR filter core are shown in Fig. 3, Fig. 4, and Fig. 5. The entire simulation process of the synthesized FIR filter core is shown in Fig. 3. The enlarged view of the simulation waveform of the FIR computation is shown in Fig. 4. The details of the multiplication and addition operations are shown in Fig. 5.

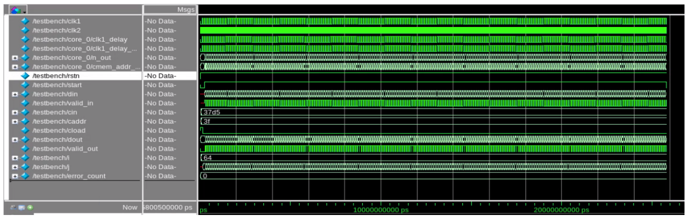


Fig. 3. Full Simulation Waveform of the FIR Core

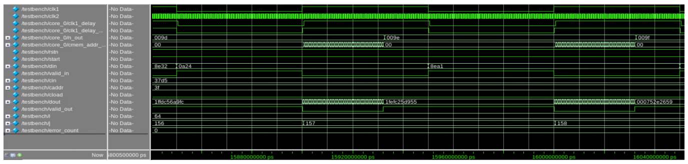


Fig. 4. Detailed Waveform of the FIR Computation

In order to verify the correctness of the FIR results calculated by our FIR filter core, we added code that could automatically compare the calculated FIR results by our FIR filter core against the results calculated by MATLAB with same set of input values. After running the simulation, we know from the simulator that our FIR filter core could produce the same results as the MATLAB code with same set of input values, which means our FIR filter core works correctly. The output message from the simulator that could show the FIR core works as expected is in Fig. 6. The core was completed with

V. PrimeTime (PT) Results

After simulating our synthesized FIR filter core and verifying the correctness, we ran PrimeTime to get the timing and power information of our FIR filter core.

A. Timing Analysis

The report of timing of the maximum path in our design is shown in Fig. 7, and the report of the timing of the minimum path in our design is shown in Fig. 8.

From both MAX and MIN timing reports, we know that our synthesized FIR filter core does not have timing violations, since the slacks for both MAX and MIN path are positive.

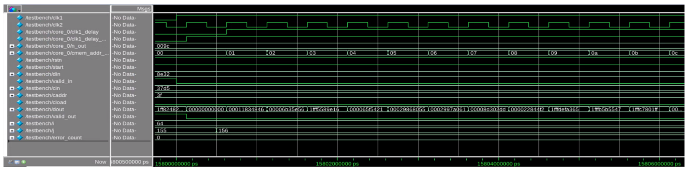
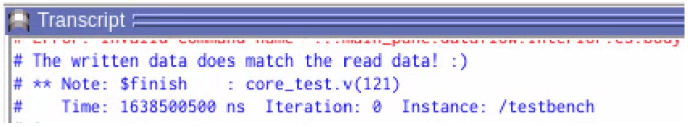
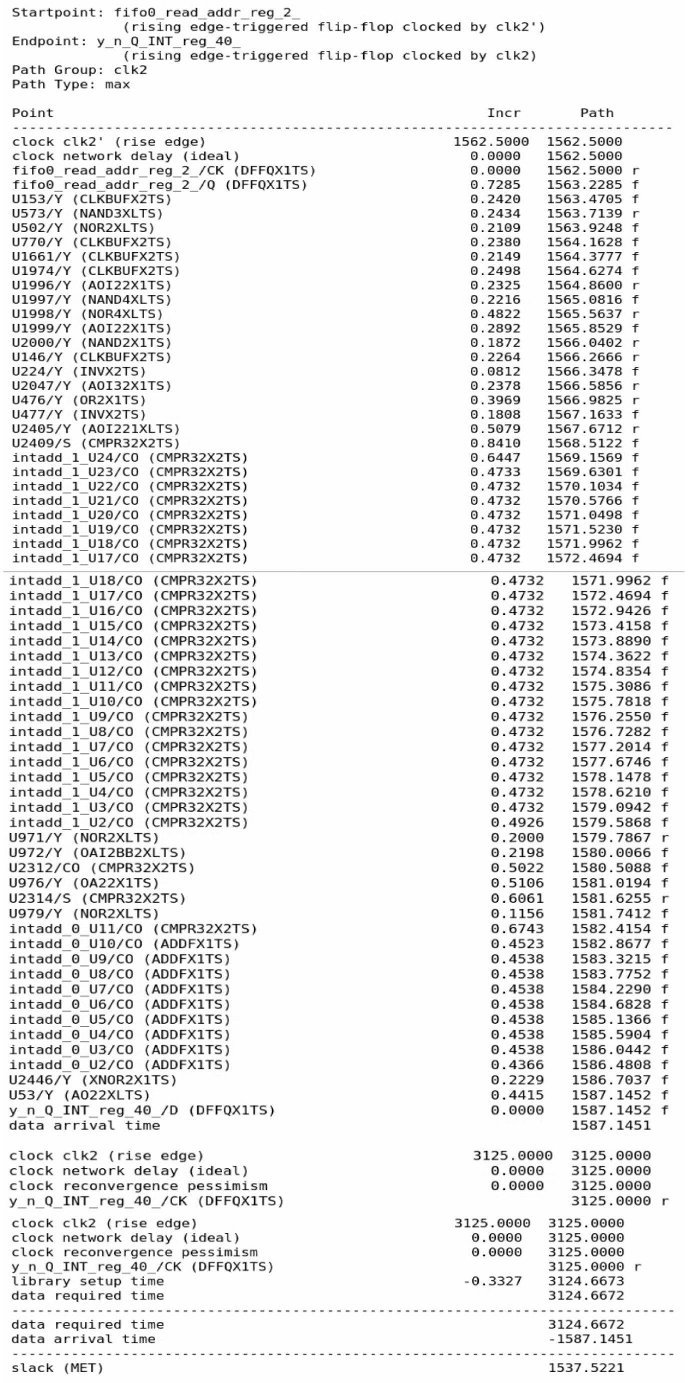


Fig. 5. Detailed Waveform of the Multiplication and Addition



```
# The written data does match the read data! :)
# ** Note: $finish    : core_test.v(121)
#    Time: 1638500500 ns  Iteration: 0  Instance: /testbench
```

Fig. 6. Output Message From the Simulator

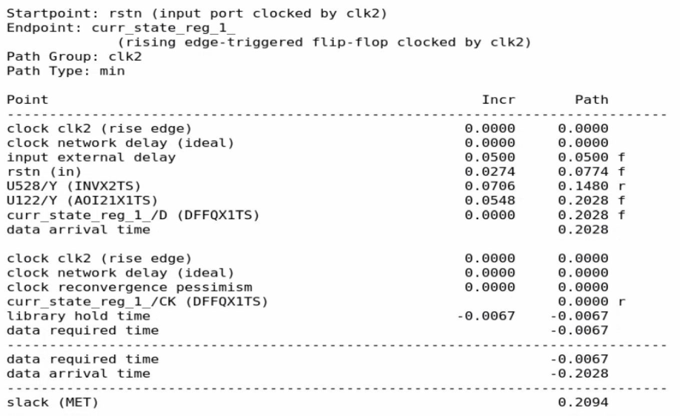


```
Startpoint: fifo0_read_addr_reg_2_
              (rising edge-triggered flip-flop clocked by clk2')
Endpoint: y_n_Q_INT_reg_40_
              (rising edge-triggered flip-flop clocked by clk2)
Path Group: clk2
Path Type: max

Point                                                      Incr       Path
-------------------------------------------------------------------------------
clock clk2' (rise edge)                               1562.5000  1562.5000
clock network delay (ideal)                              0.0000  1562.5000
fifo0_read_addr_reg_2_/CK (DFFQX1TS)                     0.0000  1562.5000 r
fifo0_read_addr_reg_2_/Q (DFFQX1TS)                      0.7285  1563.2285 f
U153/Y (CLKBUFX2TS)                                      0.2420  1563.4705 f
U573/Y (NAND3XLTS)                                       0.2434  1563.7139 r
U502/Y (NOR2XLTS)                                        0.2109  1563.9248 f
U770/Y (CLKBUFX2TS)                                      0.2380  1564.1628 f
U1661/Y (CLKBUFX2TS)                                     0.2149  1564.3777 f
U1974/Y (CLKBUFX2TS)                                     0.2498  1564.6274 f
U1996/Y (AOI22X1TS)                                      0.2325  1564.8600 r
U1997/Y (NAND4XLTS)                                      0.2216  1565.0816 f
U1998/Y (NOR4XLTS)                                       0.4822  1565.5637 r
U1999/Y (AOI22X1TS)                                      0.2892  1565.8529 f
U2000/Y (NAND2X1TS)                                      0.1872  1566.0402 r
U146/Y (CLKBUFX2TS)                                      0.2264  1566.2666 r
U224/Y (INVX2TS)                                         0.0812  1566.3478 f
U2047/Y (AOI32X1TS)                                      0.2378  1566.5856 r
U476/Y (OR2X1TS)                                         0.3969  1566.9825 r
U477/Y (INVX2TS)                                         0.1808  1567.1633 f
U2405/Y (AOI221XLTS)                                     0.5079  1567.6712 r
U2409/S (CMPR32X2TS)                                     0.8410  1568.5122 f
intadd_1_U24/CO (CMPR32X2TS)                             0.6447  1569.1569 f
intadd_1_U23/CO (CMPR32X2TS)                             0.4733  1569.6301 f
intadd_1_U22/CO (CMPR32X2TS)                             0.4732  1570.1034 f
intadd_1_U21/CO (CMPR32X2TS)                             0.4732  1570.5766 f
intadd_1_U20/CO (CMPR32X2TS)                             0.4732  1571.0498 f
intadd_1_U19/CO (CMPR32X2TS)                             0.4732  1571.5230 f
intadd_1_U18/CO (CMPR32X2TS)                             0.4732  1571.9962 f
intadd_1_U17/CO (CMPR32X2TS)                             0.4732  1572.4694 f
```

```
intadd_1_U18/CO (CMPR32X2TS)                             0.4732   1571.9962 f
intadd_1_U17/CO (CMPR32X2TS)                             0.4732   1572.4694 f
intadd_1_U16/CO (CMPR32X2TS)                             0.4732   1572.9426 f
intadd_1_U15/CO (CMPR32X2TS)                             0.4732   1573.4158 f
intadd_1_U14/CO (CMPR32X2TS)                             0.4732   1573.8890 f
intadd_1_U13/CO (CMPR32X2TS)                             0.4732   1574.3622 f
intadd_1_U12/CO (CMPR32X2TS)                             0.4732   1574.8354 f
intadd_1_U11/CO (CMPR32X2TS)                             0.4732   1575.3086 f
intadd_1_U10/CO (CMPR32X2TS)                             0.4732   1575.7818 f
intadd_1_U9/CO (CMPR32X2TS)                              0.4732   1576.2550 f
intadd_1_U8/CO (CMPR32X2TS)                              0.4732   1576.7282 f
intadd_1_U7/CO (CMPR32X2TS)                              0.4732   1577.2014 f
intadd_1_U6/CO (CMPR32X2TS)                              0.4732   1577.6746 f
intadd_1_U5/CO (CMPR32X2TS)                              0.4732   1578.1478 f
intadd_1_U4/CO (CMPR32X2TS)                              0.4732   1578.6210 f
intadd_1_U3/CO (CMPR32X2TS)                              0.4732   1579.0942 f
intadd_1_U2/CO (CMPR32X2TS)                              0.4926   1579.5868 f
U971/Y (NOR2XLTS)                                        0.2000   1579.7867 r
U972/Y (OAI2BB2XLTS)                                     0.2198   1580.0066 f
U2312/CO (CMPR32X2TS)                                    0.5022   1580.5088 f
U976/Y (OA22X1TS)                                        0.5106   1581.0194 f
U2314/S (CMPR32X2TS)                                     0.6061   1581.6255 r
U979/Y (NOR2XLTS)                                        0.1156   1581.7412 f
intadd_0_U11/CO (CMPR32X2TS)                             0.6743   1582.4154 f
intadd_0_U10/CO (ADDFX1TS)                               0.4523   1582.8677 f
intadd_0_U9/CO (ADDFX1TS)                                0.4538   1583.3215 f
intadd_0_U8/CO (ADDFX1TS)                                0.4538   1583.7752 f
intadd_0_U7/CO (ADDFX1TS)                                0.4538   1584.2290 f
intadd_0_U6/CO (ADDFX1TS)                                0.4538   1584.6828 f
intadd_0_U5/CO (ADDFX1TS)                                0.4538   1585.1366 f
intadd_0_U4/CO (ADDFX1TS)                                0.4538   1585.5904 f
intadd_0_U3/CO (ADDFX1TS)                                0.4538   1586.0442 f
intadd_0_U2/CO (ADDFX1TS)                                0.4366   1586.4808 f
U2446/Y (XNOR2X1TS)                                      0.2229   1586.7037 f
U53/Y (AO22XLTS)                                         0.4415   1587.1452 f
y_n_Q_INT_reg_40_/D (DFFQX1TS)                           0.0000   1587.1452 f
data arrival time                                                 1587.1451

clock clk2 (rise edge)                                3125.0000   3125.0000
clock network delay (ideal)                              0.0000   3125.0000
clock reconvergence pessimism                            0.0000   3125.0000
y_n_Q_INT_reg_40_/CK (DFFQX1TS)                                   3125.0000 r
```

```
clock clk2 (rise edge)                               3125.0000  3125.0000
clock network delay (ideal)                             0.0000  3125.0000
clock reconvergence pessimism                           0.0000  3125.0000
y_n_Q_INT_reg_40_/CK (DFFQX1TS)                                 3125.0000 r
library setup time                                     -0.3327  3124.6673
data required time                                              3124.6672
-------------------------------------------------------------------------------
data required time                                              3124.6672
data arrival time                                               -1587.1451
-------------------------------------------------------------------------------
slack (MET)                                                     1537.5221
```

Fig. 7. MAX Path Timing Report

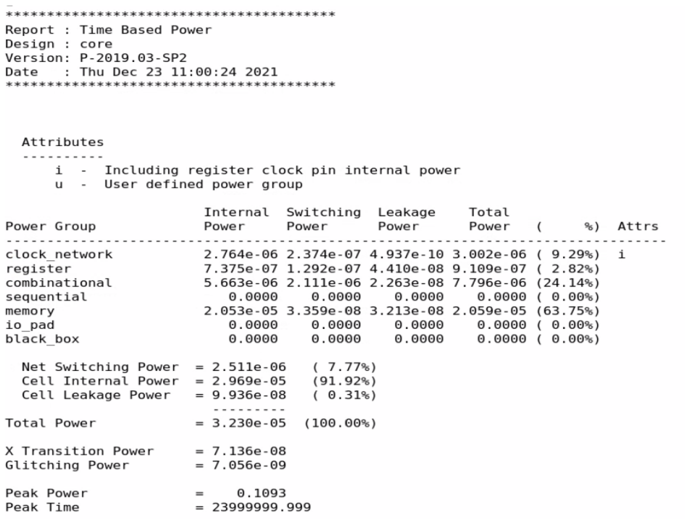


```
Startpoint: rstn (input port clocked by clk2)
Endpoint: curr_state_reg_1_
             (rising edge-triggered flip-flop clocked by clk2)
Path Group: clk2
Path Type: min

Point                                                     Incr       Path
-------------------------------------------------------------------------------
clock clk2 (rise edge)                                   0.0000     0.0000
clock network delay (ideal)                              0.0000     0.0000
input external delay                                     0.0500     0.0500 f
rstn (in)                                                0.0274     0.0774 f
U528/Y (INVX2TS)                                         0.0706     0.1480 r
U122/Y (AOI21X1TS)                                       0.0548     0.2028 f
curr_state_reg_1_/D (DFFQX1TS)                           0.0000     0.2028 f
data arrival time                                                   0.2028

clock clk2 (rise edge)                                   0.0000     0.0000
clock network delay (ideal)                              0.0000     0.0000
clock reconvergence pessimism                            0.0000     0.0000
curr_state_reg_1_/CK (DFFQX1TS)                                     0.0000 r
library hold time                                       -0.0067    -0.0067
data required time                                                 -0.0067
-------------------------------------------------------------------------------
data required time                                                 -0.0067
data arrival time                                                  -0.2028
-------------------------------------------------------------------------------
slack (MET)                                                         0.2094
```

Fig. 8. MIN Path Timing Report

B. Power Analysis

The report of power for our FIR filter core is shown in Fig. 9.



```
****************************************
Report : Time Based Power
Design : core
Version: P-2019.03-SP2
Date   : Thu Dec 23 11:00:24 2021
****************************************


  Attributes
  ----------
      i  -  Including register clock pin internal power
      u  -  User defined power group

                        Internal  Switching  Leakage    Total
Power Group             Power     Power      Power      Power   (     %)  Attrs
--------------------------------------------------------------------------------
clock_network           2.764e-06 2.374e-07 4.937e-10 3.002e-06 ( 9.29%)  i
register                7.375e-07 1.292e-07 4.410e-08 9.109e-07 ( 2.82%)
combinational           5.663e-06 2.111e-06 2.263e-08 7.796e-06 (24.14%)
sequential                 0.0000    0.0000    0.0000    0.0000 ( 0.00%)
memory                  2.053e-05 3.359e-08 3.213e-08 2.059e-05 (63.75%)
io_pad                     0.0000    0.0000    0.0000    0.0000 ( 0.00%)
black_box                  0.0000    0.0000    0.0000    0.0000 ( 0.00%)

  Net Switching Power  = 2.511e-06   ( 7.77%)
  Cell Internal Power  = 2.969e-05   (91.92%)
  Cell Leakage Power   = 9.936e-08   ( 0.31%)
                         ---------
Total Power            = 3.230e-05  (100.00%)

X Transition Power     = 7.136e-08
Glitching Power        = 7.056e-09

Peak Power             =    0.1093
Peak Time              = 23999999.999
```

Fig. 9. Power Report

From the power report, we know that our FIR filter core consumes total power of $3.23E^{-5}mW$ for 256 samples. PrimeTime was not run on 10,000 samples due to the extremely long computation time. The VCD for the 256 sample run is attached in the archive.

VI. RMSE And NRMSE Results

$RMSE = 1.2133E^{-04}$, $NRMSE = 3.6100E^{-10}$ and Maximum Percent Error = $(6.7970E^{-05})(100)$ = 0.006797%.

References

[1] Discrete time graphical convolution example. Accessed: 12-23-2021. [Online]. Available: https://electricalacademia.com/signals-and-systems/example-of-discrete-time-graphical-convolution/

[2] C. E. Cummings, "Simulation and synthesis techniques for asynchronous fifo design," 2002, Sunburst Design.