EE 599 Spring 2020

Homework2

ChaoGaogaoc@usc.edu,4458723430

Github: https://github.com/gaochaousc/EE-599_-gaoc-_-4458723430-

2020/03/05

1 Barrel Shifter [50 Points]

A Barrel Shifter is a logic circuit for shifting a word by a varying amount. It has a control input (select bits) that specifies the number of bit positions that it shifts. A Barrel Shifter is implemented with a sequence of shift multiplexers, each shifting a word by 2^k bit positions for different values of k. The diagram below (Figure 1) shows a pipeline clock-wise barrel shifter for 8 inputs.

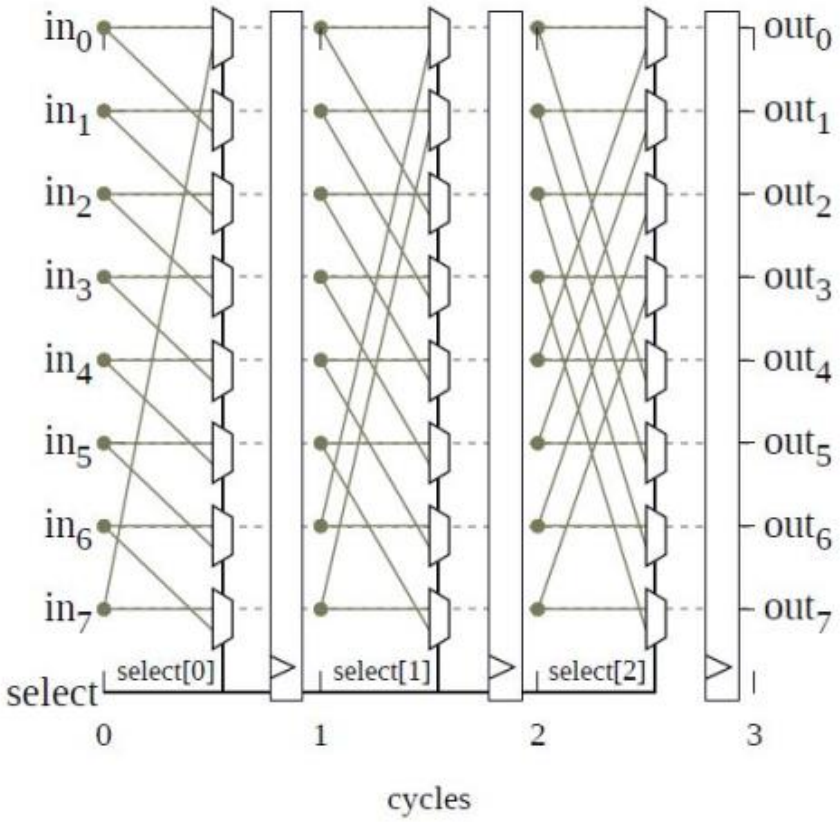


Figure 1: An Example Pipeline Barrel Shifter

Notice that we can shift the input elements by 0 - 7 elements using an implementation like Figure 1. Table 1 shows all the shifting possibilities for the above example.

Similarly, a scalable Barrel Shifter with N inputs and maximum shift of N-1 can be implemented.

Table 1: All the shifting possibilities for the given example

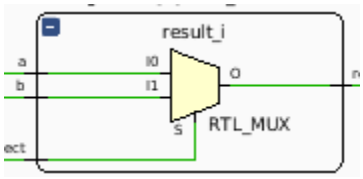
Select[2]	Select[1]	Select[0]	Shift
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

1.1 Implementation : Using Xilinx Vivado Software (Use Zynq-7000 xc7z007sclg225-2 FPGA)

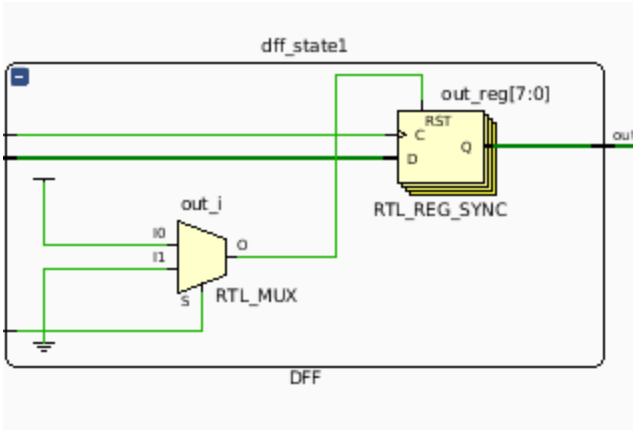
Consider only 8-bit arithmetic. **You must implement a scalable design.**

1. Implement a **barrel shifter** design in Verilog which takes n inputs with 8 bits and shift them by r (value of r is passed to select bits and $r_{max} = (n-1)$).
2. For a 16 elements design with a maximum shift of 15, write a testbench and verify the waveforms.
3. Elaborate the design and include all the schematics screenshots of the modules in the report.
4. Synthesis the design and include the schematics screenshots in the report.
5. Generate Resource and timing estimations and include them in the report.
6. Redo parts 3, 4, 5 for 64 elements (with maximum shift of 63).

1. Schematic Screenshot (for 8 elements)
   a. Mux

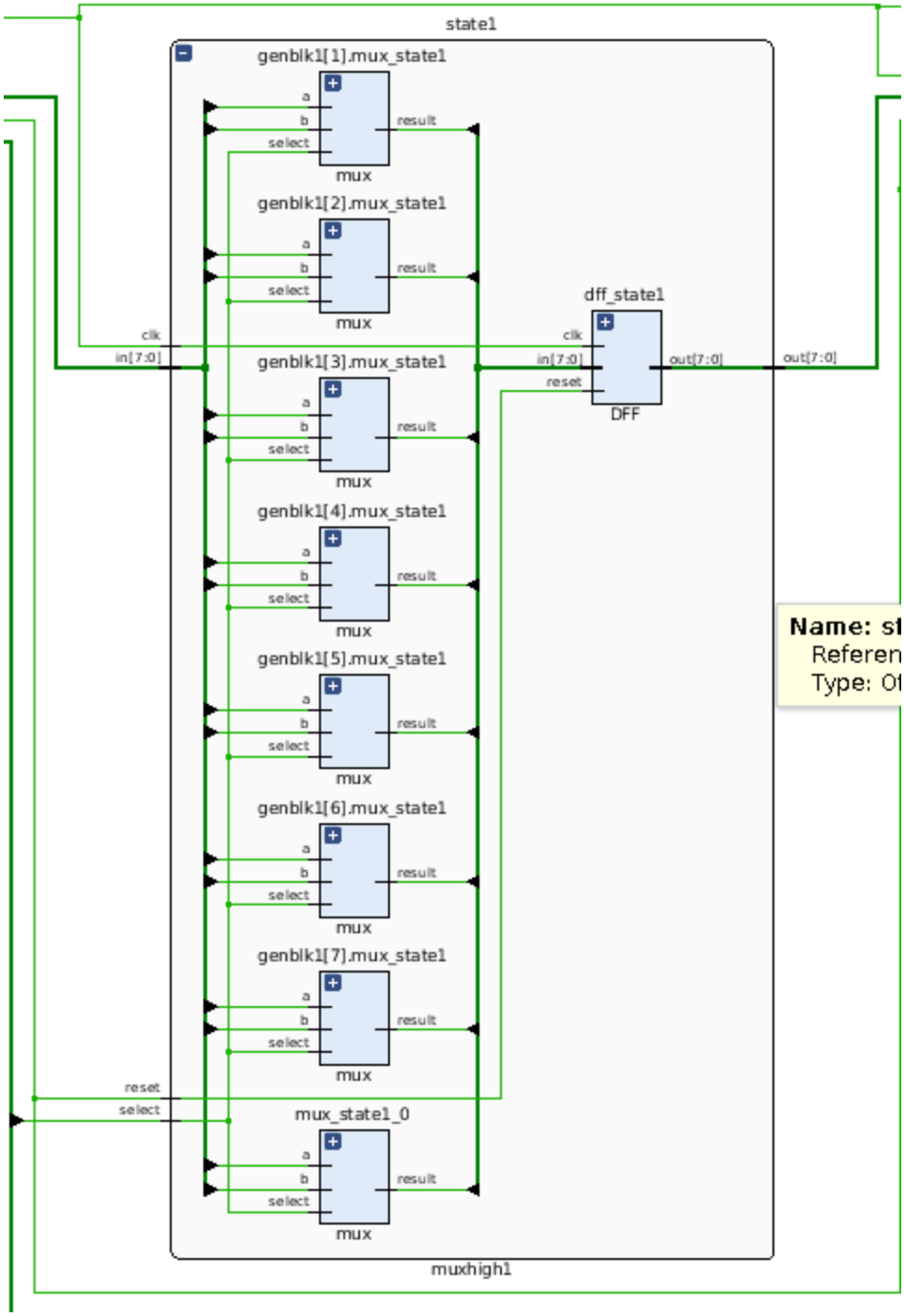


   b. D-FF



c. State Machine

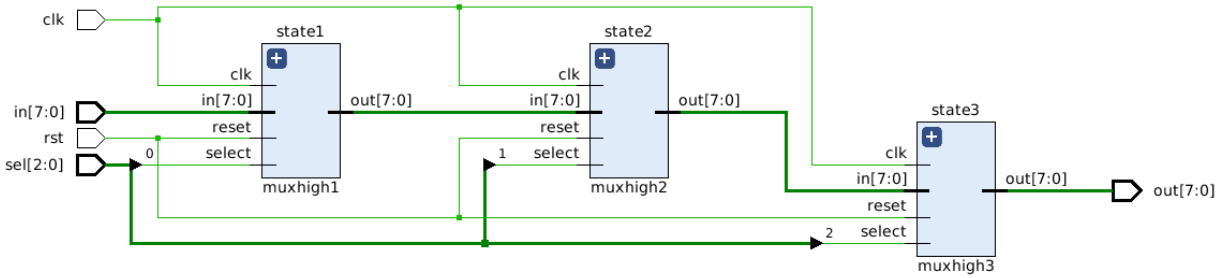
1. For single input

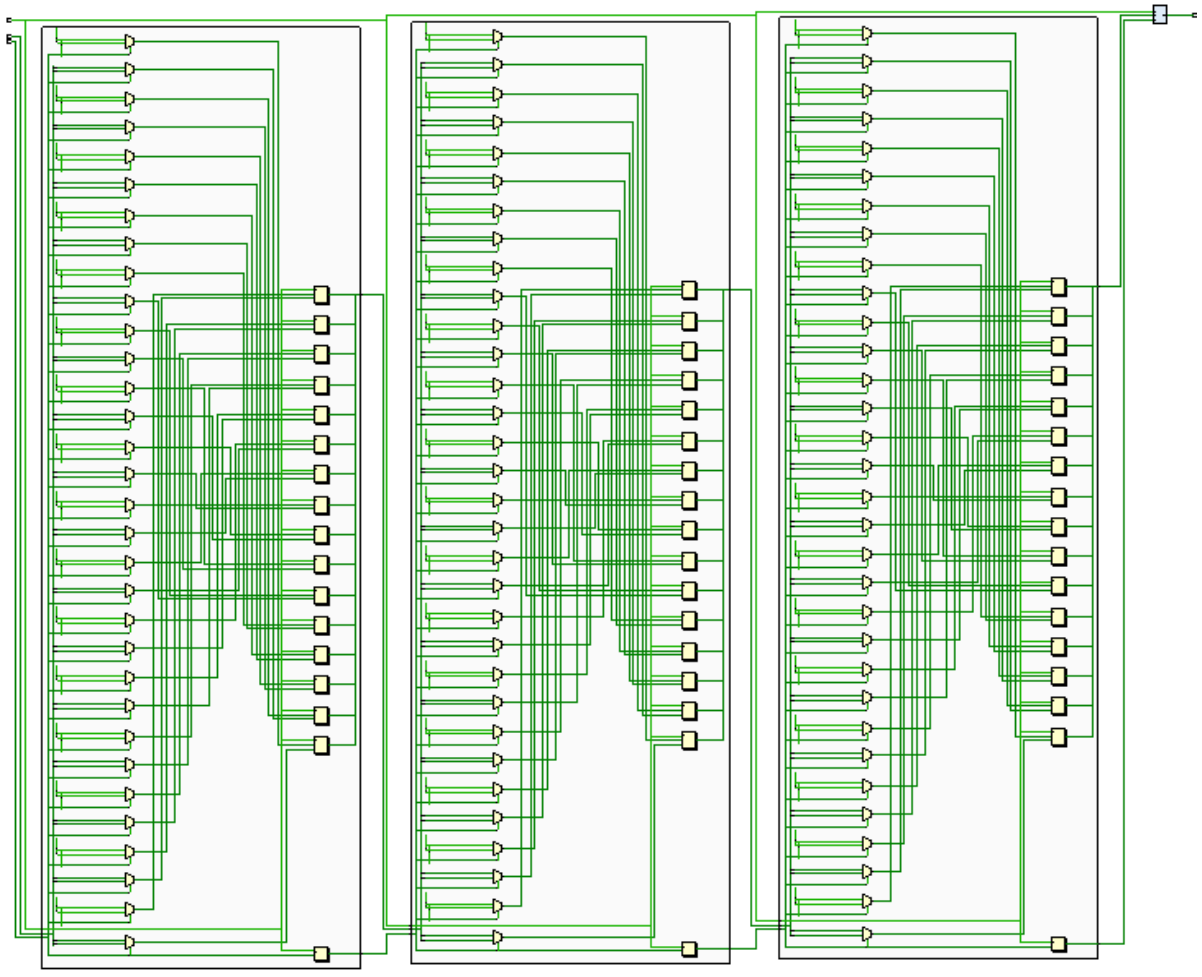


d. Whole Design

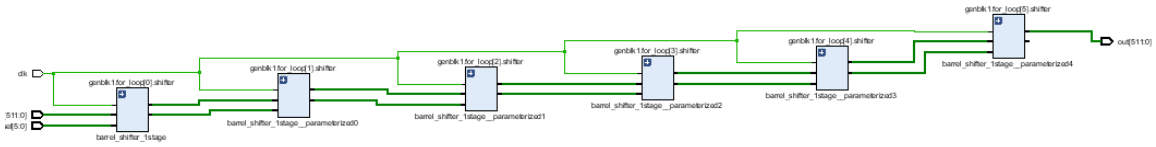
2. For multiple input data set =16,64

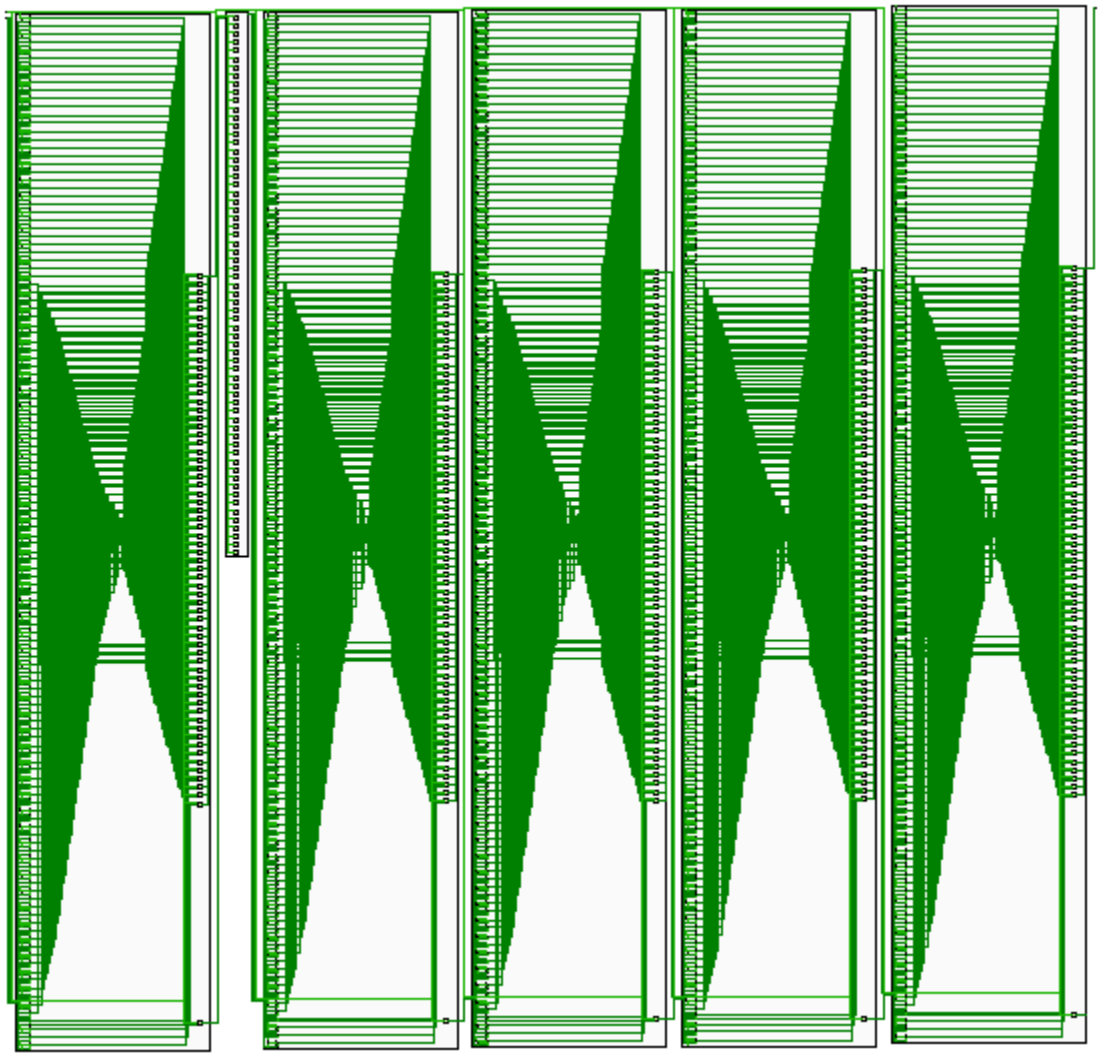
For data set =16:



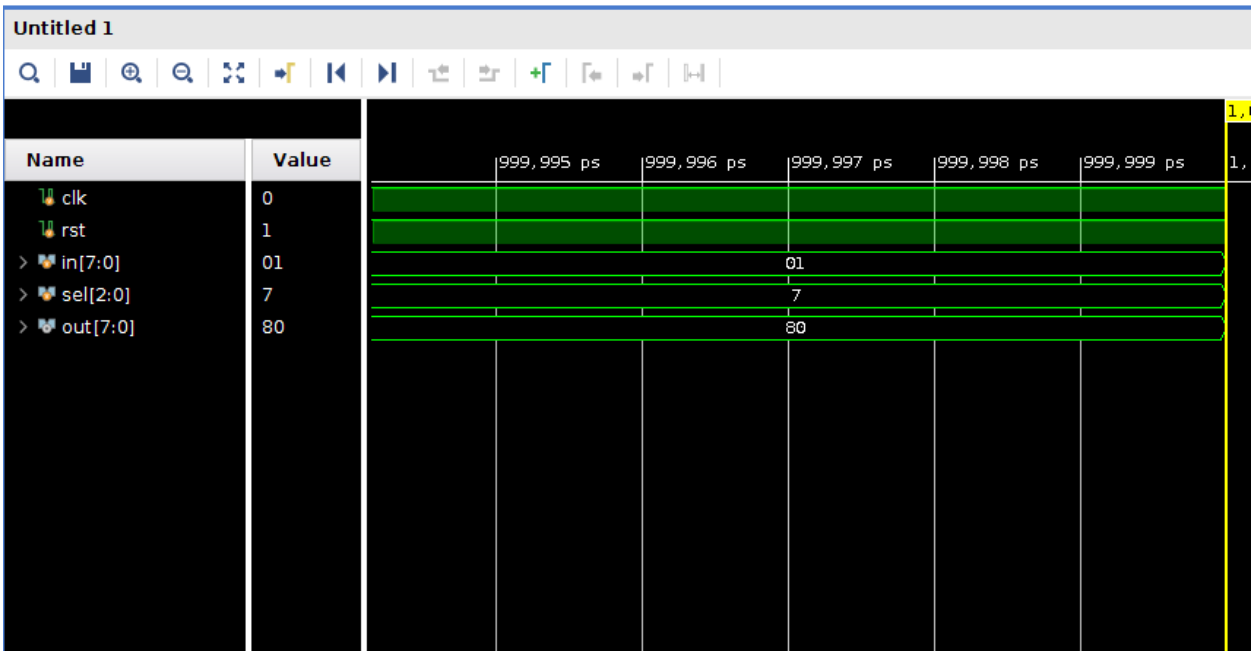


For 64 elements:





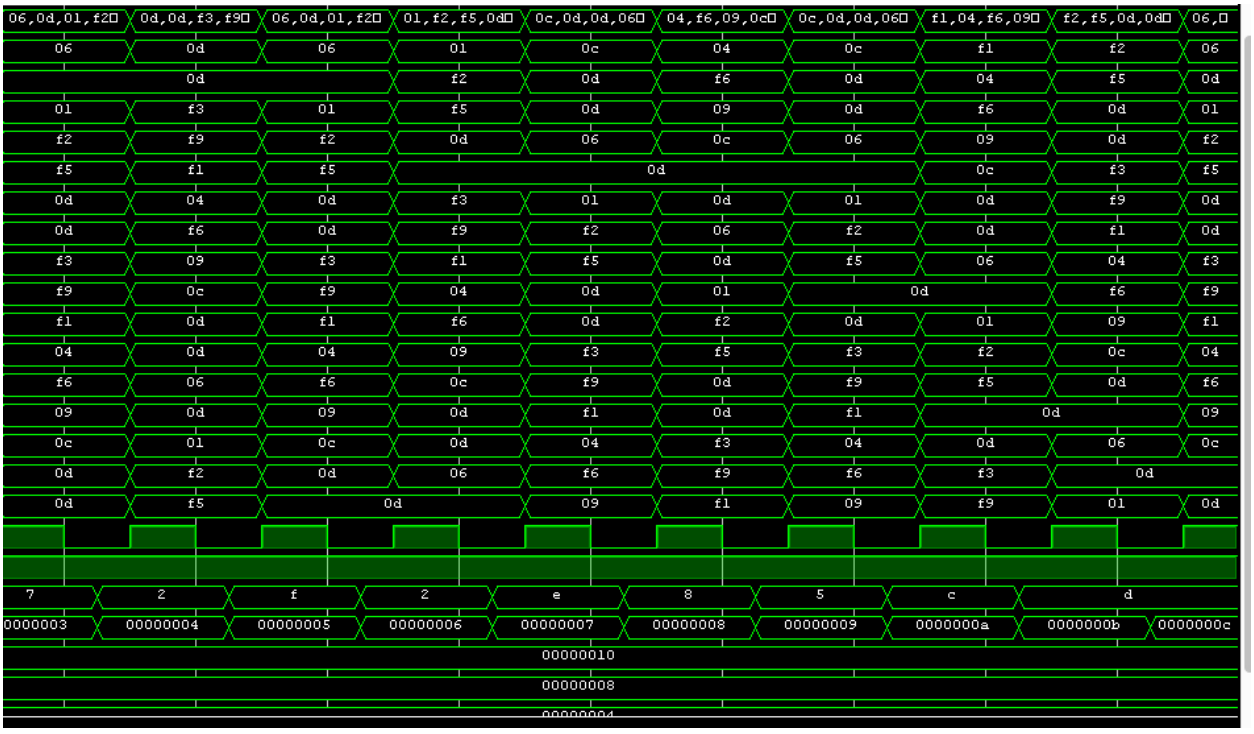
3. Function Waveform Test
   Here, we gave the input different value and different number for shifting and it returns different result. For single input and single output.



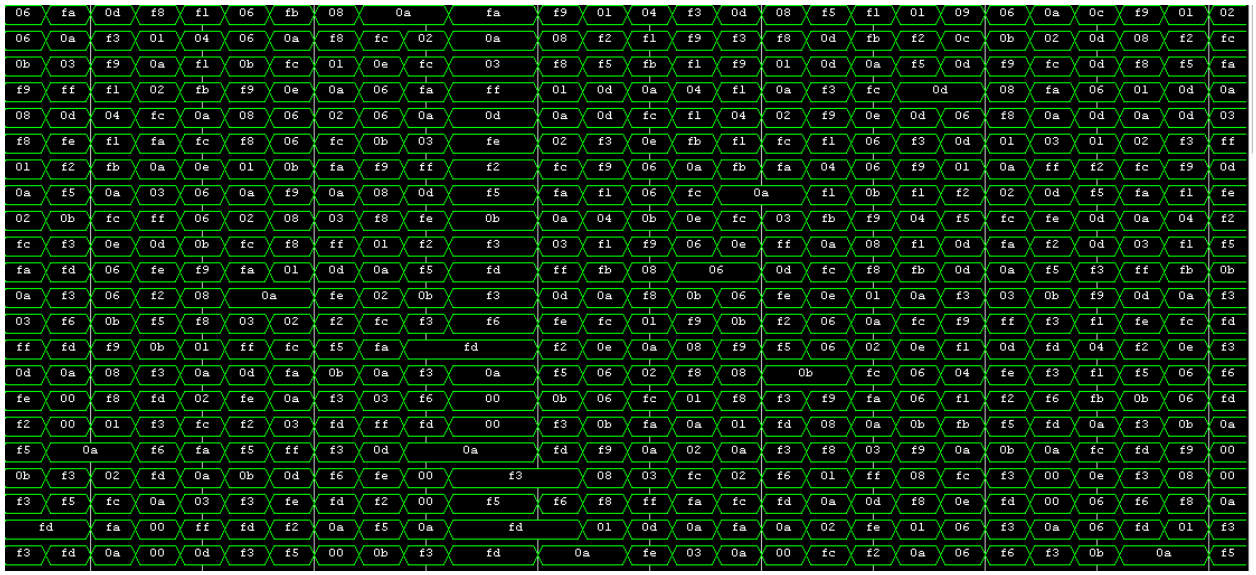
For multiple inputs and multiple output.

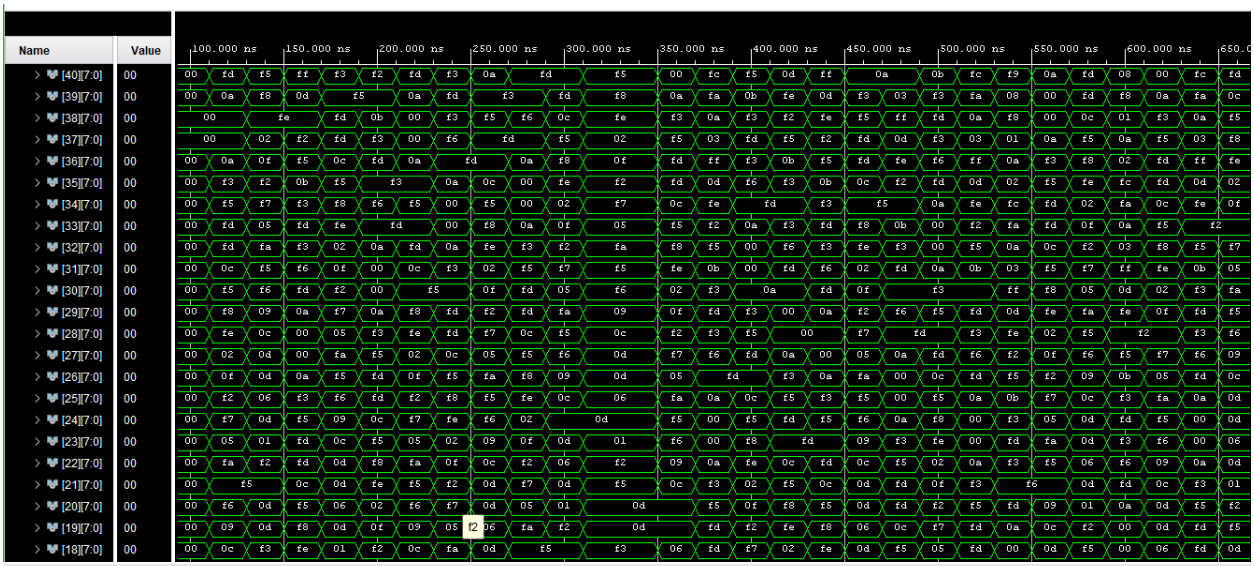
Size=16 Datasize=8bits:

Here comes the array, from <0> to <15>, and the outcome will be finished in the next several cycles.

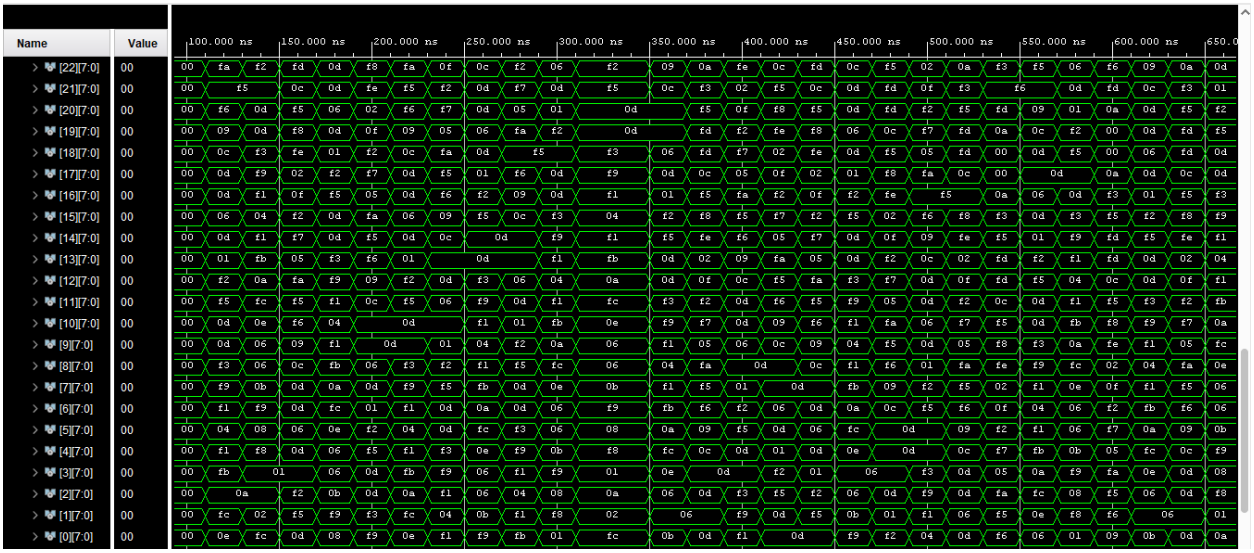


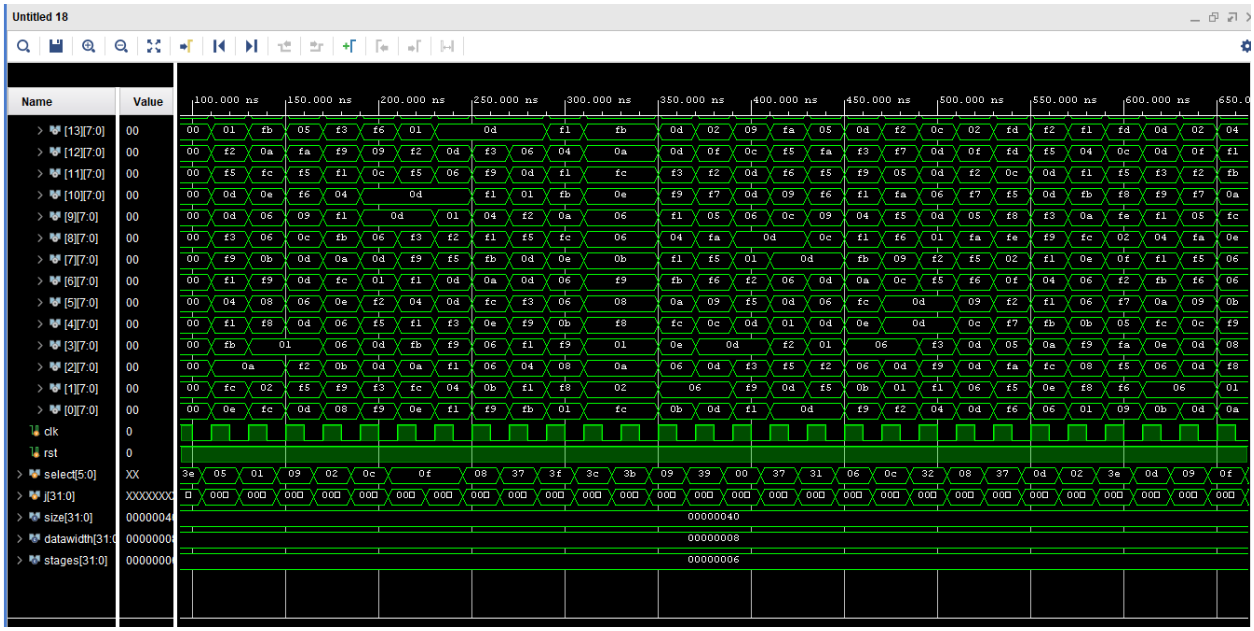
Size=64 Datasize=8bits:


06 fa 0d f8 f1 06 fb 08 0a fa f9 01 04 f3 0d 08 f5 f1 01 09 06 0a 0c f9 01 02
06 0a f3 01 04 06 0a f8 fc 02 0a 08 f2 f1 f9 f3 f8 0d fb f2 0c 0b 02 0d 08 f2 fc
0b 03 f9 0a f1 0b fc 01 0e fc 03 f8 f5 fb f1 f9 01 0d 0a f5 0d f9 fc 0d f8 f5 fa
f9 ff f1 02 fb f9 0e 0a 06 fa ff 01 0d 0a 04 f1 0a f3 fc 0d 08 fa 06 01 0d 0a
08 0d 04 fc 0a 08 06 02 06 0a 0d 0a 0d fc f1 04 02 f9 0e 0d 06 f8 0a 0d 0a 0d 03
f8 fe f1 fa fc f8 06 fc 0b 03 fe 02 f3 0e fb f1 fc f1 06 f3 0d 01 03 01 02 f3 ff
01 f2 fb 0a 0e 01 0b fa f9 ff f2 fc f9 06 0a fb fa 04 06 f9 01 0a ff f2 fc f9 0d
0a f5 0a 03 06 0a f9 0a 08 0d f5 fa f1 06 fc 0a f1 0b f1 f2 02 0d f5 fa f1 fe
02 0b fc ff 06 02 08 03 f8 fe 0b 0a 04 0b 0e fc 03 fb f9 04 f5 fc fe 0d 0a 04 f2
fc f3 0e 0d 0b fc f8 ff 01 f2 f3 03 f1 f9 06 0e ff 0a 08 f1 0d fa f2 0d 03 f1 f5
fa fd 06 fe f9 fa 01 0d 0a f5 fd ff fb 08 06 0d fc f8 fb 0d 0a f5 f3 ff fb 0b
0a f3 06 f2 08 0a fe 02 0b f3 0d 0a f8 0b 06 fe 0e 01 0a f3 03 0b f9 0d 0a f3
03 f6 0b f5 f8 03 02 f2 fc f3 f6 fe fc 01 f9 0b f2 06 0a fc f9 ff f3 f1 fe fc fd
ff fd f9 0b 01 ff fc f5 fa fd f2 0e 0a 08 f9 f5 06 02 0e f1 0d fd 04 f2 0e f3
0d 0a 08 f3 0a 0d fa 0b 0a f3 0a f5 06 02 f8 08 0b fc 06 04 fe f3 f1 f5 06 f6
fe 00 f8 fd 02 fe 0a f3 03 f6 00 0b 06 fc 01 f8 f3 f9 fa 06 f1 f2 f6 fb 0b 06 fd
f2 00 01 f3 fc f2 03 fd ff fd 00 f3 0b fa 0a 01 fd 08 0a 0b fb f5 fd 0a f3 0b 0a
f5 0a f6 fa f5 ff f3 0d 0a fd f9 0a 02 0a f3 f8 03 f9 0a 0b 0a fc fd f9 00
0b f3 02 fd 0a 0b 0d f6 fe 00 f3 08 03 fc 02 f6 01 ff 08 fc f3 00 0e f3 08 00
f3 f5 fc 0a 03 f3 fe fd f2 00 f5 f6 f8 ff fa fc fd 0a 0d f8 0e fd 00 06 f6 f8 0a
fd fa 00 ff fd f2 0a f5 0a fd 01 0d 0a fa 0a 02 fe 01 06 f3 0a 06 fd 01 f3
f3 fd 0a 00 0d f3 f5 00 0b f3 fd 0a fe 03 0a 00 fc f2 0a 06 f6 f3 0b 0a f5

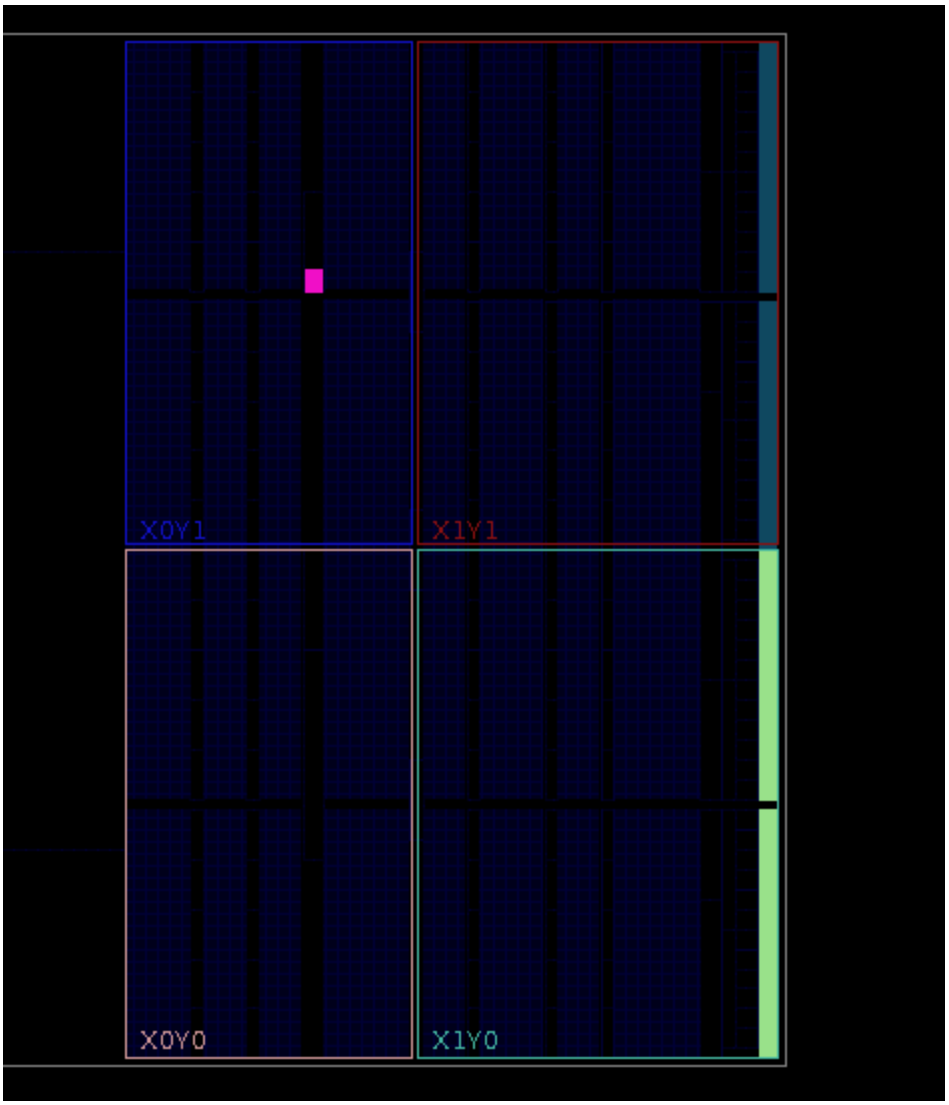

Name
Value
100.000 ns
150.000 ns
200.000 ns
250.000 ns
300.000 ns
350.000 ns
400.000 ns
450.000 ns
500.000 ns
550.000 ns
600.000 ns
[40][7:0]
[39][7:0]
[38][7:0]
[37][7:0]
[36][7:0]
[35][7:0]
[34][7:0]
[33][7:0]
[32][7:0]
[31][7:0]
[30][7:0]
[29][7:0]
[28][7:0]
[27][7:0]
[26][7:0]
[25][7:0]
[24][7:0]
[23][7:0]
[22][7:0]
[21][7:0]
[20][7:0]
[19][7:0]
[18][7:0]

Untitled 18


Name
Value
100.000 ns
150.000 ns
200.000 ns
250.000 ns
300.000 ns
350.000 ns
400.000 ns
450.000 ns
500.000 ns
550.000 ns
600.000 ns
[22][7:0]
[21][7:0]
[20][7:0]
[19][7:0]
[18][7:0]
[17][7:0]
[16][7:0]
[15][7:0]
[14][7:0]
[13][7:0]
[12][7:0]
[11][7:0]
[10][7:0]
[9][7:0]
[8][7:0]
[7][7:0]
[6][7:0]
[5][7:0]
[4][7:0]
[3][7:0]
[2][7:0]
[1][7:0]
[0][7:0]



4. Synthesis Screenshot



5. Timing Estimation

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.609 ns	Worst Hold Slack (WHS):	0.127 ns	Worst Pulse Width Slack (WPWS):	2.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	16	Total Number of Endpoints:	16	Total Number of Endpoints:	25

All user specified timing constraints are met.

6. Resources Estimation

Hierarchy

Name	Slice LUTs (14400)	Slice Registers (28800)	Bonded IOB (54)	BUFGCTRL (32)
top	13	24	21	1
state1 (muxhigh1)	8	8	0	0
state2 (muxhigh2)	4	8	0	0
state3 (muxhigh3)	1	8	0	0

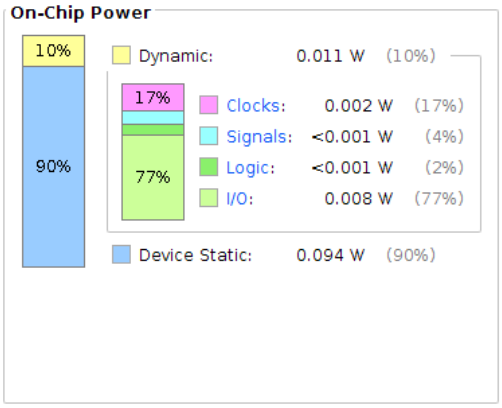
7. Power Estimation

Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

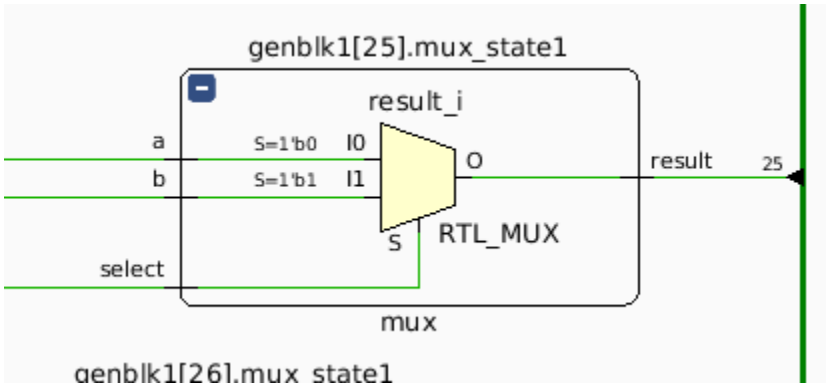
Total On-Chip Power:	**0.105 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**26.2°C**
Thermal Margin:	73.8°C (6.2 W)
Effective θJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

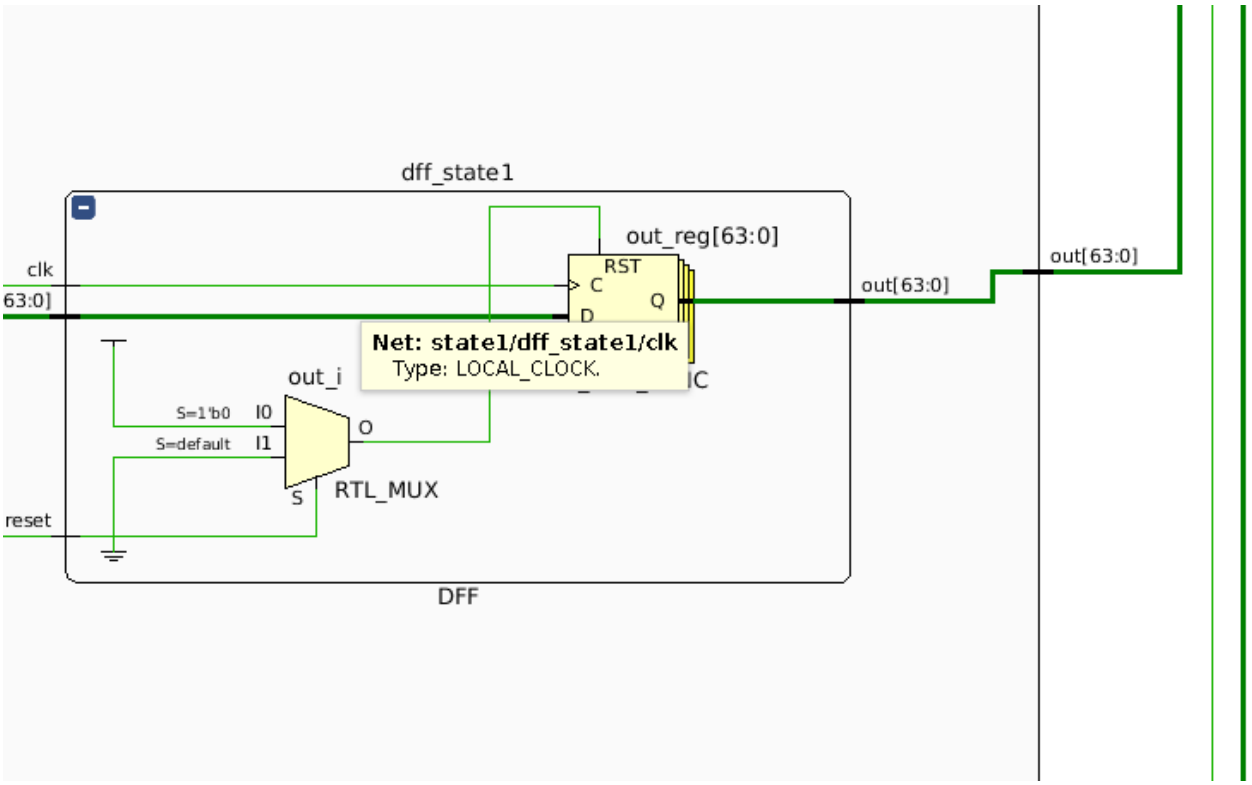


1. Schematic Screenshot (for 64 elements)

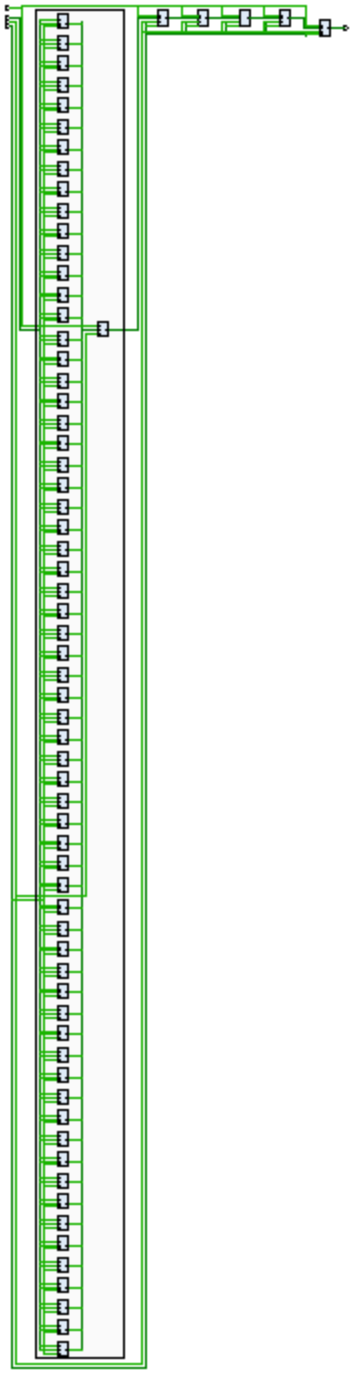
a. Mux



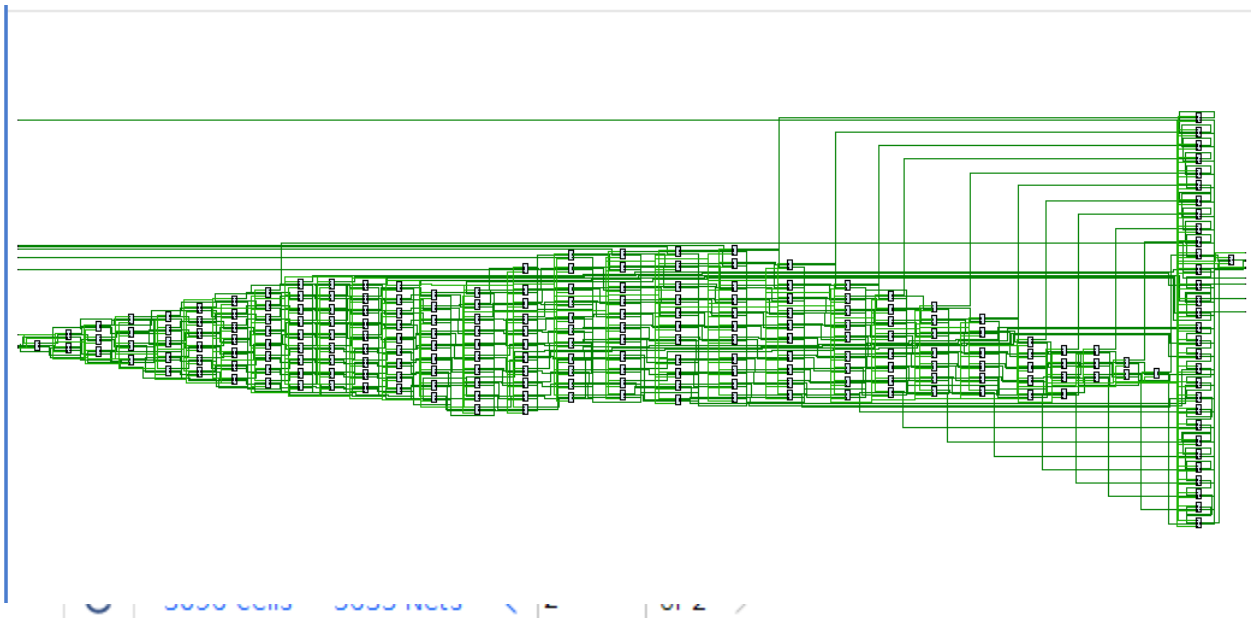
b. D-FF

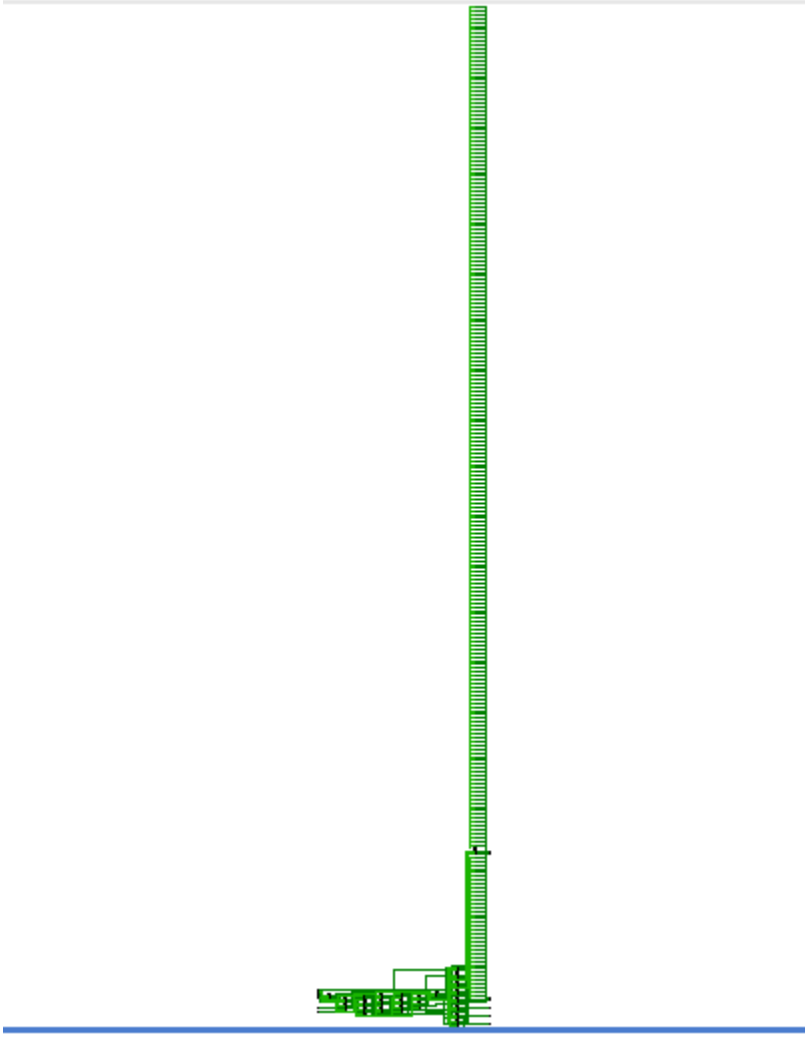


c. State Machine

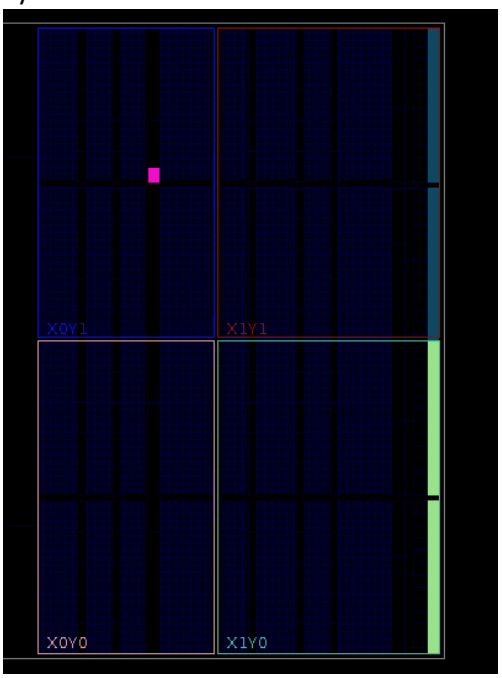


d. Whole Design





2. Synthesis Screenshot



3. Timing Estimation

4. Resources Estimation

Hierarchy

Name	Slice LUTs (14400)	Slice Registers (28800)	Bonded IOB (54)	BUFGCTRL (32)
top	193	384	136	1
state1 (state1)	64	64	0	0
state2 (state2)	32	64	0	0
state3 (state3)	32	64	0	0
state4 (state4)	32	64	0	0
state5 (state5)	32	64	0	0
state6 (state6)	1	64	0	0

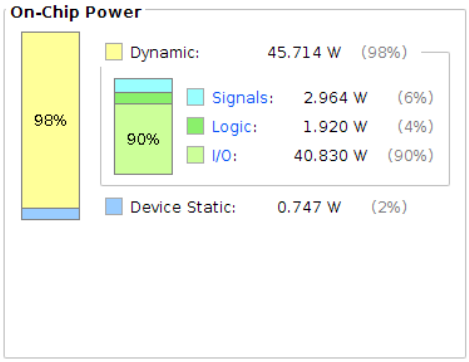
5. Power Estimation

Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**46.462 W (Junction temp exceeded!)**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**125.0°C**
Thermal Margin:	-460.9°C (-39.5 W)
Effective θJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity



2 Systolic Array for Dense Matrix-Matrix Multiplication [50 Points]

The Systolic Architecture consists of an array of processing elements, where data flows between neighboring elements, synchronously, from different directions. Processing element takes data from Top, Left, and output the results to Right, Bottom.

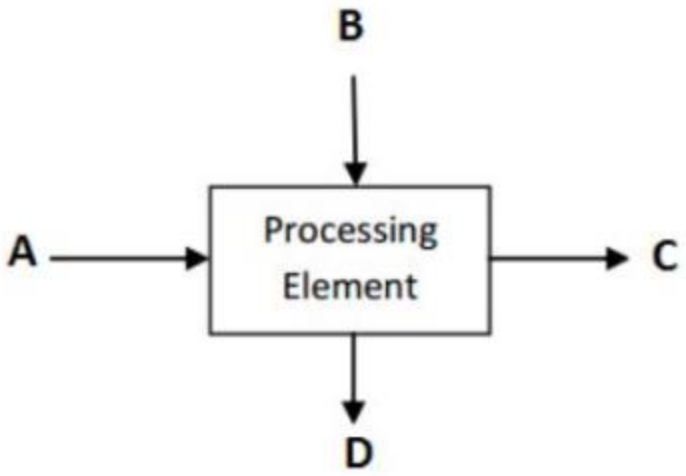


Figure 2: Inputs and outputs to the Processing Elements

One of the main applications of Systolic Architecture is matrix multiplication. As the following figure depicts, in_a, in_b are inputs to the processing element and out_a, out_b are output to the processing element. out_c is to get the output result of each processing element.

Processing elements are arranged in the form of an array. In the following example, we

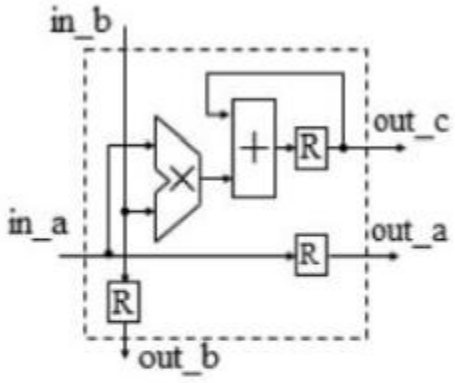


Figure 3: Internals of the PE

analyze, multiplication of 3×3 matrices, which can be easily extended. Let say the two matrices are A and B. Figure 4 depicts how matrix A and B are fed into PE array.

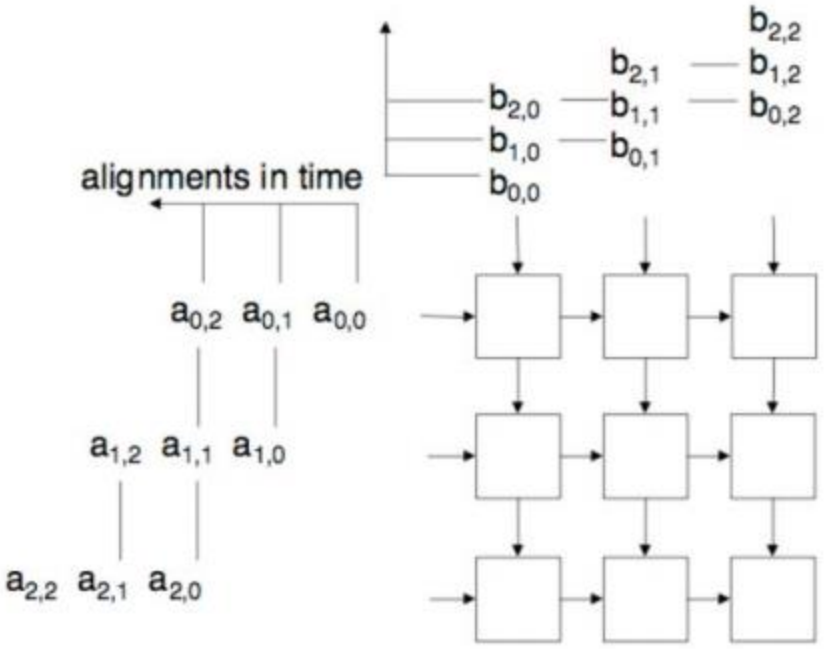


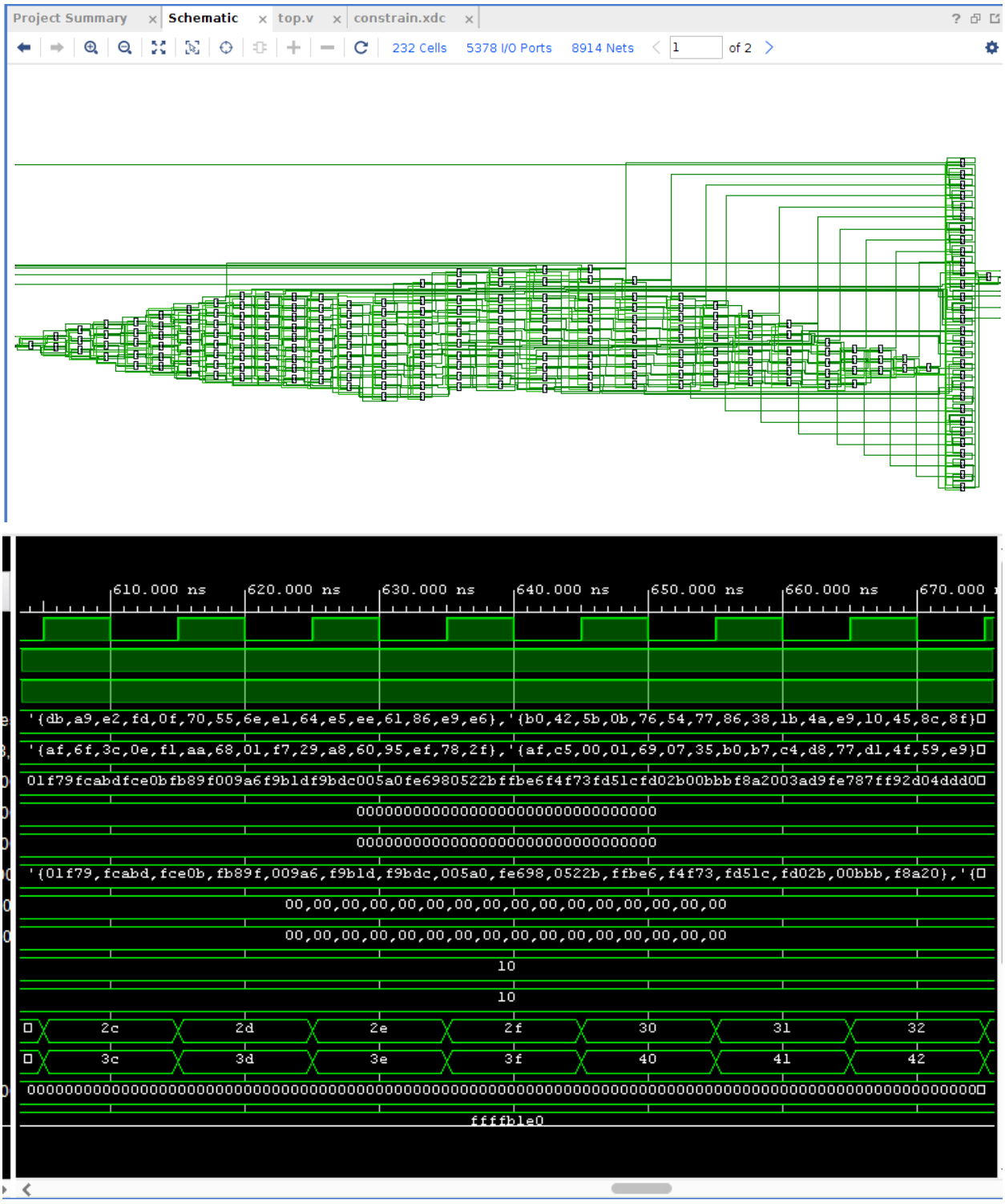
Figure 4: Example 3×3 Systolic Array

2.0.1 Implementation : Using Xilinx Vivado Software (Use Zynq-7000 xc7z007sclg225-2 FPGA)

Consider only 8-bit arithmetic. All the values are already stored in the BRAM.
You must implement a scalable design.

1. Implement a **Systolic Array** design in Verilog, which takes two $n \times n$ matrices and multiply them together.
2. Write a testbench that takes two 16×16 matrices to the design and provide an output of 16×16.
3. Simulate the design using the Vivado simulator and include the waveform in the report (clearly indicate the locations where final outputs produced).
4. Elaborate the design for 16×16 and include all the schematic screenshots of the modules in the report.
5. Synthesis the design and include the schematics screenshots in the report.
6. Generate Resource and timing estimations and include them in the report.
7. Generate power estimation reports and include them in the report.
8. Similarly provides reports for 32×32 matrix multiplication.

A. 16*16 Systolic Array
   1. Schematic Implementation & Function Test
      Here we randomly choose number from 0-128, and generate outcome as a txt format in testbench, after that, we can compare the outcome value by writing just a simple python.


Project Summary
Schematic
top.v
constrain.xdc
232 Cells
5378 I/O Ports
8914 Nets
1
of 2
610.000 ns
620.000 ns
630.000 ns
640.000 ns
650.000 ns
660.000 ns
'{db,a9,e2,fd,0f,70,55,6e,e1,64,e5,ee,61,86,e9,e6},'{b0,42,5b,0b,76,54,77,86,38,1b,4a,e9,10,45,8c,8f}
'{af,6f,3c,0e,f1,aa,68,01,f7,29,a8,60,95,ef,78,2f},'{af,c5,00,01,69,07,35,b0,b7,c4,d8,77,d1,4f,59,e9}
01f79fcabdfce0bfb89f009a6f9b1df9bdc005a0fe6980522bffbe6f4f73fd51cfd02b00bbbf8a2003ad9fe787ff92d04ddd00
00000000000000000000000000000000
00000000000000000000000000000000
'{01f79,fcabd,fce0b,fb89f,009a6,f9b1d,f9bdc,005a0,fe698,0522b,ffbe6,f4f73,fd51c,fd02b,00bbb,f8a20},'{
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
10
10
2c
2d
2e
2f
30
31
32
3c
3d
3e
3f
40
41
42
ffffble0

Matrix_Result.txt - Notepad

File Edit Format View Help

```
  3114  -10640  -27926  -13544  -16209   -1263   12293  -33605    7028   -3902   12368   -3678    5507  -18234     992   20080
 10920  -22488  -33234  -31020  -35348   -7150   26906   15509    2755  -53529   34111  -21830    5012   11705    1524    3324
-17906   40885    9855  -18866    3021  -25479    9385  -32502    3997   21070  -38247    -941   19045   -7646  -11898  -13227
 27949   51370   -5816   -3119   26269  -26682   12086   12738  -11100  -41620  -13960  -23479  -11838   50478   -9924   -1964
 18809   56296  -10850   17478   29892  -26402  -26188    4757   14483   -2820  -50983  -25427     848   39672   17274   21739
 31847    1790    4919    1981   33137   -5055   13080   11359  -28525  -23813   35856   13710  -21713   25215   -8073  -18945
-36227    4776   -6294  -11310    8916   17916  -16025    8769   13360   26046  -35808   23550    6674  -26450   19016   22804
 12125    8648   -7393   22332   15958  -23286  -37675  -27645  -27212  -39425  -18406      55  -18551   21627  -35294   32473
-31557    2750  -25527  -12004  -46575    2442  -13473      10   34352   13280  -49281  -14723   17137  -16335   16825   12036
-13391   16883   22203   24910   11320   26897    1131  -27146    2056   18480   -9227    2309   -4817  -17997   -5820    5112
 -4970   -3988   -6161   43121   25463   38745    2804   12475    4426   12924  -14788   14585   10028  -18794    3846   21449
 12333  -41526   13528   22571  -18082   -9926   -9863   13394  -11995    8536   26373   40363    4070  -10070    1819   34060
 19176  -17637   18361    8911    4845    5052    1498   -3612   -4438   25221    8006    4528    8635     587   18941   -7739
 26506  -22704   14955  -15880   14619  -20331  -15743   -4646  -26791    9559   56665   31503  -24687    8602    3674   -5321
 21499   11313   13196   32578   -3418    3982   16266   -4968   -2656   -5712    8891   46261   19933   -1747   -6265   15065
-30176    3003  -12245  -10980  -45197   -1050   21035   -6504    1440  -25636  -25827    2470  -18273  -12789  -13635    8057
```

Matrix_A.txt - Notepad

File Edit Format View Help

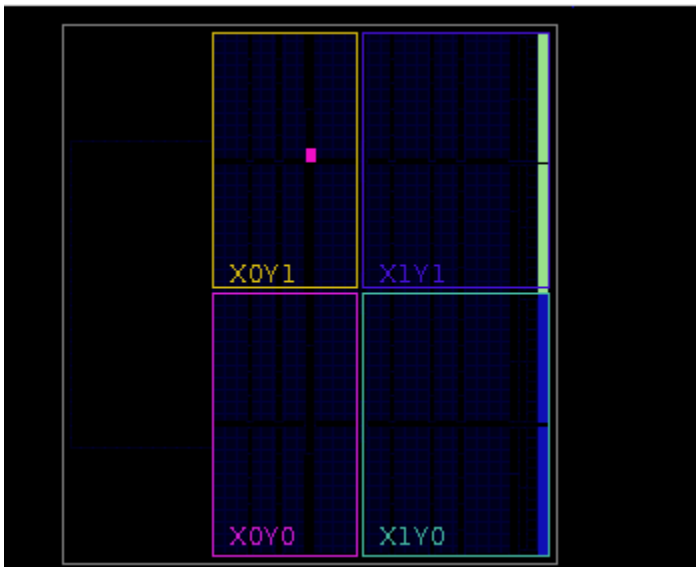
```
  36 -119   13  -27    1  118  109  121  -59  101 -110  114  -24   92  -83  -29
   0   42 -106 -115  107 -126   29   35  -54  114   65  120  107   70  -68 -117
   5  -69   -2  113   98   31  120  -97  -37  -55  -41   22  -62   -9   18  109
  31    5   91  -65   88 -114  122  115  -81  -33  119  -26   41  -38  -75  121
 -48   43   92   -3   86  -25   54  121   20 -124   91   89  -10   54  -58  119
 -76   40  -57 -120   -3 -100  -38  102  -13   94   85  -71   22   38  125    6
 126  -49   -6   23    6  117  -87  -59   75  -20   78   41   14  -97   42   30
  -7   74  107   54   68   18  127  122   50    4   74  -95   -5  -17   54 -113
   8   27   40   75  -98  108   81  -63   88  -42  -30  115   18  -71  -95    1
  39  -97   88   20 -106   85   43   45    4  -25  -55   73  -95  122    3  -14
 -24    6   56  -88  -59  116  -80   42   21   23   73   37 -115    7   12  -69
  -9  -36  -96  114   93   75  -98   -5 -125   29   68  -32   82   13    4 -116
  23    4  124   58  -90   50  -55  -67  -36    6  -76  -84    7   17  -53  108
  36  -71  -44   29    2  -39 -100   99  -56   58  -85 -110   89  -18   91   59
-113 -116   69   16  -23   74   27   56 -122  119   84  118   11   91   66  -80
 -26  -23 -122   97  -18  -27  100  -31  110   85  112   15   -3  -30  -87  -37
```

Matrix_B.txt - Notepad

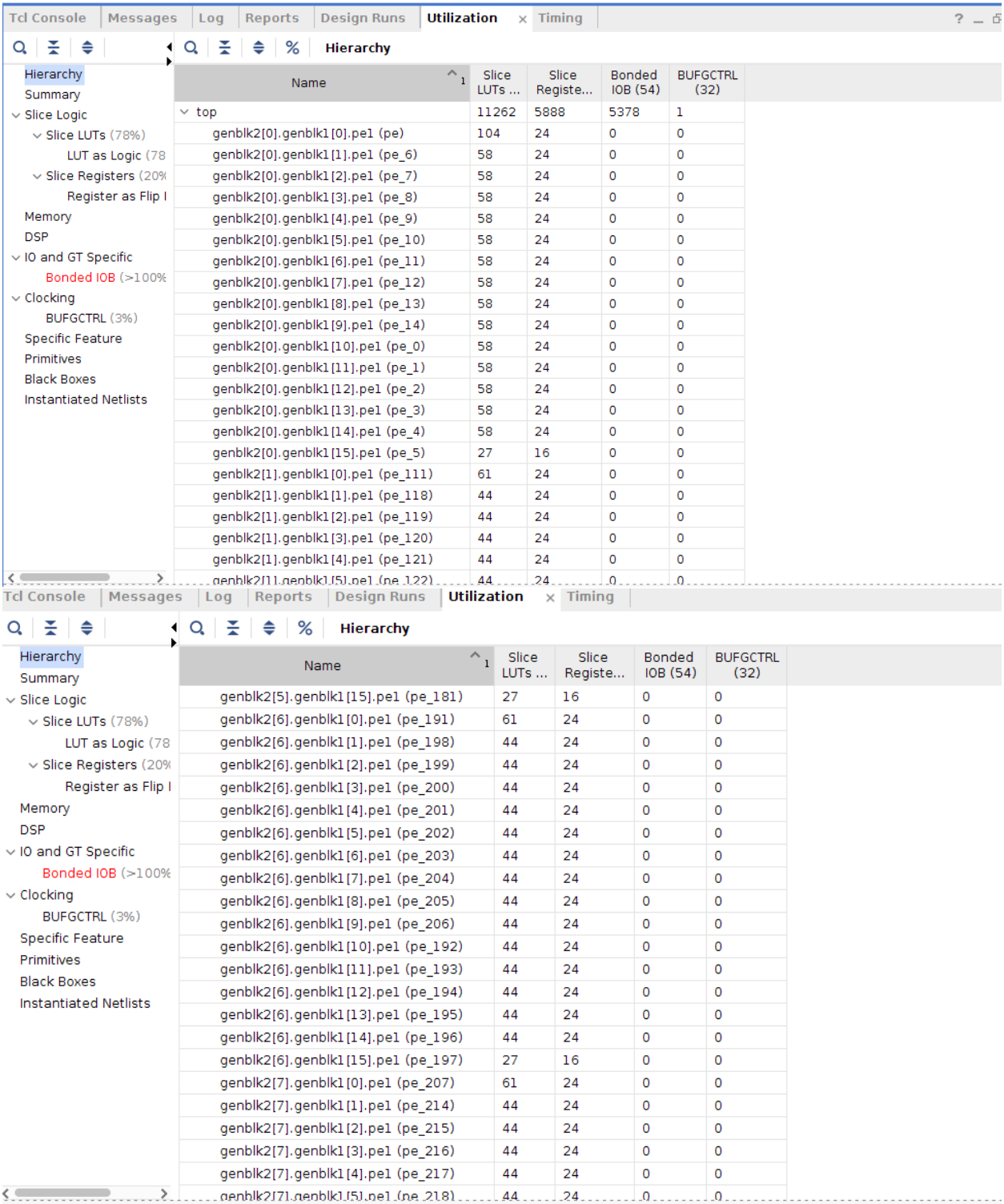
File Edit Format View Help

```
-127  -29   13 -110   13   61   12  -58  -86   -9   15  -50  -59  -67  -27   10
  32  -99 -109  -45  -43  -82  -49   10  -68   10  -40 -119   54   46   42  -15
  79   58   21   89   76   15  -73  -36    9  -48  -47   12   72   61   -2   57
 -45   -8   73   42 -122 -100  -90  -93  -77   68  -53   90  -19  -27  -33   68
  42   14 -102  -61   78 -118  -72  -72   19  -39  -51  -19   74   21    4  105
   8  -83  -82   28   41    6  -67  -16   58  122 -102   55   64  -74   92  120
 -37  121  -31  -95  -48  -75   65 -104  -13 -118  -88  -95  -26   42 -115  -72
  72   19  -57   58   57  -76 -122  -14  -67  -28   41   14 -117   73  -11  107
 -82   18  -83  -62   13   24 -122  -69   83  -37 -124  -40  -72  -27   43   72
  33 -124 -106 -110  -79  -19  117  -89  -89  -71   91   81   42  -59   -4   -2
 -17  112   64  118   64   57   60   98  -31  -61 -122   65  -38   44  -15   54
 -98   85  -96  -76 -115   -7   -3   15   99   49 -107  -19  120  -46   70   16
 106  -86  -32 -117  -91  -94 -108   75   71  -71   80   12  -22  104  115   35
  73  -33  103   48   73   32   19  -86  -35    4   57  -46  -48   23   38    0
 -23   89   79  -47  119  -40  -60  -73  -80   53    7  105    1    0  -59  -81
  47  120  -17 -107   96  -88   41   -9    1  104  -86  -15   14   60  111  -81
```

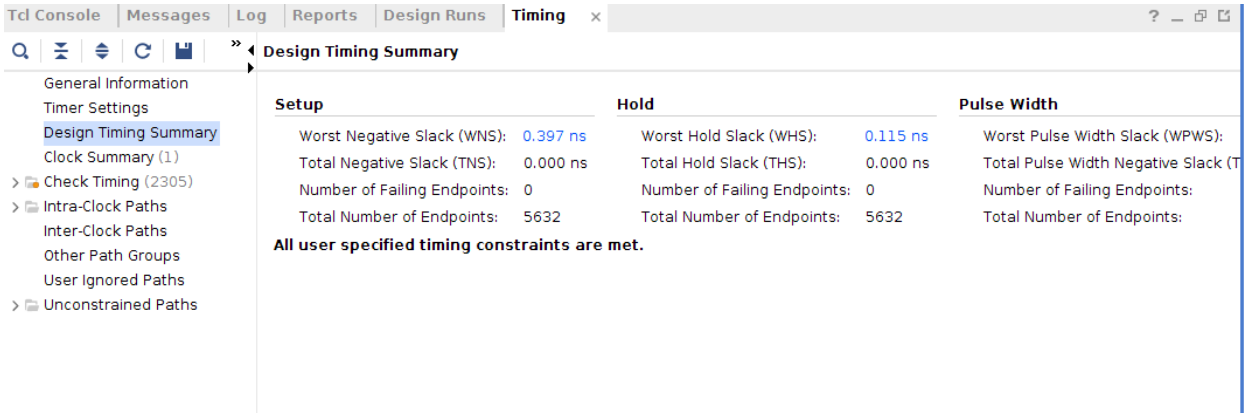
2. Synthesis schematic



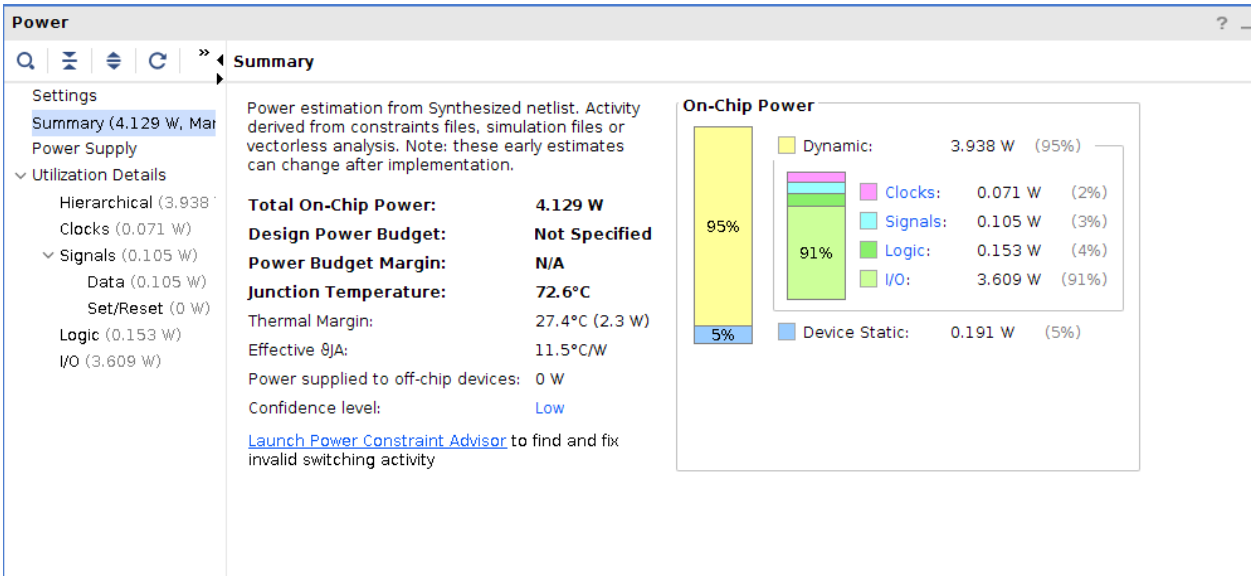
3. Resources Utilization



4. Timing Report

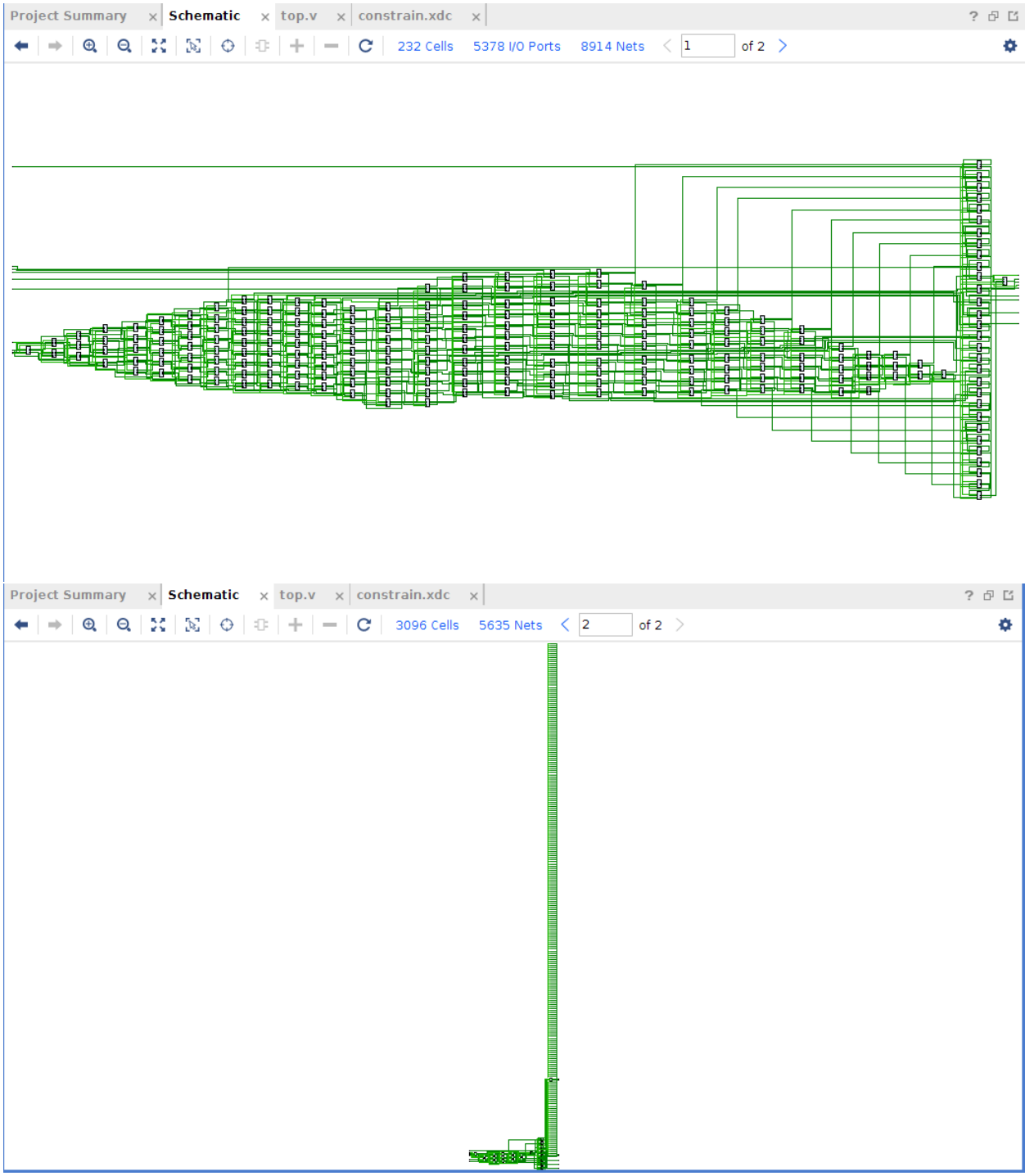


5. Power Report

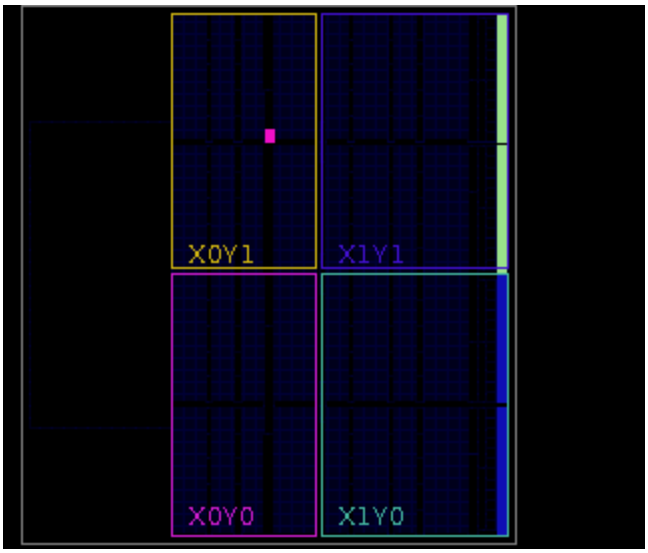


B. 32*32 16*16 Systolic Array

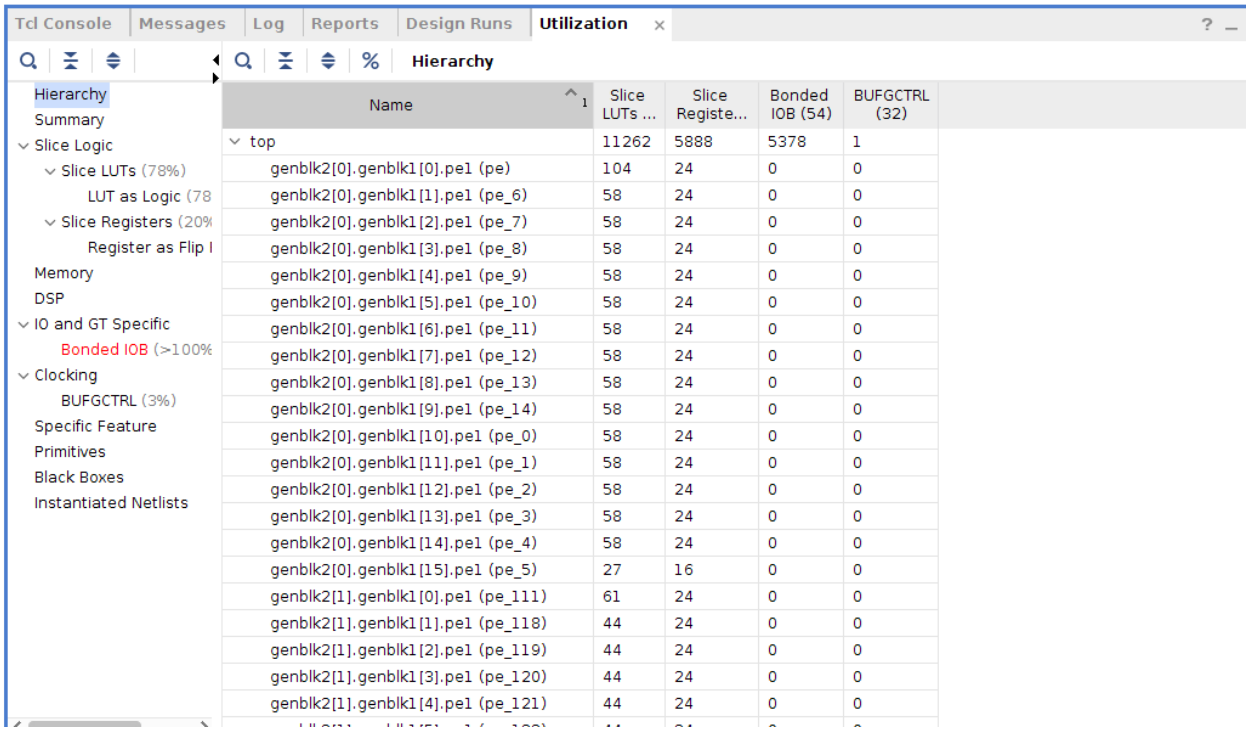
1. Schematic and Function Wave form

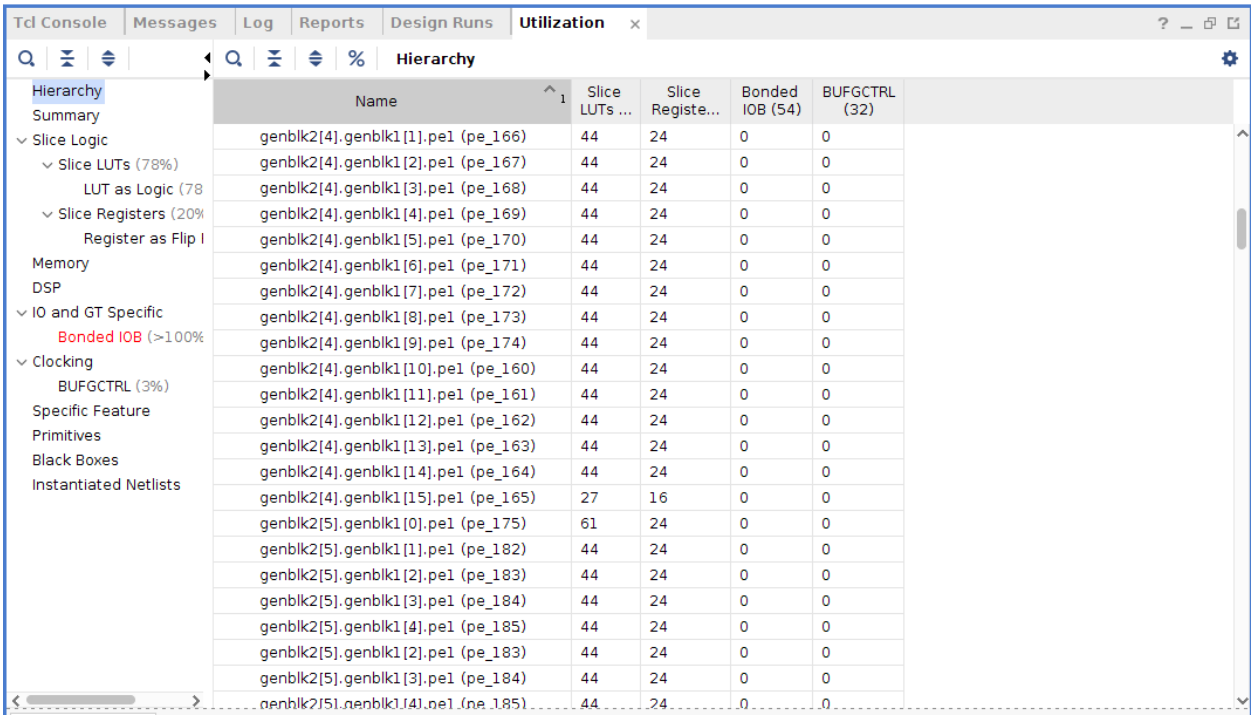


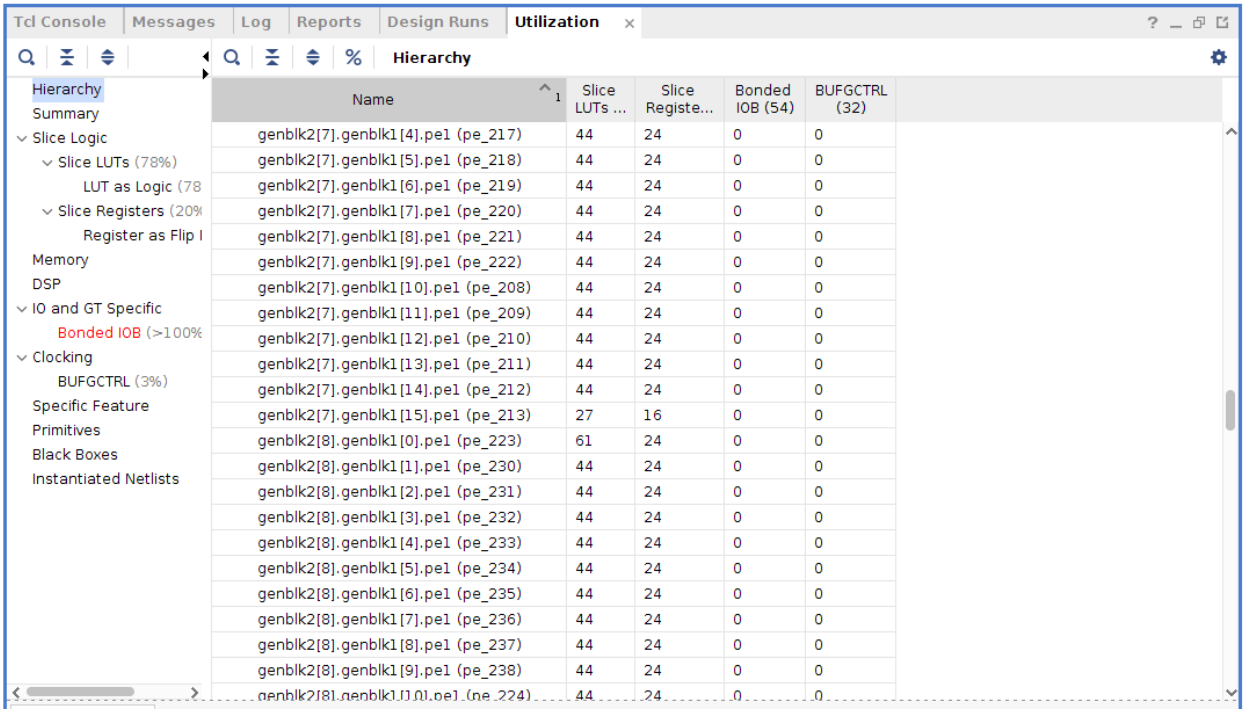
2. Synthesis schematic



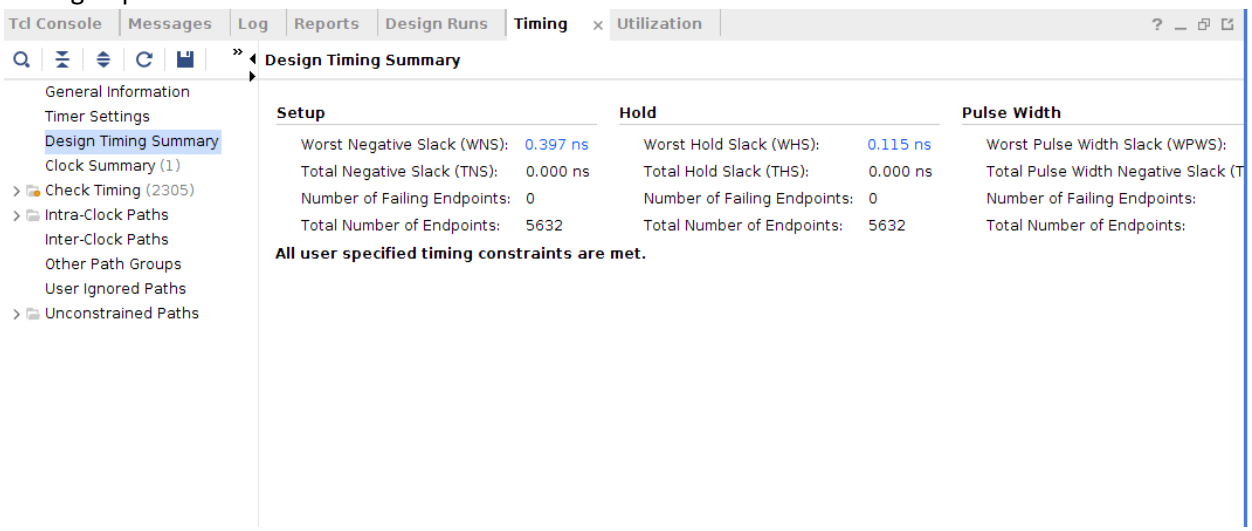
3. Resources Utilization


Tcl Console
Messages
Log
Reports
Design Runs
Utilization
Hierarchy
Hierarchy
Summary
Slice Logic
Slice LUTs (78%)
LUT as Logic (78
Slice Registers (20%
Register as Flip I
Memory
DSP
IO and GT Specific
Bonded IOB (>100%
Clocking
BUFGCTRL (3%)
Specific Feature
Primitives
Black Boxes
Instantiated Netlists
Name
Slice LUTs ...
Slice Registe...
Bonded IOB (54)
BUFGCTRL (32)
top 11262 5888 5378 1
genblk2[0].genblk1[0].pe1 (pe) 104 24 0 0
genblk2[0].genblk1[1].pe1 (pe_6) 58 24 0 0
genblk2[0].genblk1[2].pe1 (pe_7) 58 24 0 0
genblk2[0].genblk1[3].pe1 (pe_8) 58 24 0 0
genblk2[0].genblk1[4].pe1 (pe_9) 58 24 0 0
genblk2[0].genblk1[5].pe1 (pe_10) 58 24 0 0
genblk2[0].genblk1[6].pe1 (pe_11) 58 24 0 0
genblk2[0].genblk1[7].pe1 (pe_12) 58 24 0 0
genblk2[0].genblk1[8].pe1 (pe_13) 58 24 0 0
genblk2[0].genblk1[9].pe1 (pe_14) 58 24 0 0
genblk2[0].genblk1[10].pe1 (pe_0) 58 24 0 0
genblk2[0].genblk1[11].pe1 (pe_1) 58 24 0 0
genblk2[0].genblk1[12].pe1 (pe_2) 58 24 0 0
genblk2[0].genblk1[13].pe1 (pe_3) 58 24 0 0
genblk2[0].genblk1[14].pe1 (pe_4) 58 24 0 0
genblk2[0].genblk1[15].pe1 (pe_5) 27 16 0 0
genblk2[1].genblk1[0].pe1 (pe_111) 61 24 0 0
genblk2[1].genblk1[1].pe1 (pe_118) 44 24 0 0
genblk2[1].genblk1[2].pe1 (pe_119) 44 24 0 0
genblk2[1].genblk1[3].pe1 (pe_120) 44 24 0 0
genblk2[1].genblk1[4].pe1 (pe_121) 44 24 0 0

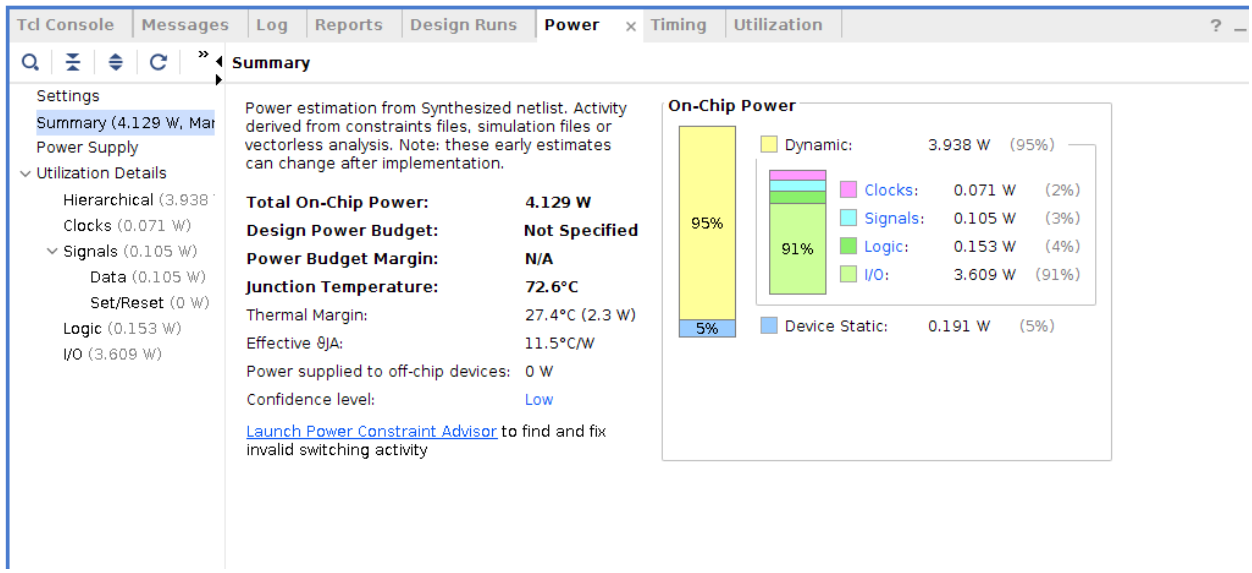

Tcl Console
Messages
Log
Reports
Design Runs
Utilization
Hierarchy
Hierarchy
Summary
Slice Logic
Slice LUTs (78%)
LUT as Logic (78
Slice Registers (20%
Register as Flip I
Memory
DSP
IO and GT Specific
Bonded IOB (>100%
Clocking
BUFGCTRL (3%)
Specific Feature
Primitives
Black Boxes
Instantiated Netlists
Name
Slice LUTs ...
Slice Registe...
Bonded IOB (54)
BUFGCTRL (32)
genblk2[4].genblk1[1].pe1 (pe_166) 44 24 0 0
genblk2[4].genblk1[2].pe1 (pe_167) 44 24 0 0
genblk2[4].genblk1[3].pe1 (pe_168) 44 24 0 0
genblk2[4].genblk1[4].pe1 (pe_169) 44 24 0 0
genblk2[4].genblk1[5].pe1 (pe_170) 44 24 0 0
genblk2[4].genblk1[6].pe1 (pe_171) 44 24 0 0
genblk2[4].genblk1[7].pe1 (pe_172) 44 24 0 0
genblk2[4].genblk1[8].pe1 (pe_173) 44 24 0 0
genblk2[4].genblk1[9].pe1 (pe_174) 44 24 0 0
genblk2[4].genblk1[10].pe1 (pe_160) 44 24 0 0
genblk2[4].genblk1[11].pe1 (pe_161) 44 24 0 0
genblk2[4].genblk1[12].pe1 (pe_162) 44 24 0 0
genblk2[4].genblk1[13].pe1 (pe_163) 44 24 0 0
genblk2[4].genblk1[14].pe1 (pe_164) 44 24 0 0
genblk2[4].genblk1[15].pe1 (pe_165) 27 16 0 0
genblk2[5].genblk1[0].pe1 (pe_175) 61 24 0 0
genblk2[5].genblk1[1].pe1 (pe_182) 44 24 0 0
genblk2[5].genblk1[2].pe1 (pe_183) 44 24 0 0
genblk2[5].genblk1[3].pe1 (pe_184) 44 24 0 0
genblk2[5].genblk1[4].pe1 (pe_185) 44 24 0 0
genblk2[5].genblk1[2].pe1 (pe_183) 44 24 0 0
genblk2[5].genblk1[3].pe1 (pe_184) 44 24 0 0



4. Timing Report



5. Power Report


Tcl Console
Messages
Log
Reports
Design Runs
Power
Timing
Utilization
Summary
Settings
Summary (4.129 W, Mar
Power Supply
Utilization Details
Hierarchical (3.938 W)
Clocks (0.071 W)
Signals (0.105 W)
Data (0.105 W)
Set/Reset (0 W)
Logic (0.153 W)
I/O (3.609 W)
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.
Total On-Chip Power: 4.129 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 72.6°C
Thermal Margin: 27.4°C (2.3 W)
Effective θJA: 11.5°C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low
Launch Power Constraint Advisor to find and fix invalid switching activity
On-Chip Power
95%
5%
91%
Dynamic: 3.938 W (95%)
Clocks: 0.071 W (2%)
Signals: 0.105 W (3%)
Logic: 0.153 W (4%)
I/O: 3.609 W (91%)
Device Static: 0.191 W (5%)