ESE 507 | PROJECT 3

HARDWARE GENERATION TOOL

Aswin Natesh Venkatesh & SBU ID: 111582677
Gosakan Srinivasan SBU ID: 111579886

Evaluation Report - Part 1-3

QUESTION 1:

The hardware generation tool was designed in a generic way that supports scalability and flexibility. The scaling of matrix size and number of bits is handled by the System Verilog(SV) script while the pipeline is handled by the generator tool. The SV script design incorporates parameterized modules that scale the design for the required matrix size and number of bits. The values are passed directly from the user to the SV Script. The hardware generator tool structures data-path of the SV script for required amount of pipelining. The generator tool also breaks down the input value stream to smaller chunks and stores them in separate ROMS/memory location. The SV script Control path and MAC modules are designed in a highly generic manner supporting any given matrix size, bit size and degree of parallelism. There are no upper bound limits of M and N values as such and the only limiting factor that decides this is the variable size in both the C and SV Script. Moreover, it is observed that passing very huge M and N values greater than 64 consumes tremendous time for simulation and synthesis.

QUESTION 2:

The control module uses a Finite State Machine (FSM) consisting of 4 discrete states. Each state is assigned a specific set of instructions to be carried out. The FSM keeps track of this and transit states upon successful completion of operations by individual states. The 4 states are listed below in the table.

State No	*Tasks Performed*	*Transition Signal*
State 0	**Idle State** | Wait for Input Valid Signal & Data.	S_Valid == 1
State 1	**Data Input** | Handshake and receive Input Data Values [M & X]	Write_complete == 1
State 2	**Matrix Vector Multiplication** | Feed Data Input to MAC Module, obtain computed values and store them to memory.	Mac_complete == 1
State 3	**Display State** | Handshake and send results out.	Data_displayed == 1

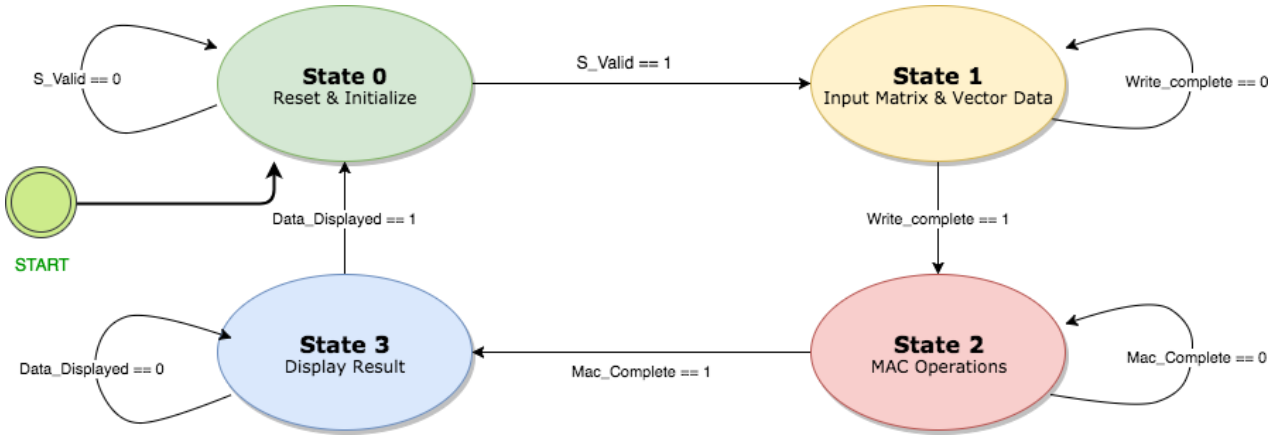


Figure 1: FSM State Transition Diagram

The control modules involve various counters to assert flag signals and write signals. This module also generates the addresses for read/write operations from memory and effectively controls and directs data in/out from in the data path module. The individual state operations are detailed below.

Detailed STATE operations at every positive clock edge: -

STATE 0: Wait for S_Valid signal, Initialize variables and Initiate State 1.

STATE 1: Generate Address values for Input Matrix M, Vector X and corresponding Write Enable signals. Write_complete flag is asserted once data is received completely.

STATE 2: Feed Row wise and Column Wise matrix and vector values into MAC Module, and stores results in Memory Y. Mac_complete flag is asserted once MAC operations are complete. (Address and Write signals are asserted accordingly)

STATE 3: Wait for M_ready signal and display results. Data_displayed flag is asserted once results are displayed and initiate State 0. This completes of one iteration for the given MVM Inputs.

This is the FSM design used in Project 2(Matrix Vector Multiplication) and is used in Project 3 SV Script with no major changes in it. This FSM was generic enough to work with any given matrix size, bit size and degree of parallelism. Therefore, the scaling and flexibility for M, N and T are handled very efficiently by the control module.

QUESTION 3:

For parallelism, the input Matrix elements (W Matrix & Vector B) are split and stored in smaller ROMs according to the Matrix Vector Multiplication traversal sequence by the C Code. Further in the SV Code, P number of MAC modules and Y memories for storing results are instantiated in the data-path module. Doing this simplifies the address and control-path complexity as multiple ROMs can be traversed simultaneously using the same address value. The data outputs from the ROMs are then fed to the respective MAC modules, and outputs are stored sequentially in the Y Memory. Since the total number of elements stored in split ROMs remain same, the memory space occupied is left unchanged. This design architecture simplifies the overall complexity of the SV script and lends itself to scalability at reduced cost.

QUESTION 4:

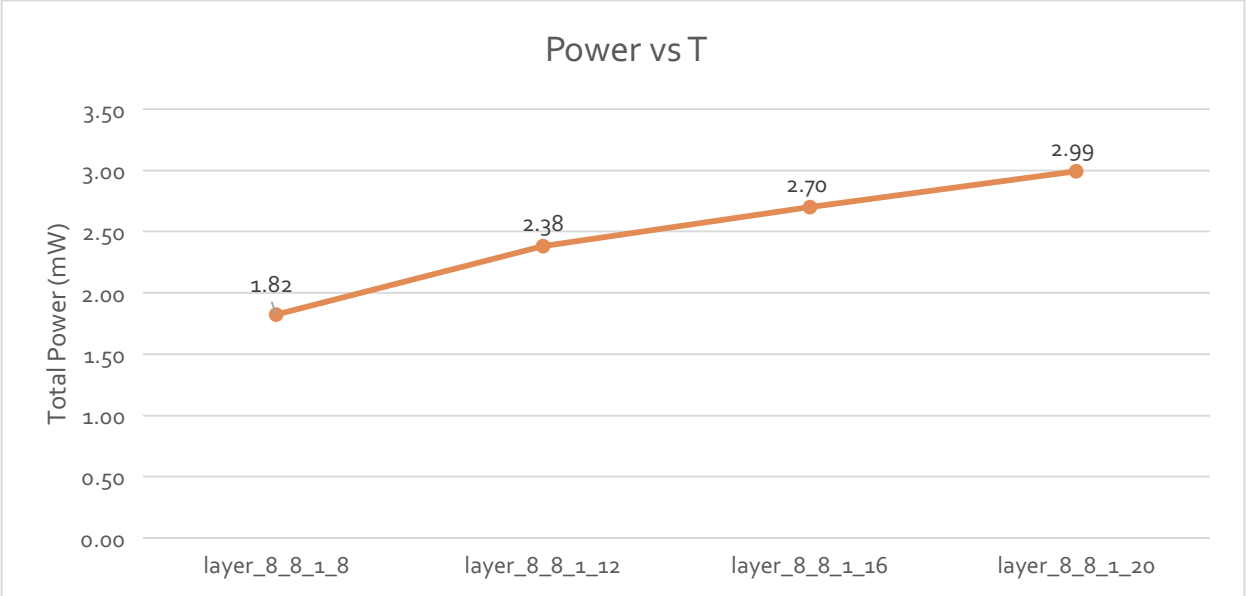
- Increasing M and N values increases the problem size and memory resources required, therefore leads to increase in costs. It however has no/less effect on the precision and performance metrics.
- Increasing P increases the performance of the system and also increases the resources required (Costs)
- Increasing T improves the precision and in turn increases the memory resources required area and power consumed (Costs).

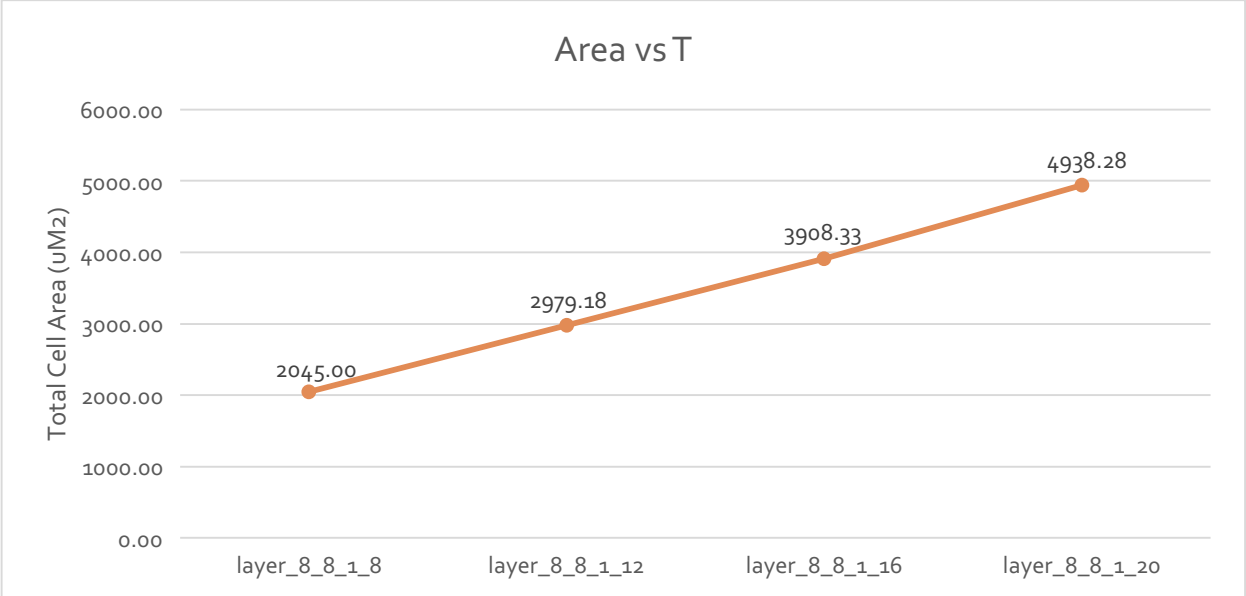
QUESTION 5:

> Performance of the system with variation in T Value

Layer	layer_8_8_1_8	layer_8_8_1_12	layer_8_8_1_16	layer_8_8_1_20
Target Frequency (GHz)	1.190	1.136	0.971	0.877
Clock Period (ns)	0.84	0.88	1.03	1.14
Combinational Area	1010.00	1541.46	2070.54	2694.31
Non Combinational Area (um2)	1035.00	1437.72	1837.79	2243.97
Total Cell Area (um2)	2045.00	2979.18	3908.33	4938.28
Total Dynamic Power (mW)	1.78	2.32	2.62	2.89
Cell Leakage Power (uW)	41.67	62.07	81.05	103.20
Total Power (mW)	1.82	2.38	2.70	2.99
Energy per Cycle Operation (pJ)	1.53	2.10	2.78	3.41
Timing Report (Slack)	0.00	0.00	0.00	0.00
	MET	MET	MET	MET
Critical Part - Start Point	z_reg[0]	z_reg[3]	z_reg[9]	z_reg[3]
Critical Part - End Point	product_reg[7]	product_reg[11]	product_reg[15]	product_reg[19]

> Performance Evaluation Graphs with variation in T values





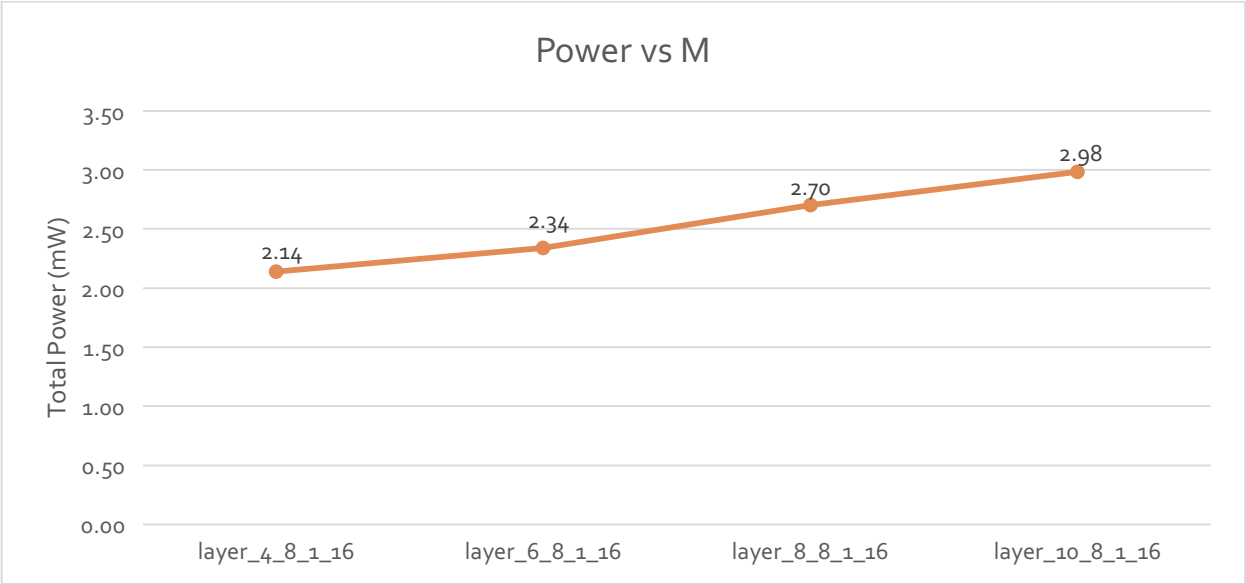
QUESTION 6:

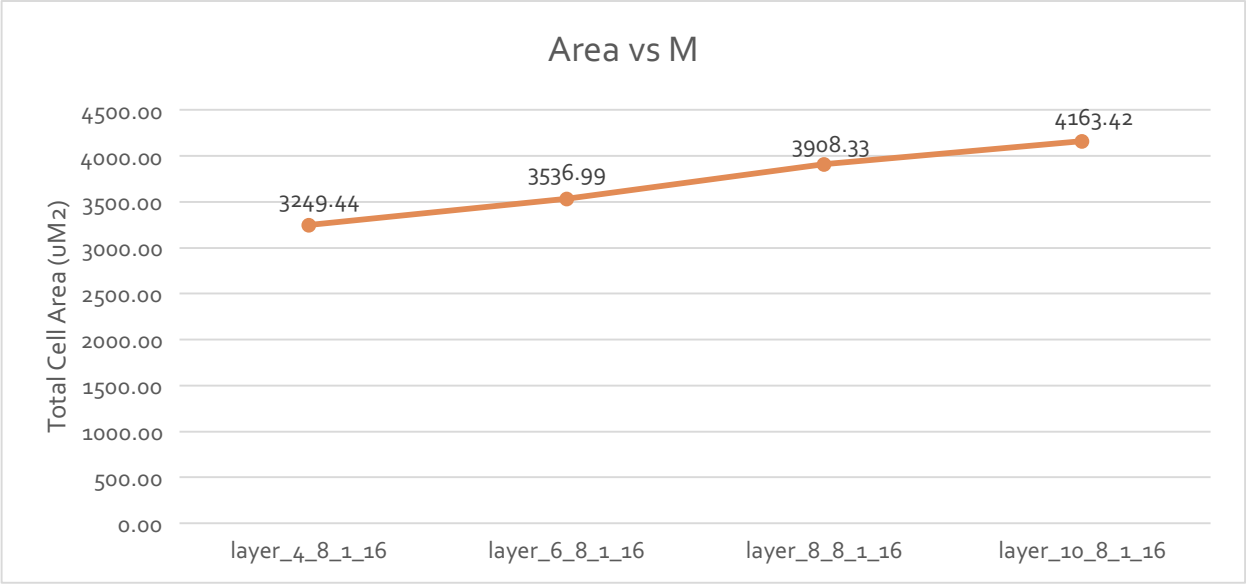
> Performance of the system with variation in M Value

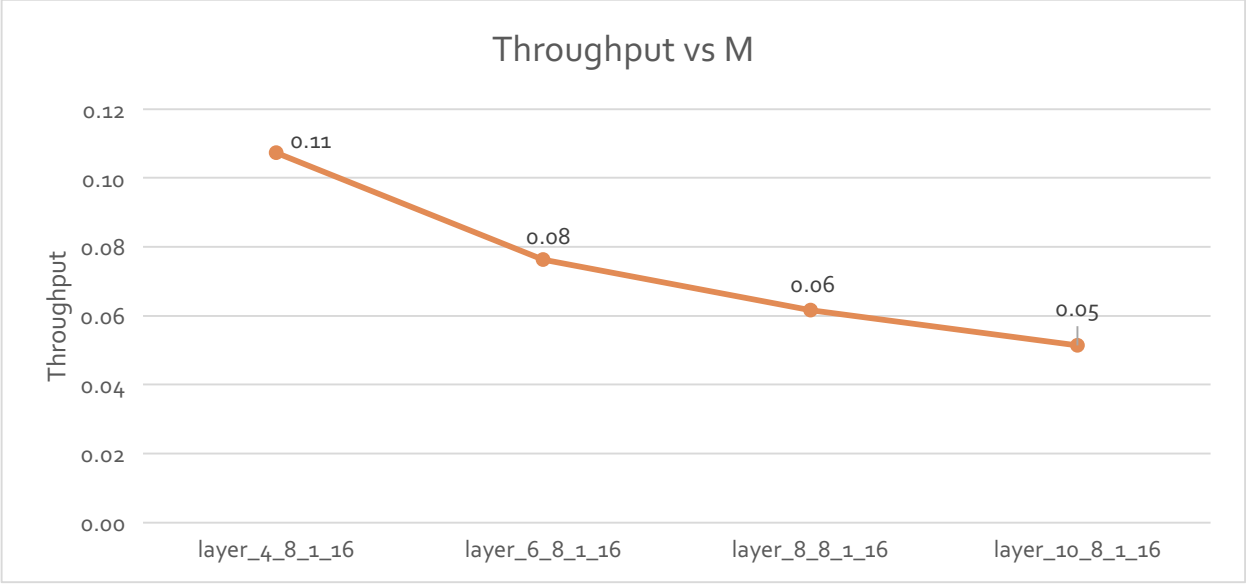
Layer	layer_4_8_1_16	layer_6_8_1_16	layer_8_8_1_16	layer_10_8_1_16
Target Frequency (GHz)	0.952	0.935	0.971	0.990
Clock Period (ns)	1.05	1.07	1.03	1.01
Combinational Area	1732.98	1843.91	2070.54	2146.88
Non Combinational Area (um2)	1516.46	1693.08	1837.79	2016.54
Total Cell Area (um2)	3249.44	3536.99	3908.33	4163.42
Total Dynamic Power (mW)	2.07	2.27	2.62	2.90
Cell Leakage Power (uW)	67.04	70.41	81.05	82.74
Total Power (mW)	2.14	2.34	2.70	2.98
Energy per Cycle Operation (pJ)	2.24	2.50	2.78	3.01
Timing Report (Slack)	0.00	0.00	0.00	0.00
	MET	MET	MET	MET
Critical Part - Start Point	z_reg[9]	z_reg[3]	z_reg[9]	data_y_reg[0]
Critical Part - End Point	product_reg[14]	product_reg[15]	product_reg[15]	data_y_reg[19]
Parameter C x CLK (ns)	710.00	980.00	1260.00	1540.00
Throughput	0.11	0.08	0.06	0.05

Yes, It is observed that the critical path changes with variation in M Value

> Performance Evaluation Graphs with variation in M values







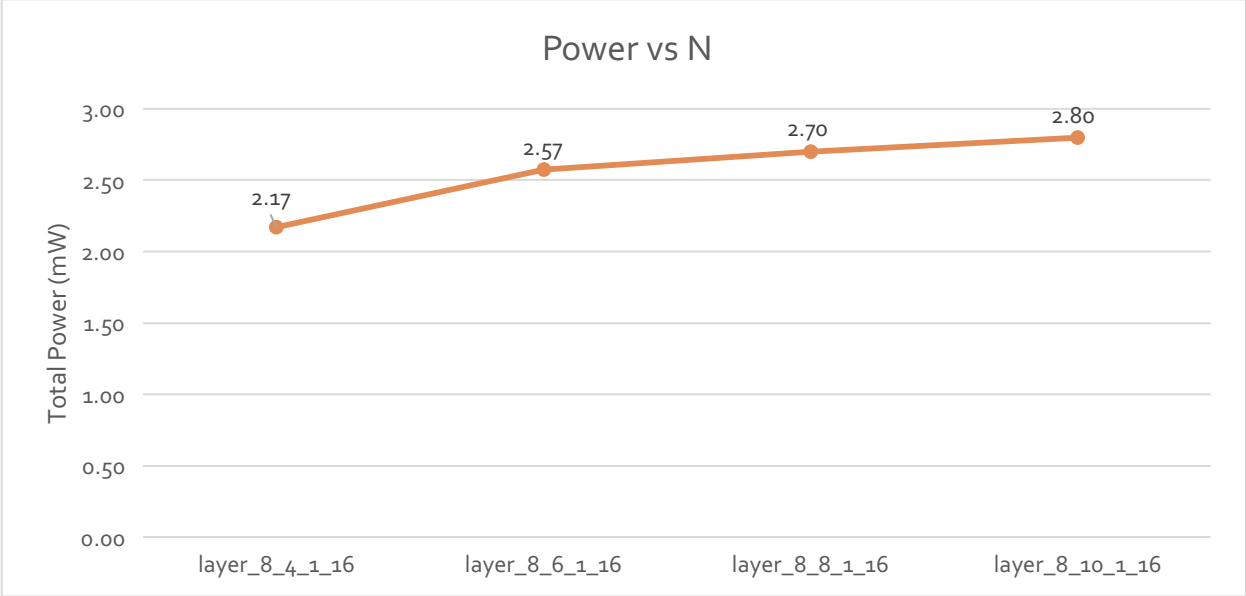
QUESTION 7:

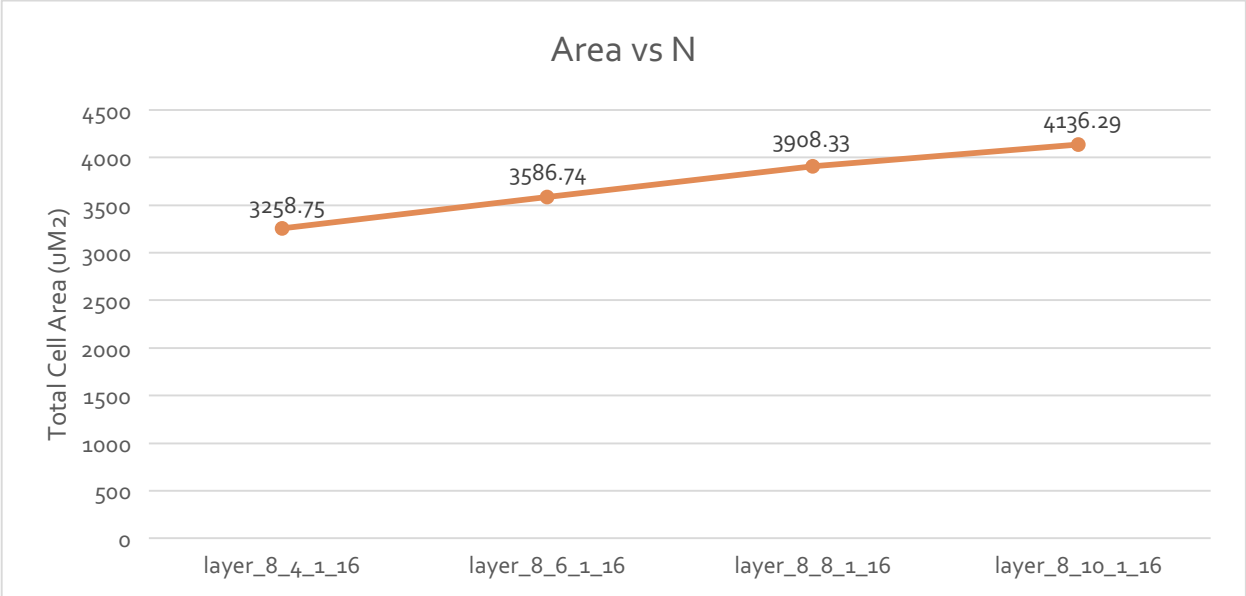
> Performance of the system with variation in N Value

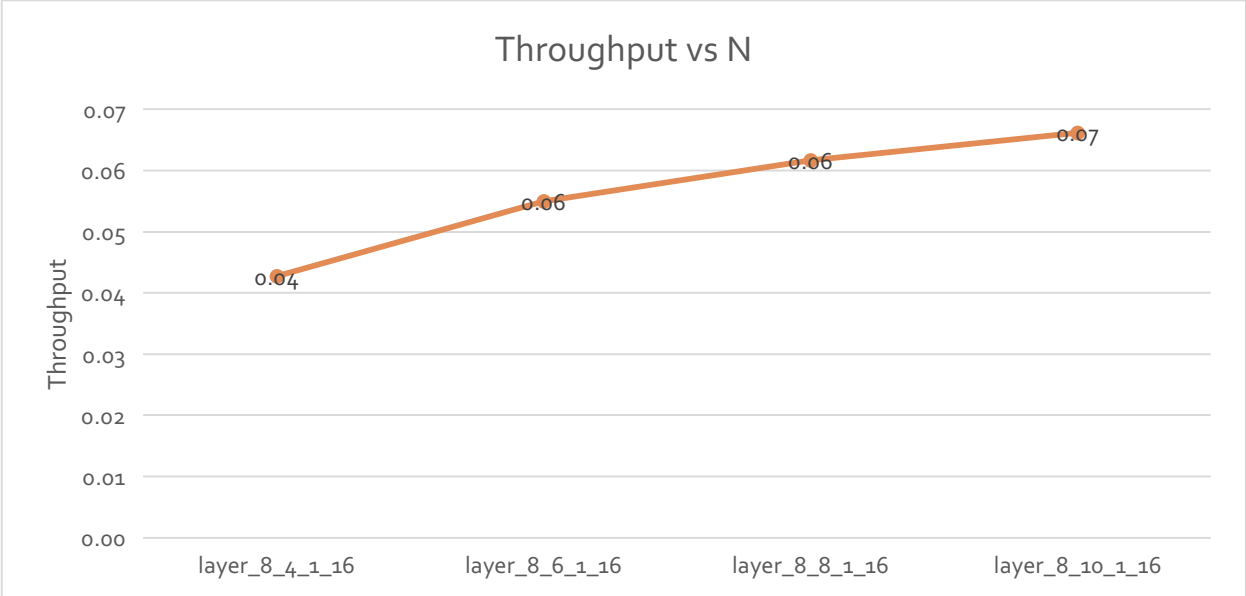
Layer	layer_8_4_1_16	layer_8_6_1_16	layer_8_8_1_16	layer_8_10_1_16
Target Frequency (GHz)	0.962	0.990	0.971	0.952
Clock Period (ns)	1.04	1.01	1.03	1.05
Combinational Area	1733.25	1895.25	2070.54	2121.88
Non Combinational Area (um2)	1525.50	1691.49	1837.79	2014.41
Total Cell Area (um2)	3258.75	3586.74	3908.33	4136.29
Total Dynamic Power (mW)	2.10	2.50	2.62	2.71
Cell Leakage Power (uW)	68.00	72.89	81.05	83.36
Total Power (mW)	2.17	2.57	2.70	2.80
Energy per Cycle Operation (pJ)	2.25	2.60	2.78	2.94
Timing Report (Slack)	0.00	0.00	0.00	0.00
	MET	MET	MET	MET
Critical Part - Start Point	z_reg[9]	z_reg[1]	z_reg[9]	z_reg[1]
Critical Part - End Point	product_reg[15]	product_reg[15]	product_reg[15]	product_reg[15]
Parameter C x CLK (ns)	900.00	1080.00	1260.00	1440.00
Throughput	0.04	0.06	0.06	0.07

Yes, It is observed that the critical path changes with variation in N Value

> Performance Evaluation Graphs with variation in N values




Area vs N
Total Cell Area (uM2)
4500
4000
3500
3000
2500
2000
1500
1000
500
0
3258.75
3586.74
3908.33
4136.29
layer_8_4_1_16
layer_8_6_1_16
layer_8_8_1_16
layer_8_10_1_16


Throughput vs N
Throughput
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0.04
0.06
0.06
0.07
layer_8_4_1_16
layer_8_6_1_16
layer_8_8_1_16
layer_8_10_1_16

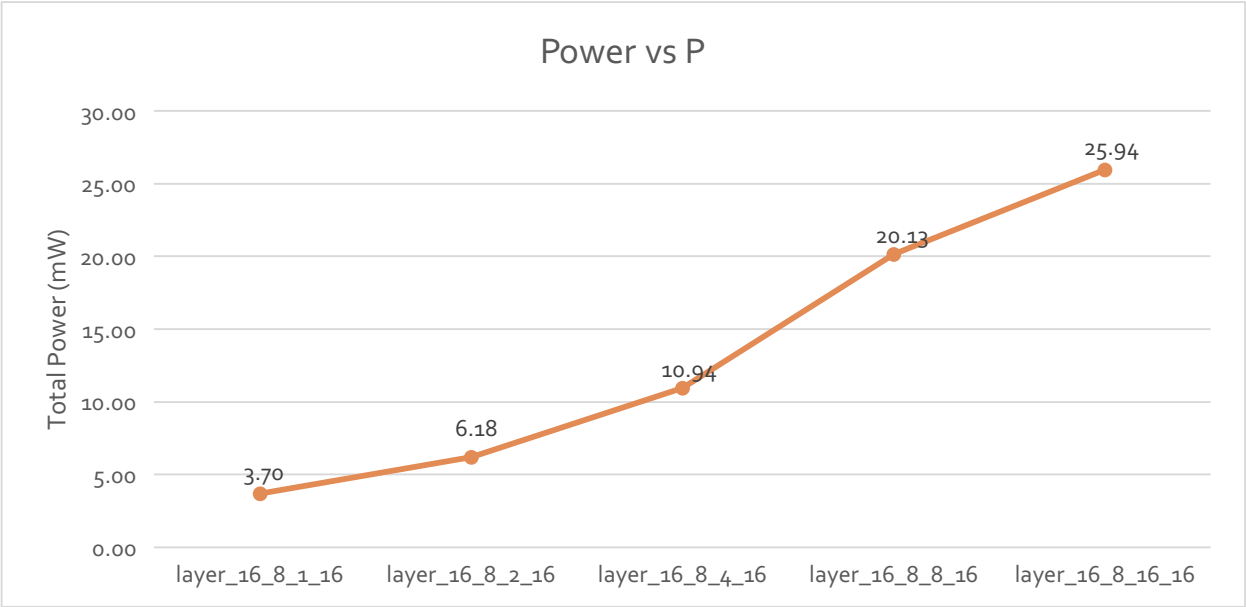
QUESTION 8:

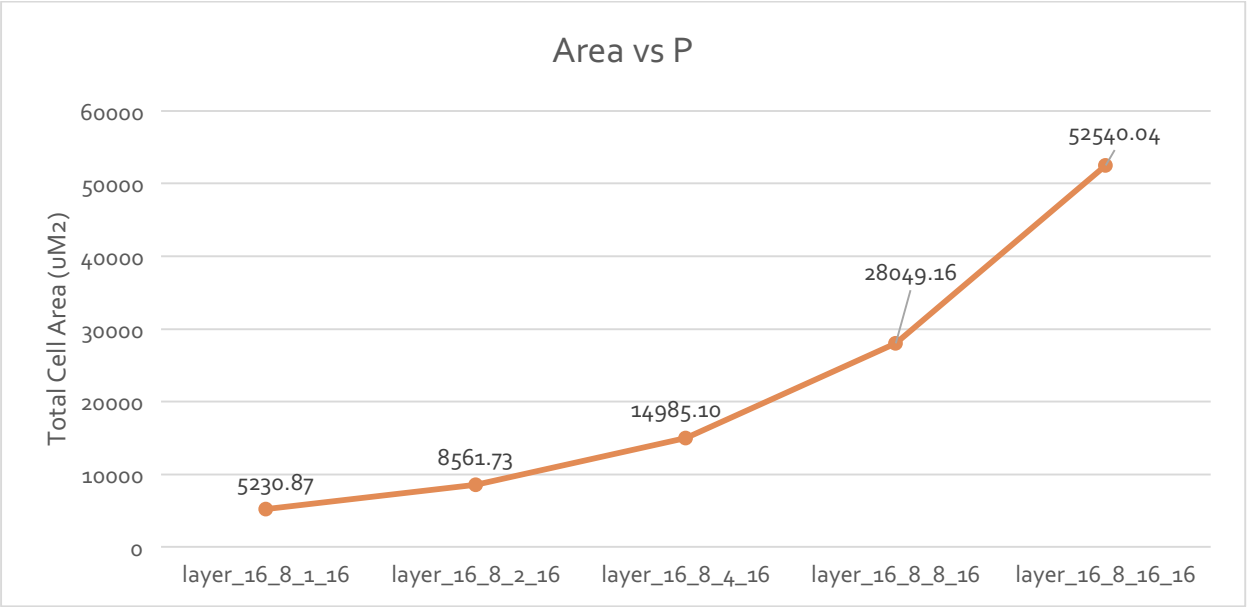
> Performance of the system with variation in P Value

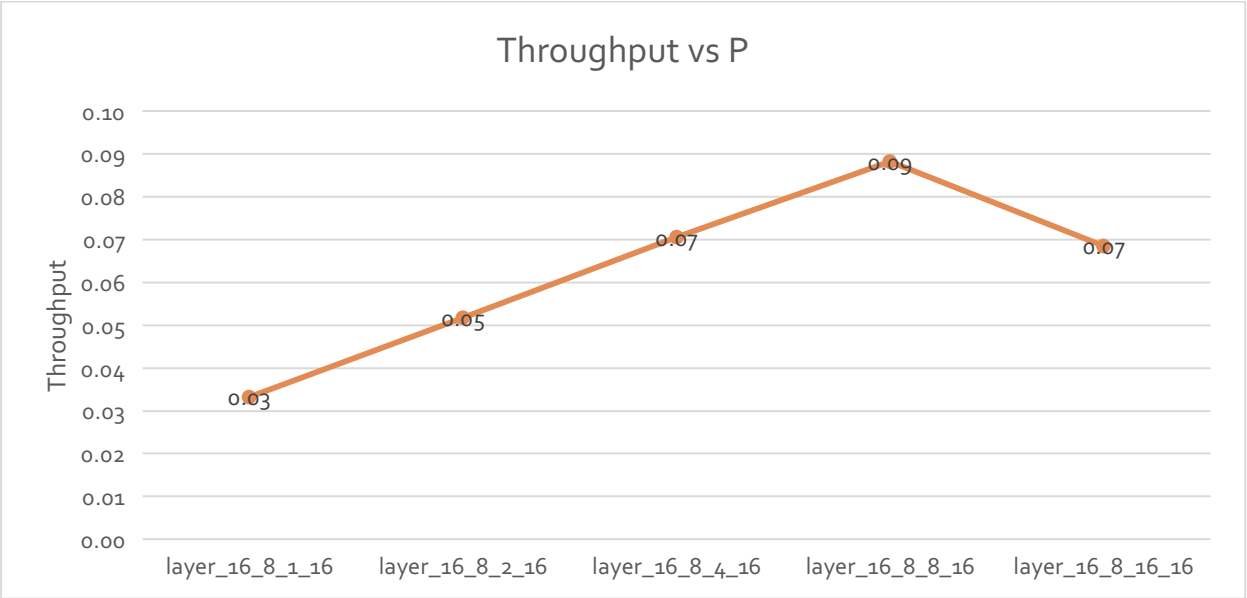
Layer	layer_16_8_1_16	layer_16_8_2_16	layer_16_8_4_16	layer_16_8_8_16	layer_16_8_16_16
Target Frequency (GHz)	0.990	0.971	0.935	0.926	0.625
Clock Period (ns)	1.01	1.03	1.07	1.08	1.60
Combinational Area	2783.68	4537.16	7968.82	15079.27	28046.77
Non Combinational Area (um2)	2447.19	4024.57	7016.28	12969.89	24493.27
Total Cell Area (um2)	5230.87	8561.73	14985.10	28049.16	52540.04
Total Dynamic Power (mW)	3.59	6.00	10.63	19.54	24.86
Cell Leakage Power (uW)	110.23	179.22	312.18	585.70	1083.30
Total Power (mW)	3.70	6.18	10.94	20.13	25.94
Energy per Cycle Operation (pJ)	3.74	6.36	11.71	21.74	41.51
Timing Report (Slack)	0.00	0.00	0.00	0.00	0.00
	MET	MET	MET	MET	MET
Critical Part - Start Point	z_reg[3]	z_reg[1]	z_reg[5]	z_reg[3]	data_out_reg[1]
Critical Part - End Point	product_reg[15]	product_reg[15]	product_reg[15]	product_reg[15]	product_reg[15]
Parameter C x CLK (ns)	2380.00	1500.00	1060.00	840.00	730.00
Throughput	0.03	0.05	0.07	0.09	0.07

It is observed that **layer_16_8_4_16** reaches the highest frequency at a very optimal power and a competitive throughput value per second. It is observed that critical path changes with variation in P Value

> Performance Evaluation Graphs with variation in P values







QUESTION 9:

The design can be parallelised further by increasing the number of multipliers and adders to multiply and accumulate single row and column. That is instead of using a single MAC unit to compute one output y[0] value, we can use several multipliers and adders to compute multiple y values simultaneously. Another approach would be to split the increase existing multiplier layer(N) into sub-multiplier layers (N1, N2...Nn) and simultaneously computing the outputs for each layer.

QUESTION 10:

Optimizing parallelism mandates the need to satisfy minimum values for P1, P2 and P3 which is 1 respectively and the highest value of any being less than (M1+M2+M3). which is the multiplier budget. With these constraints in hand, the optimal parallelism parameters are modelled as given in expression [1]. Wherein the least value for each parameter is computed by iterating for which the final value satisfies the condition that M/P is an Integer.

$$Min(Sum) = \left[\frac{N \times M1}{P1}\right] + \left[\frac{M2 \times M1}{P2}\right] + \left[\frac{M3 \times M2}{P3}\right] \qquad [1]$$

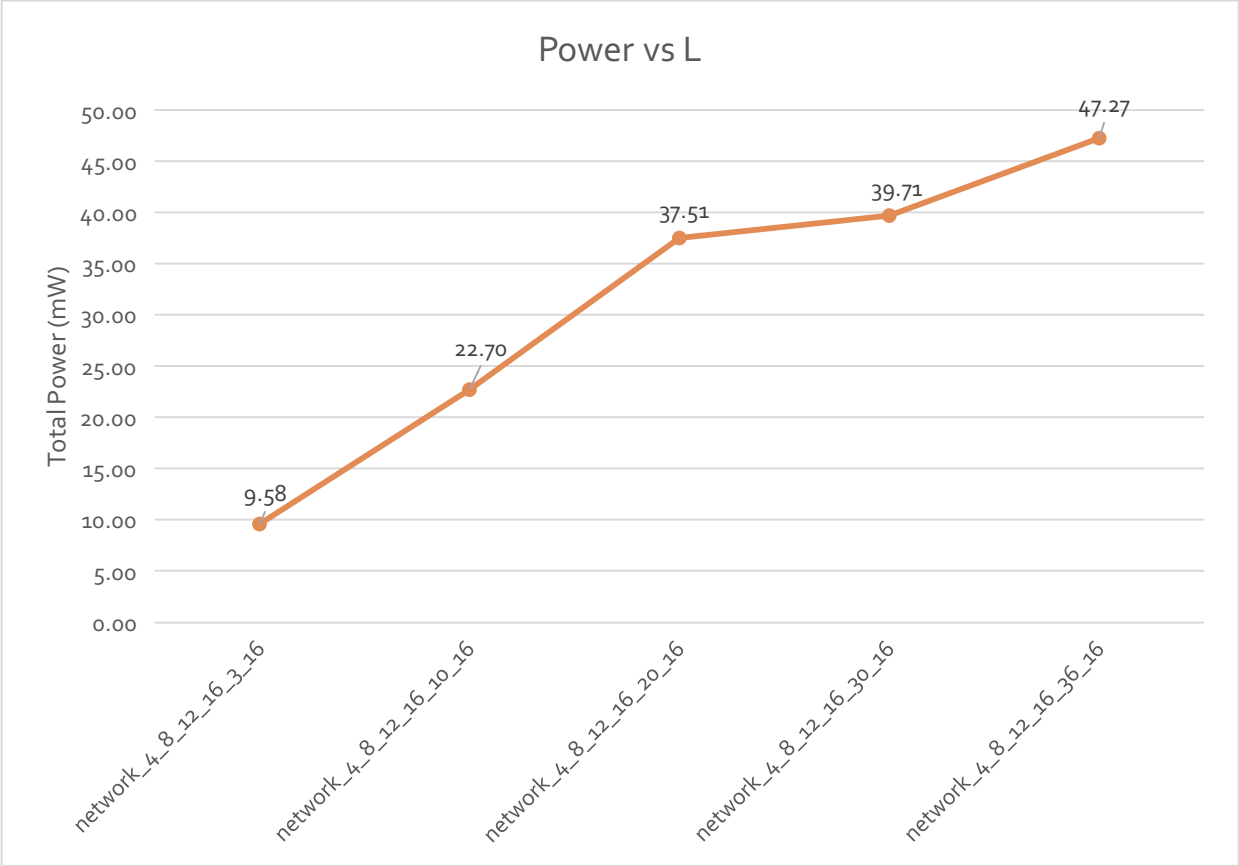
> Optimized Values for P1, P2, P3 for different network layers

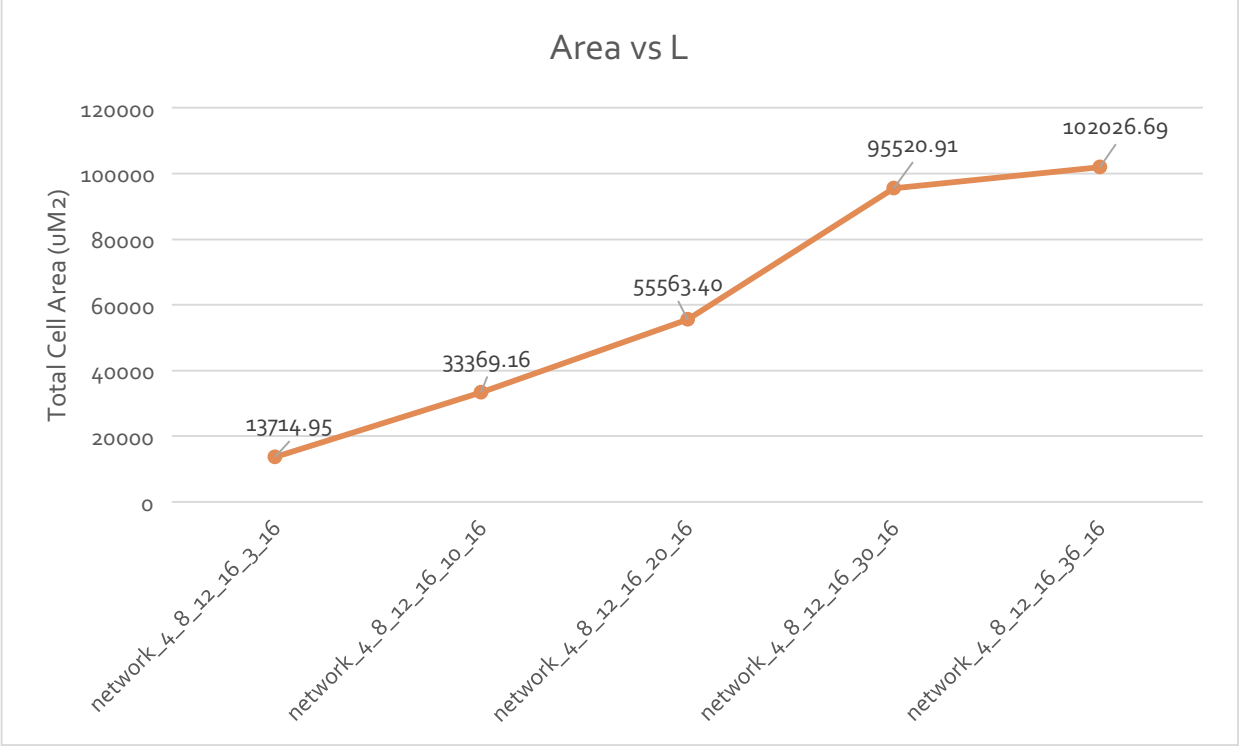
NETWORK LAYER	P1	P2	P3
network_4_8_12_16_10_16	2	4	4
network_4_8_12_16_20_16	4	6	8
network_4_8_12_16_36_16	8	12	16

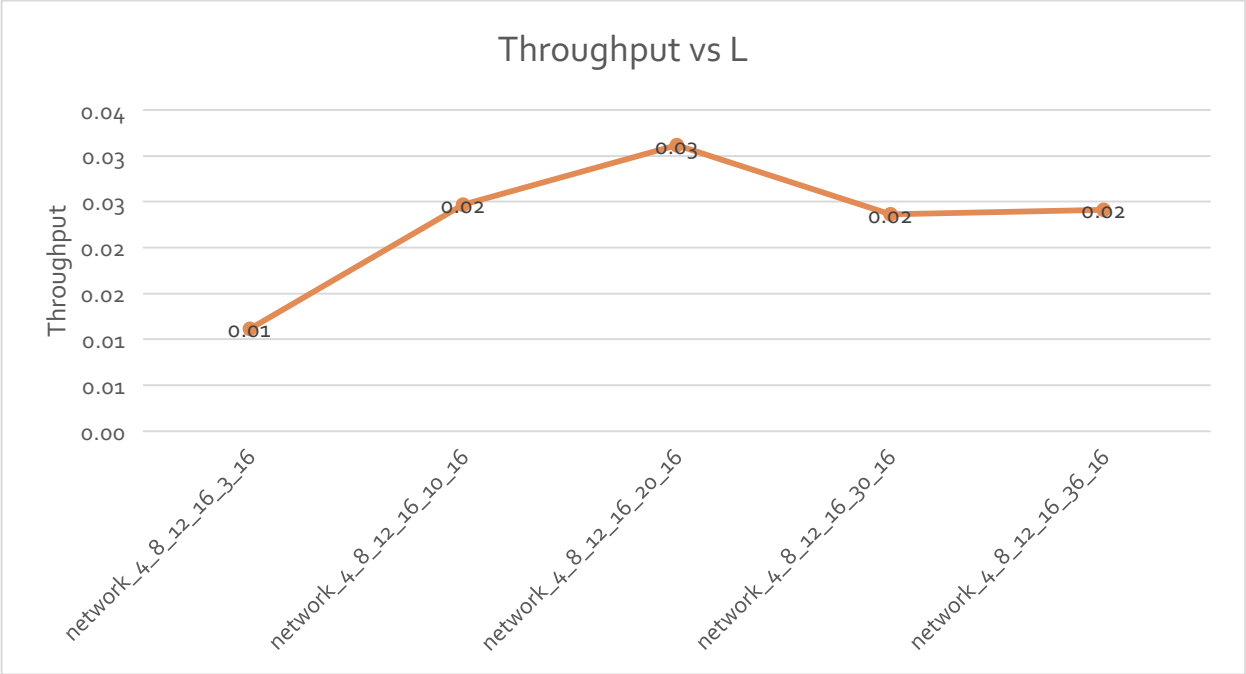
QUESTION 11:

> Performance of the system with variation in L Value

Layer	network_4_8_12_16_3_16	network_4_8_12_16_10_16	network_4_8_12_16_20_16	network_4_8_12_16_30_16	network_4_8_12_16_36_16
Target Frequency (GHz)	0.943	0.926	0.926	0.613	0.621
Clock Period (ns)	1.06	1.08	1.08	1.63	1.61
Combinational Area	7184.66	17717.99	29956.12	47192.65	54759.29
Non Combinational Area (um2)	6530.29	15651.17	25607.28	48328.26	47267.40
Total Cell Area (um2)	13714.95	33369.16	55563.40	95520.91	102026.69
Total Dynamic Power (mW)	9.31	22.03	36.38	37.94	45.25
Cell Leakage Power (uW)	274.78	672.27	1134.40	1769.50	2024.20
Total Power (mW)	9.58	22.70	37.51	39.71	47.27
Energy per Cycle Operation (pJ)	10.16	24.52	40.52	64.73	76.11
Timing Report (Slack)	0.00	0.00	0.00	0.00	0.00
	MET	MET	MET	MET	MET
Critical Part - Start Point	z_reg[1]	product_reg[1]	data_out_reg[0]	data_out_reg[1]	z_reg[3]
Critical Part - End Point	product_reg[15]	data_y_reg[15]	product_reg[15]	product_reg[15]	product_reg[15]
Parameter C x CLK (ns)	3390.00	1500.00	1190.00	1040.00	1030.00
Throughput	0.01	0.02	0.03	0.02	0.02


Power vs L
Total Power (mW)
50.00
45.00
40.00
35.00
30.00
25.00
20.00
15.00
10.00
5.00
0.00
9.58
22.70
37.51
39.71
47.27
network_4_8_12_16_3_16
network_4_8_12_16_10_16
network_4_8_12_16_20_16
network_4_8_12_16_30_16
network_4_8_12_16_36_16


Area vs L
Total Cell Area (uM2)
120000
100000
80000
60000
40000
20000
0
13714.95
33369.16
55563.40
95520.91
102026.69
network_4_8_12_16_3_16
network_4_8_12_16_10_16
network_4_8_12_16_20_16
network_4_8_12_16_30_16
network_4_8_12_16_36_16



QUESTION 12:

Generalizing the number of layers can be performed by interconnecting user defined number of single neural network layers in the top level module alone, since each of the layer is generated and capable of handling inputs and outputs independently. Therefore, scaling the neural network can be performed comprehensively.

END OF REPORT