Name: Deepak M Phadtare

ID:113786897

PROJECT-3 REPORT

Q1)

The hardware generation tool was designed in a generic way that supports scalability and flexibility. The scaling of vector 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 vector 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 SV script Control path and MAC modules are designed in a highly generic manner supporting any given vector 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 consumes tremendous time for simulation and synthesis.

Same control logic as of project 2 but slight modification due to pipelining inside MAC and FROM.

States:

init = will assign addr_x to counter x

And make x_ready = 1, y_valid = 0 , en_acc = 0 , clear = 1;

upload_x = This will start uploading x values to memory (x_ready = 1) and start incrementing counterx

(till counterx<N)

start_math = reset counter and assign counterx<=countery

start_mult = This states handles en_acc signal makes (en_acc=1) and increment

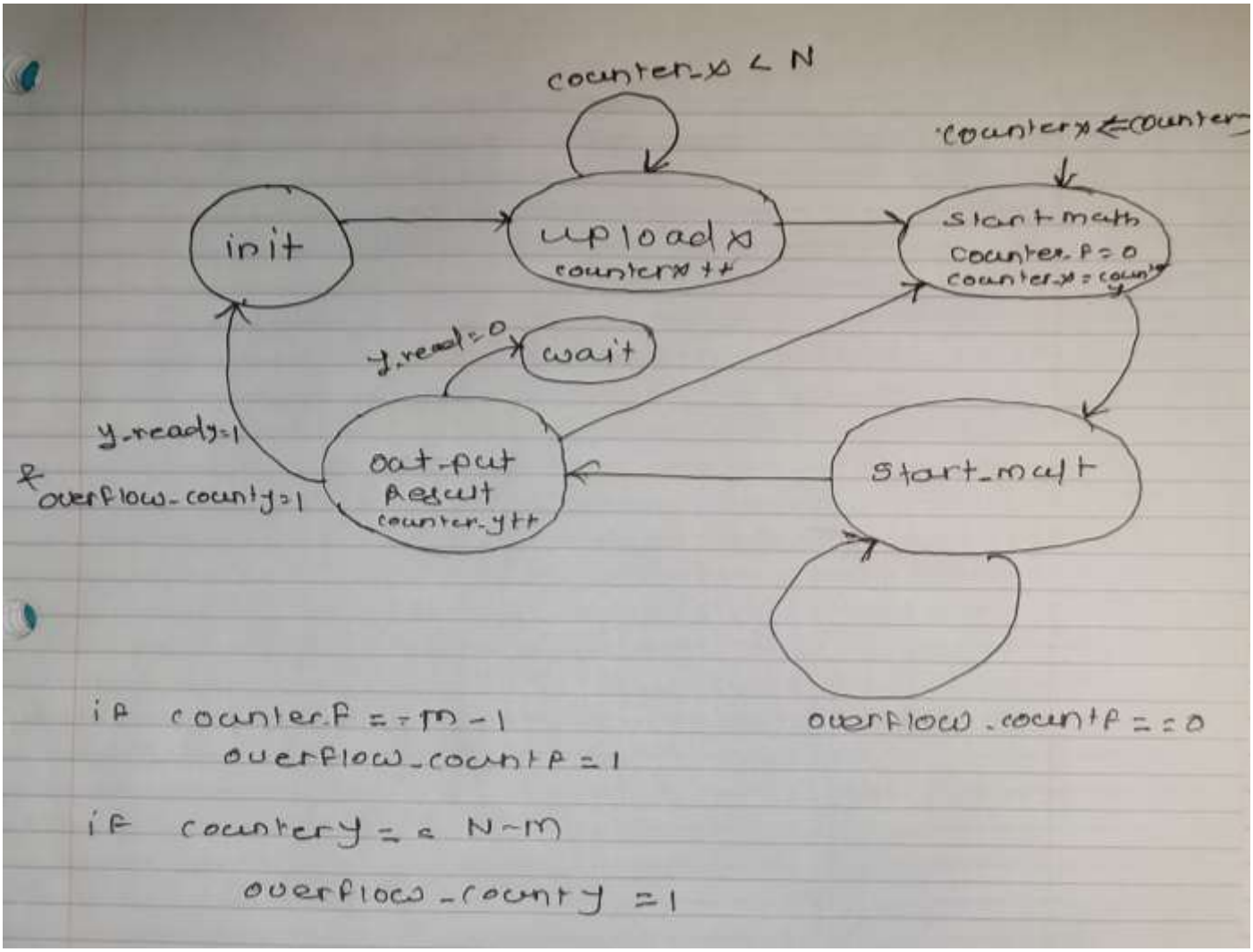
counter x and f (till overflow_countf==0)

output_result = This will display the output and increment counter y(till overflow_county==0).

wait_s = This state will incur if y_ready=0 and wait till it becomes 1.

nothing = (Not shown in Fig) This state will wait for a new input stream.

State transition as shown in Fig:

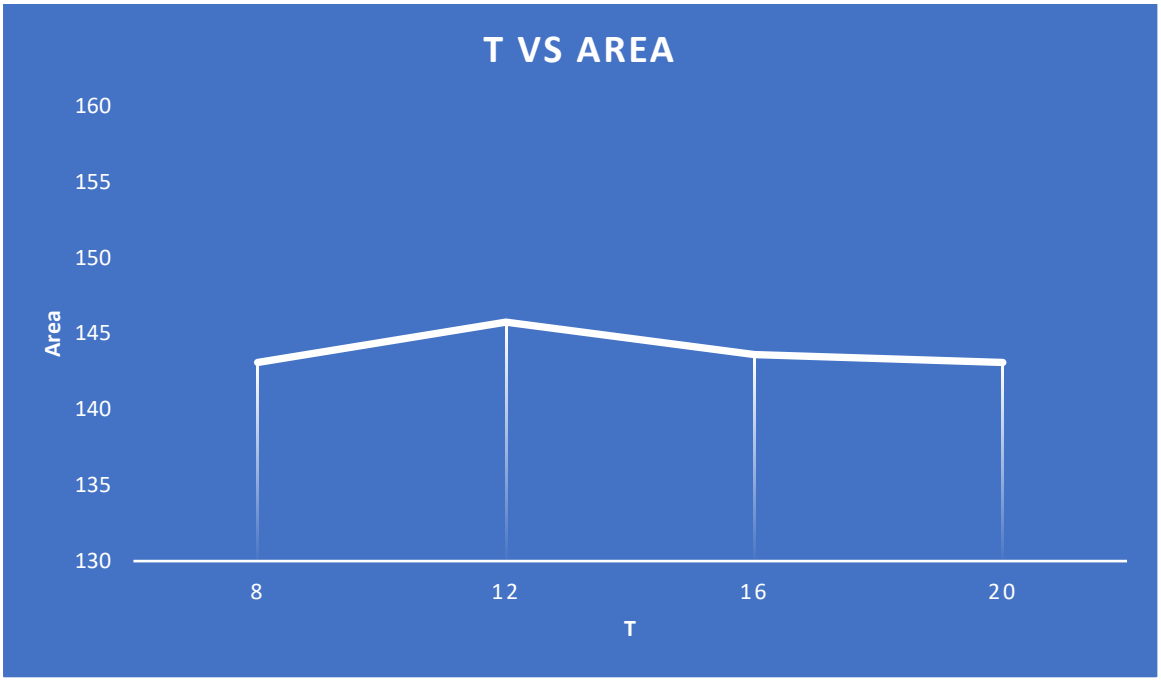


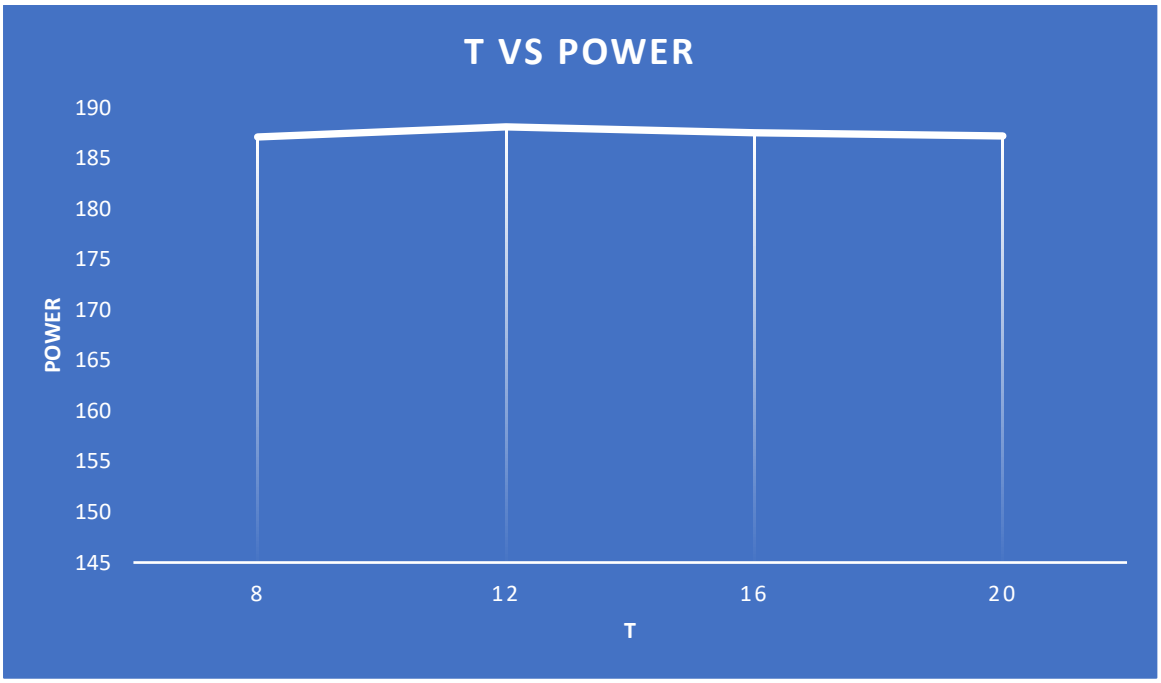
There are 2 other states start_math2 and pipeline.These states are due to the pipeline registor inside of mac unit.

Because of that pipeline registor it will take one extra cycle and these state will handle that.

Q3)

T	Power(uW)	Area(cell units)
8	187.1	143.10
12	188.1	145.76
16	187.53	143.63
20	187.2	143.10





CRITICAL PATH:

1)Conv_16_4_8_1:

Startpoint: present_state_reg[1]

Endpoint: counterx_reg[2]

2)Conv_16_4_12_1:

Startpoint: x_valid

Endpoint: counterx_reg[2]

3)Conv_16_4_16_1:

Startpoint: x_valid

Endpoint: counterx_reg[2]

4)Conv_16_4_20_1:

Startpoint: present_state_reg[1]

Endpoint: counterx_reg[2]

Q4)

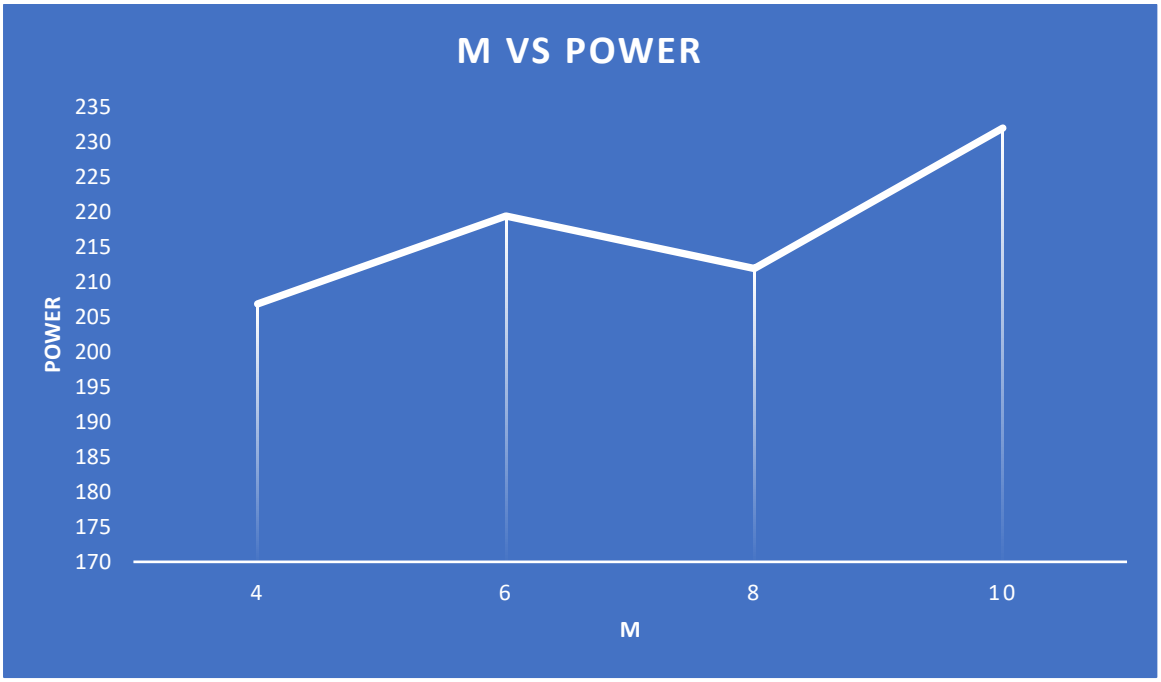
For calculating c :

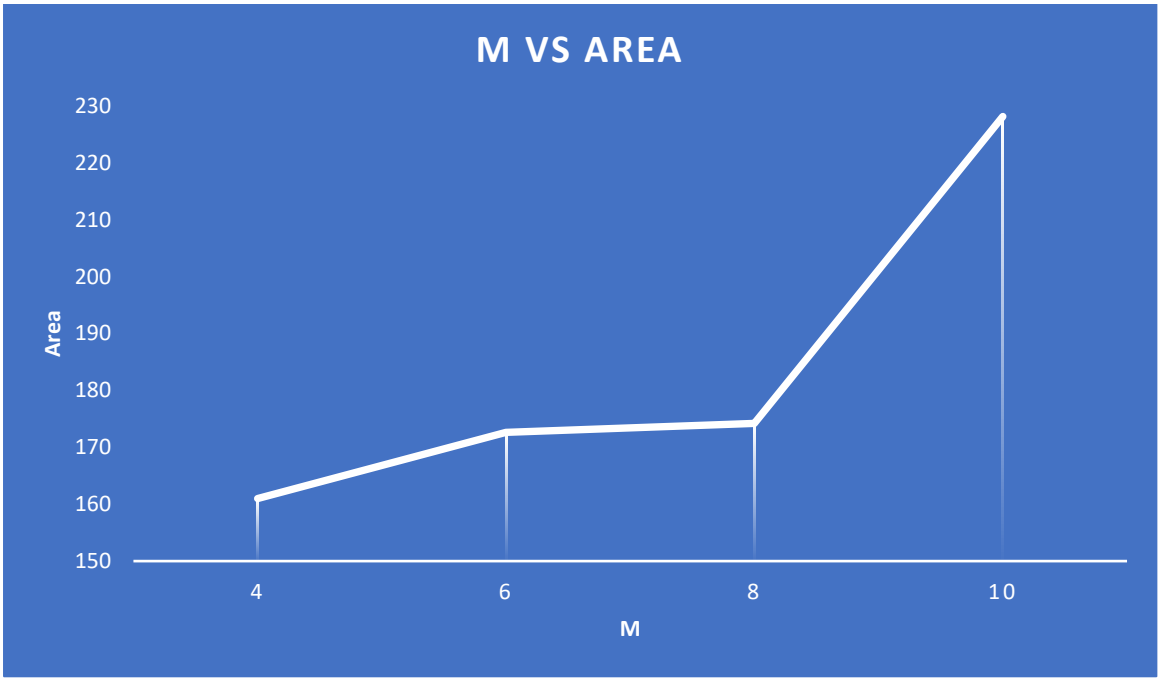
Modified respective testbenches such that x_valid and y_ready are always one.Then counted the cycles between my two upload states.

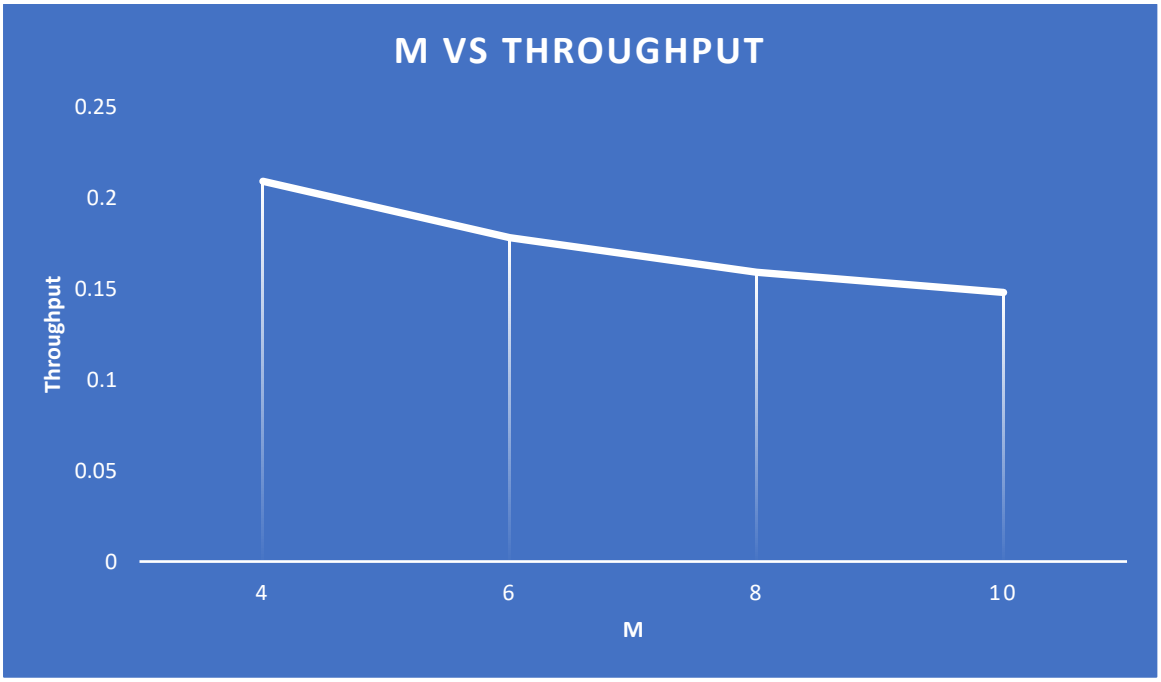
FREQ=1538.4 MHz

T=0.65 ns

M	Power(uW)	Area(cell units)	c	Throughput(billion i/p /sec)
4	206.8596	161	235	0.209
6	219.4157	172.6	275	0.178
8	211.9069	174.2	307	0.159
10	231.9894	182.7	331	0.1479







1)Conv_32_4_16_1:

Startpoint: y_ready

Endpoint: counterx_reg[3]

1)Conv_32_6_16_1:

Startpoint: present_state_reg[0]

Endpoint: counterx_reg[3]

3)Conv_32_8_16_1:

Startpoint: present_state_reg[0]

Endpoint: counterx_reg[3]

4) Conv_32_10_16_1:

Startpoint: present_state_reg[0]

Endpoint: counterx_reg[3]

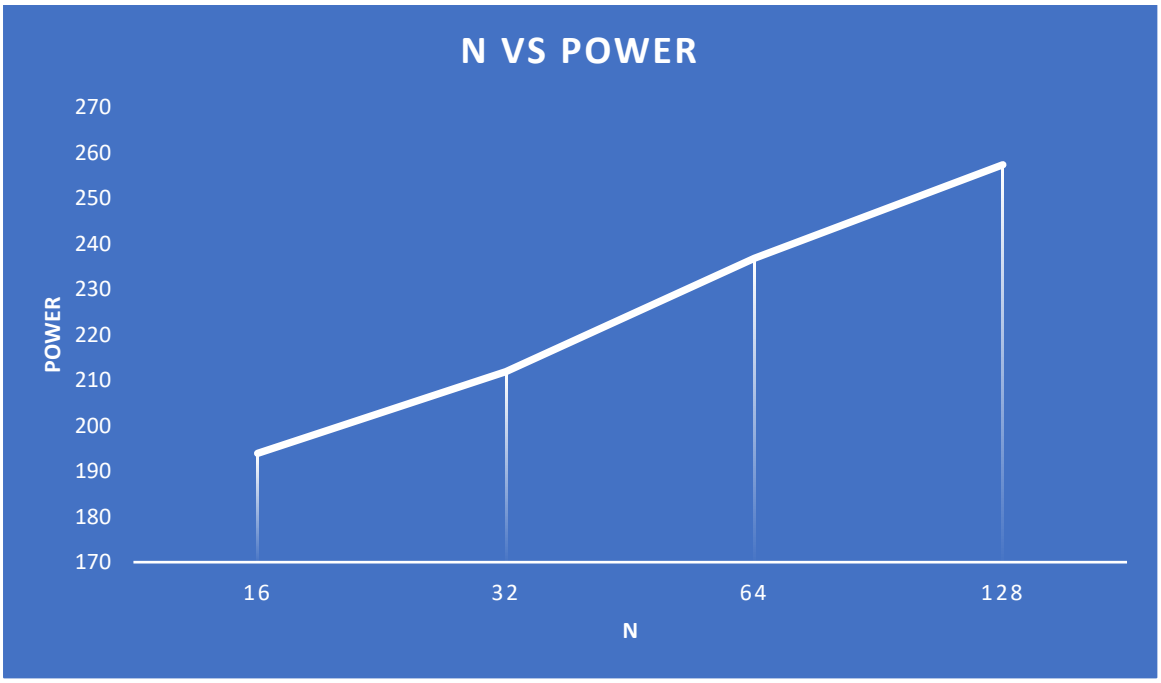
Observation on critical path: YES,location of the critical path change as M changes.

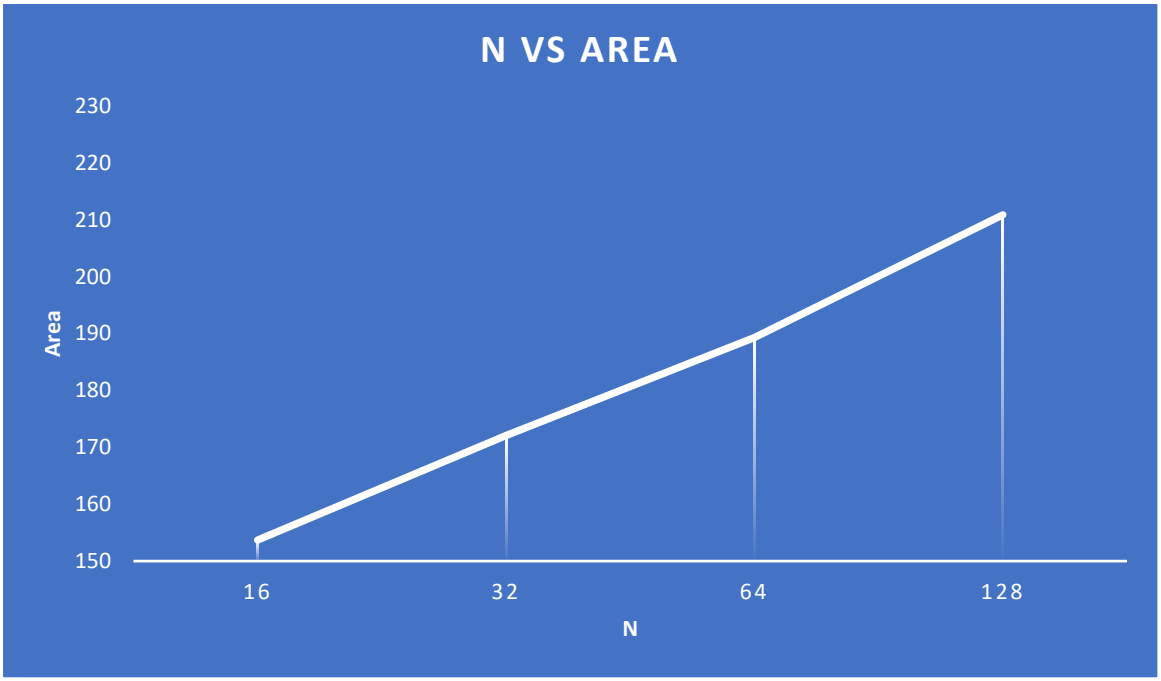
Q5)

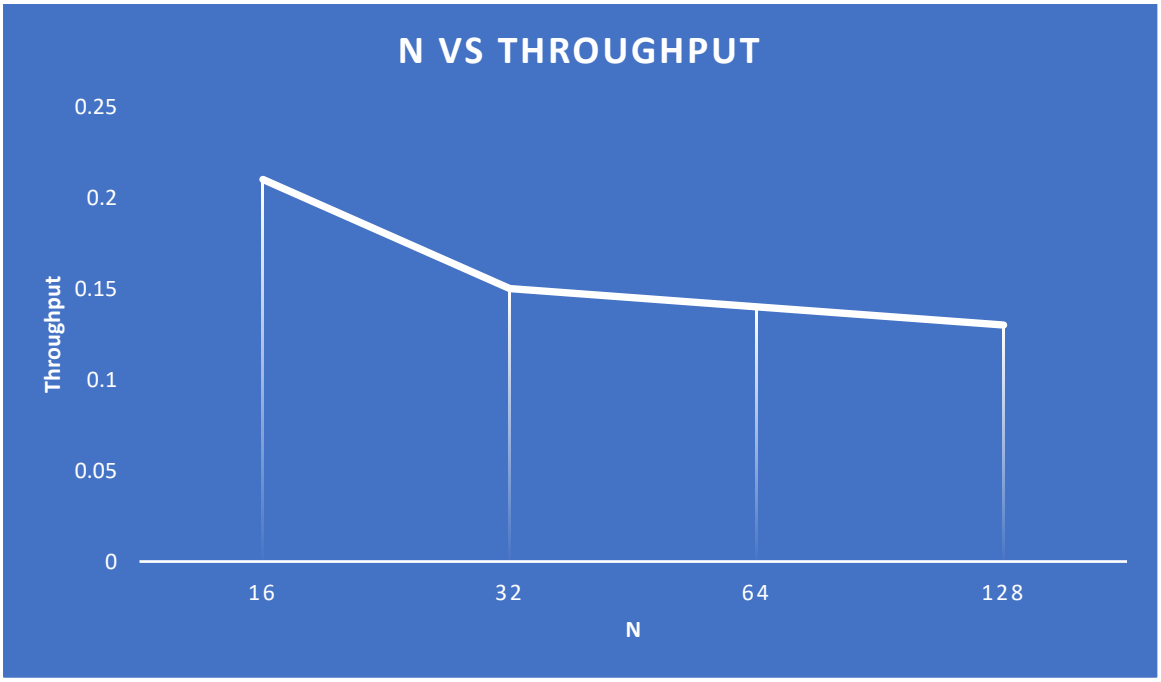
FREQ=1492.53 MHz

T=0.67 ns

N	Power(uW)	Area(cell units)	c	Throughput(billion i/p /sec)
16	193.9317	153.7	115	0.21
32	211.9069	172.1	307	0.15
64	236.8162	189.3	691	0.14
128	257.3785	210.9	1459	0.13







Critical Path:

1)conv_16_8_16_1:

Startpoint: x_valid

Endpoint: counterx_reg[2]

2) conv_32_8_16_1:

Startpoint: present_state_reg[0]

Endpoint: counterx_reg[3]

3) conv_64_8_16_1:

Startpoint: counterx_reg[0]

Endpoint: counterx_reg[5]

4) conv_128_8_16_1:

Startpoint: counterx_reg[2]

Endpoint: counterx_reg[2]

Observation on critical path: YES,location of the critical path change as N changes.