EE 599

Homework 1

Ruizhi Zhang

8230108665

Link:

https://github.com/ruizhiz/-EE-599_Ruizhi_Zhang_8230108665.git

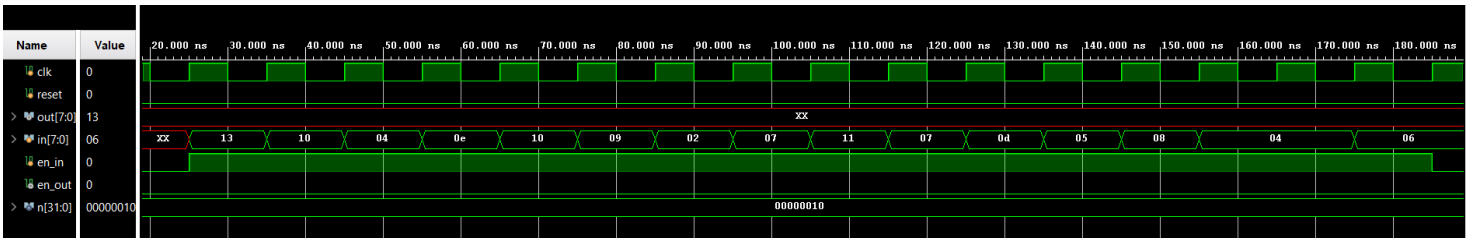
Odd-even transposition sort

The Odd_even.v and Odd_even_tb.v have been attached.

We do not use BRAM. All the data get from input directly, and then stored into registers inside circuits. Finishing comparing with each other, all of them will be output one by one.

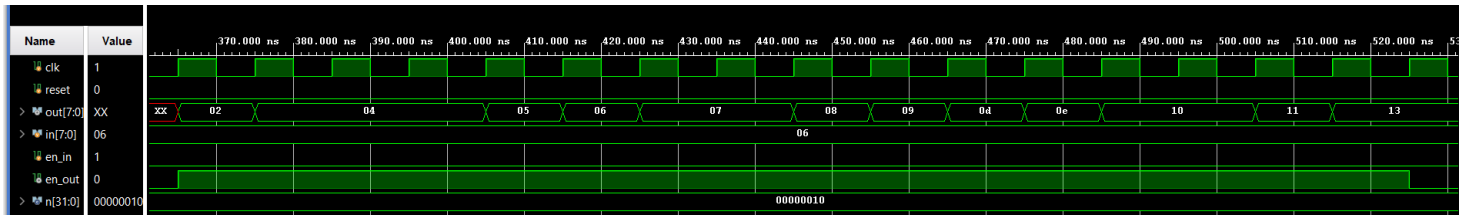
1. simulation

In Odd_even_tb.v, the input is generated by *$urandom%20;,* which means the input is [0, 20] unsigned random number.



Picture 1: the waveform of input for 16 elements.

Based on the picture above, the input is 13, 10, 04, 0e, 10, 09, 02, 07, 11, 07, 0d, 05, 08, 04, 04, 06. All of them are Hexadecimal.



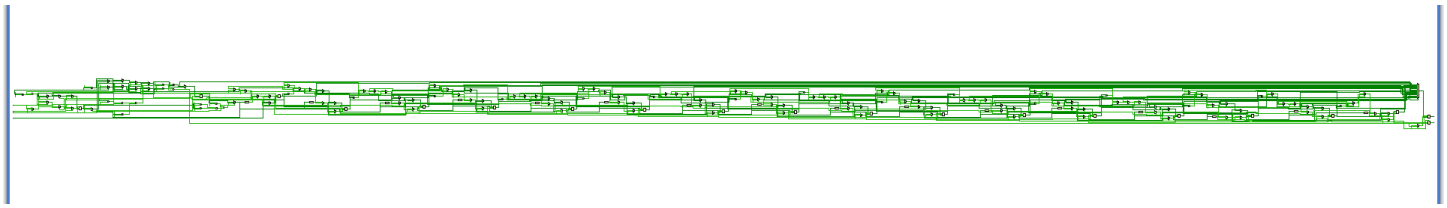
Picture 2: the waveform of output for 16 elements

The output order is 02, 04, 04, 04, 05, 06, 07, 07, 08, 09, 0d, 0e, 10, 10, 11, 13. All of them are Hexadecimal.

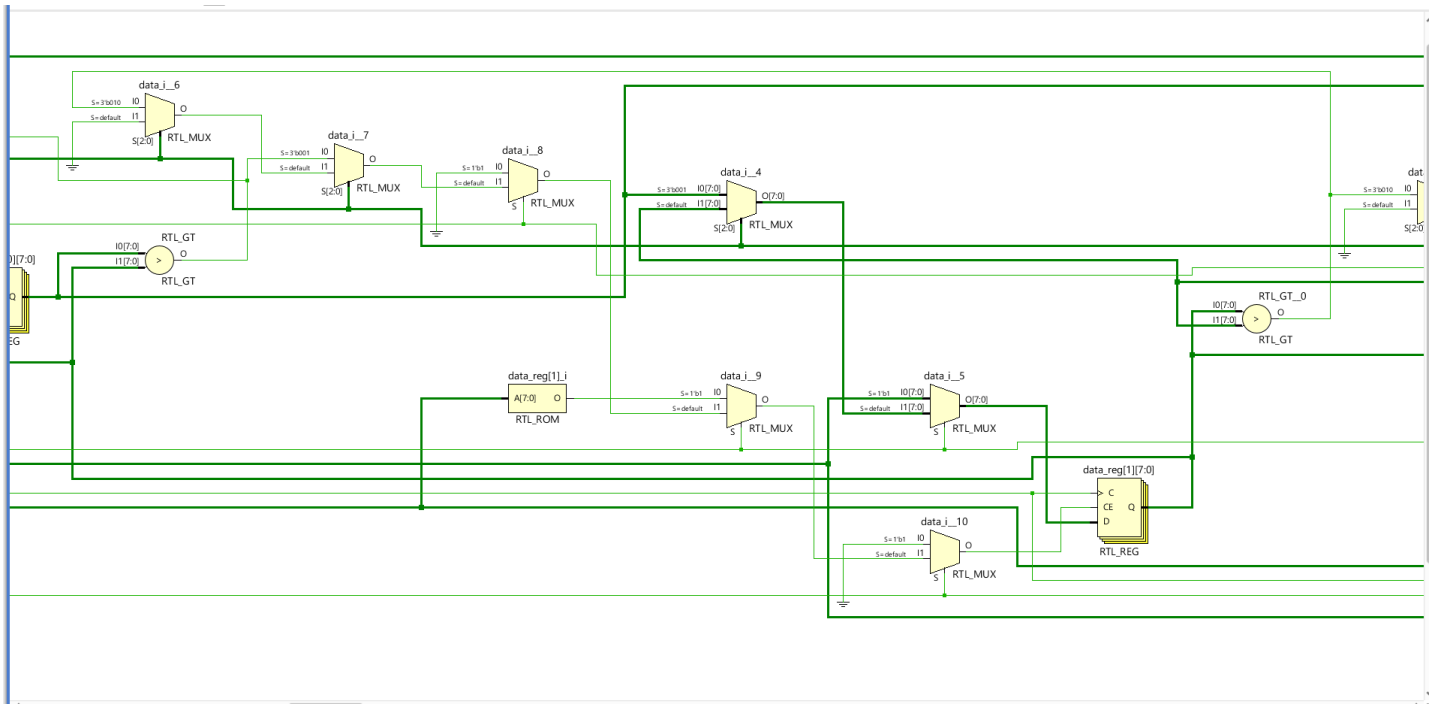
As we can see, the input has been sorted.

Part 1: 16 elements

2. Schematics.



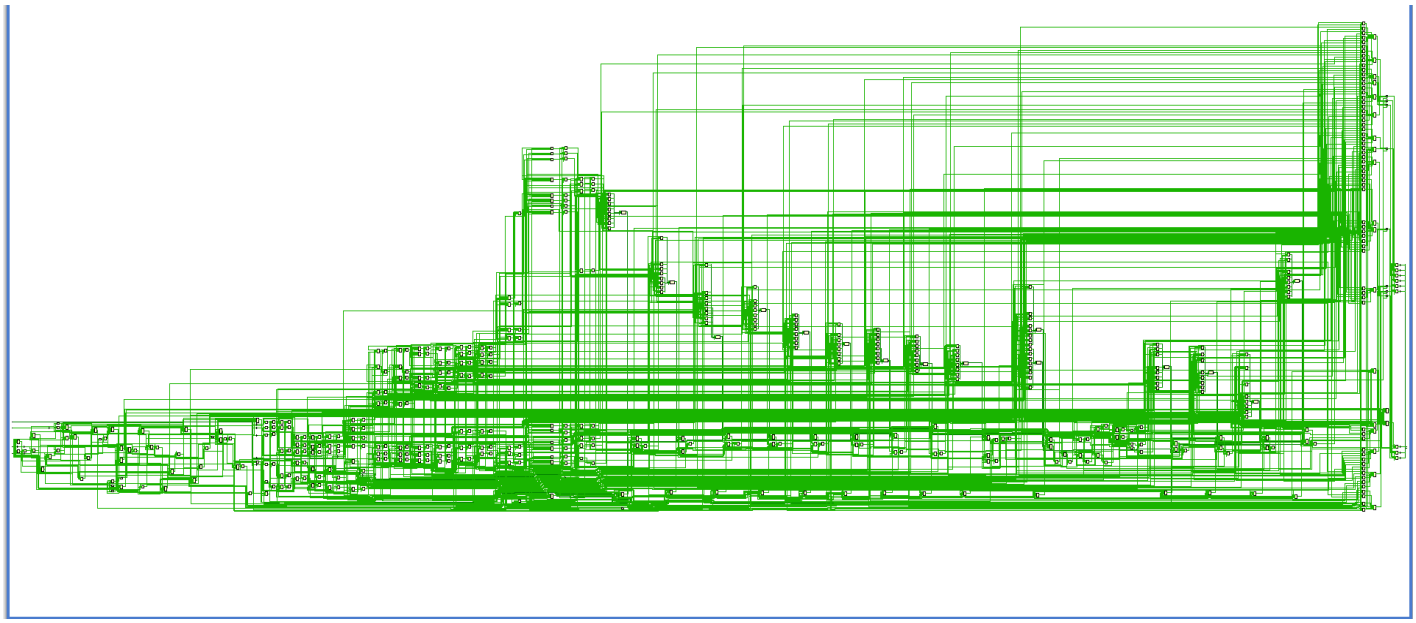
Picture 3: the schematics of our design



Picture 4: the basic unit of our design

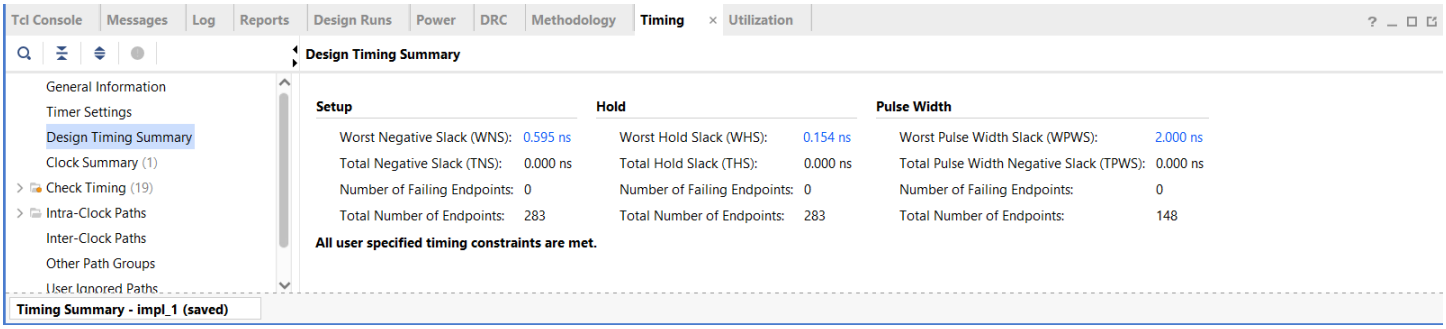
Because our design has 16 elements, they compare to each other. So, we have 16 above basic units in total, them combine to be picture 3.

3. Synthesis.

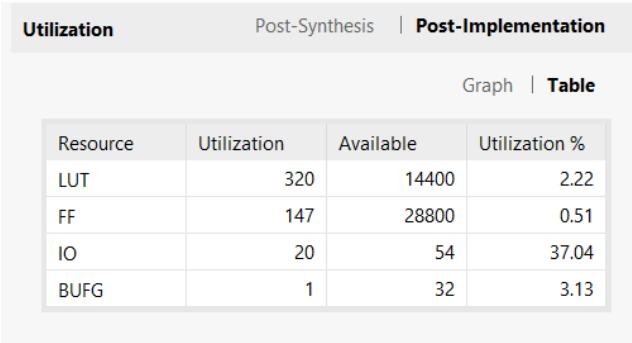


Picture 5: synthesis design

4. Resource estimation and timing estimation



Picture 6: timing estimation

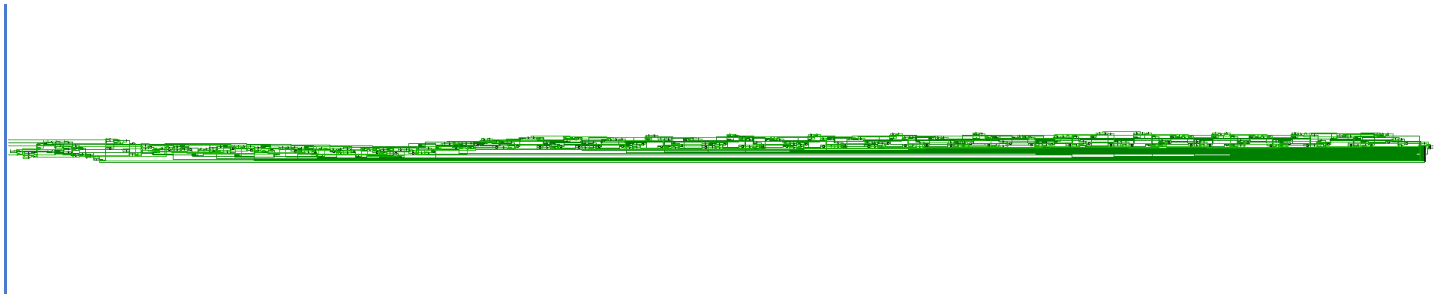


Resource	Utilization	Available	Utilization %
LUT	320	14400	2.22
FF	147	28800	0.51
IO	20	54	37.04
BUFG	1	32	3.13

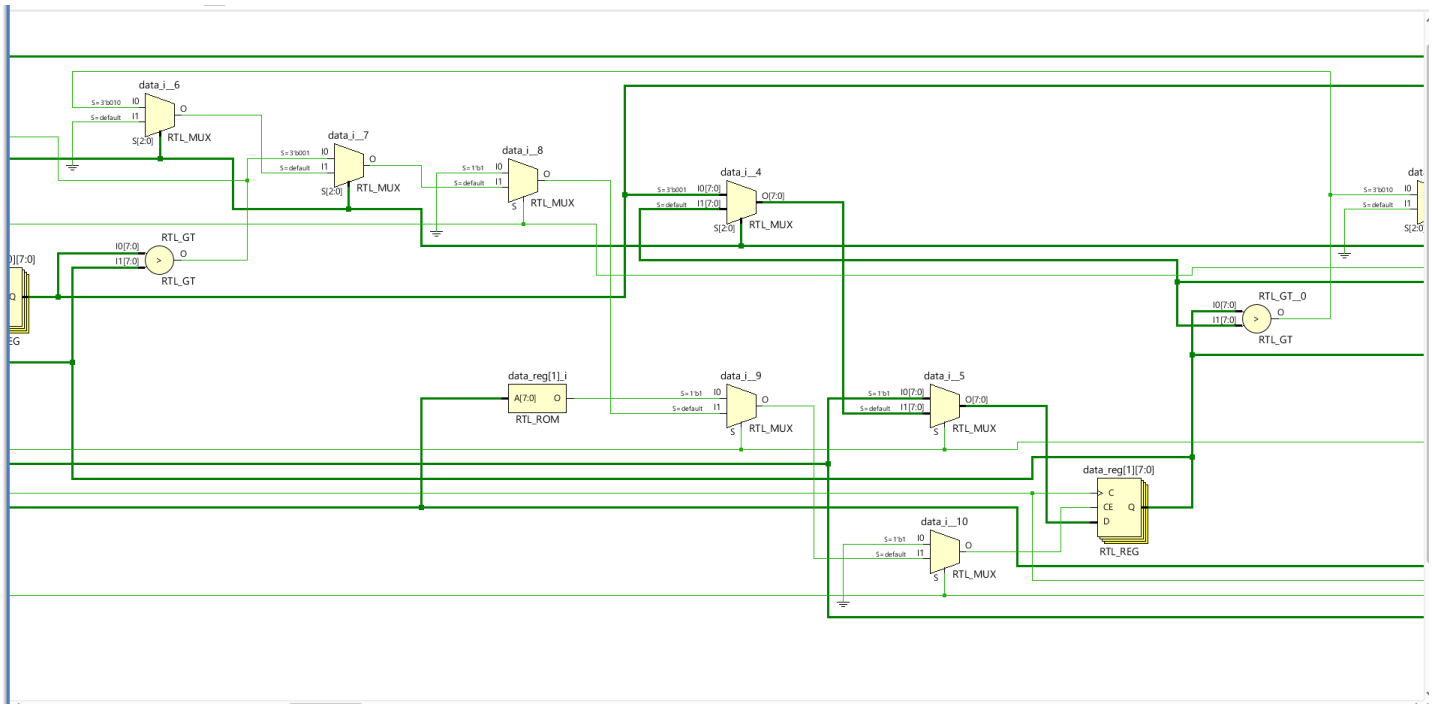
Picture 7: resource estimation

Part 2: 32 elements

1. Schematics.



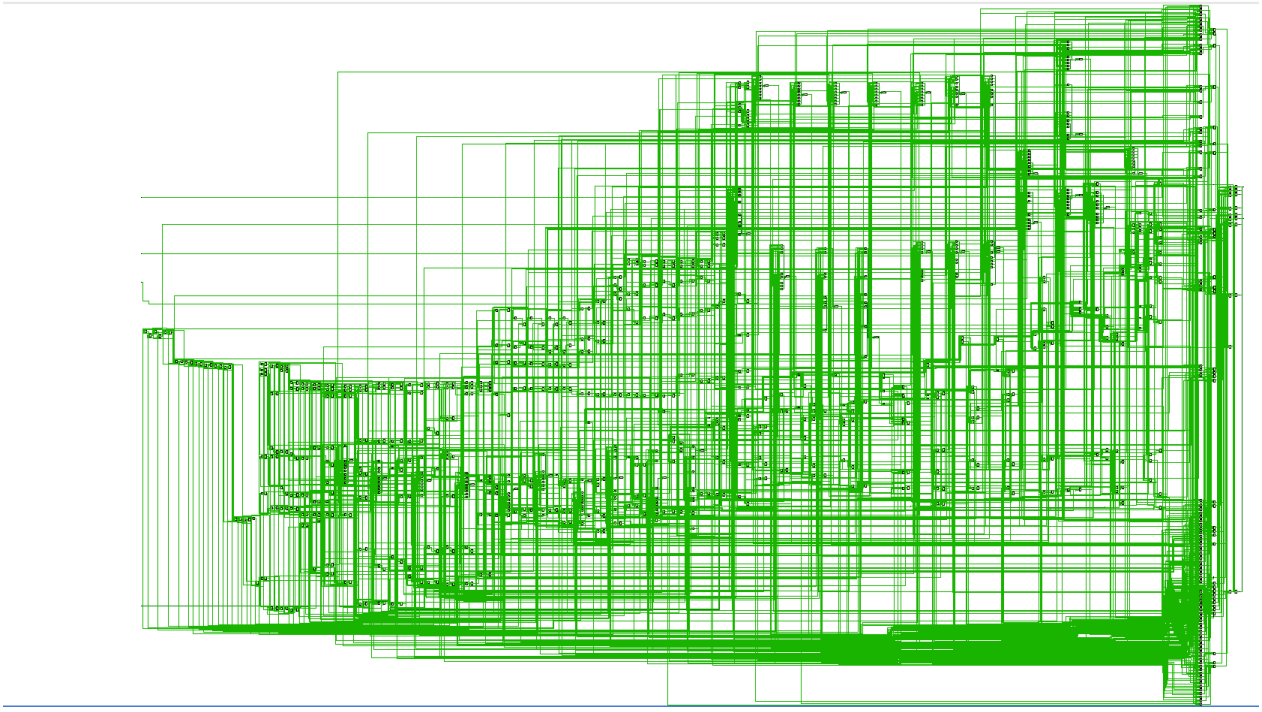
Picture 8: the schematics of our design



Picture 9: the basic unit of our design

Because our design has 32 elements, they compare to each other. So, we have 32 above basic units in total, them combine to be picture 3.

2. Synthesis.



Picture 10: synthesis design

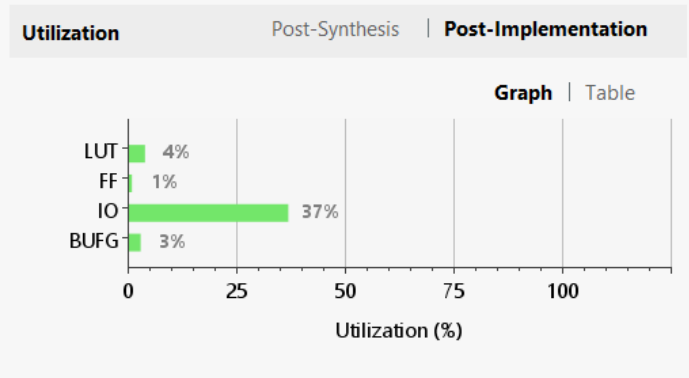
3. Resource estimation and timing estimation

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.184 ns	Worst Hold Slack (WHS):	0.089 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:	539	Total Number of Endpoints:	539	Total Number of Endpoints:	276

All user specified timing constraints are met.

Picture 11: timing estimation



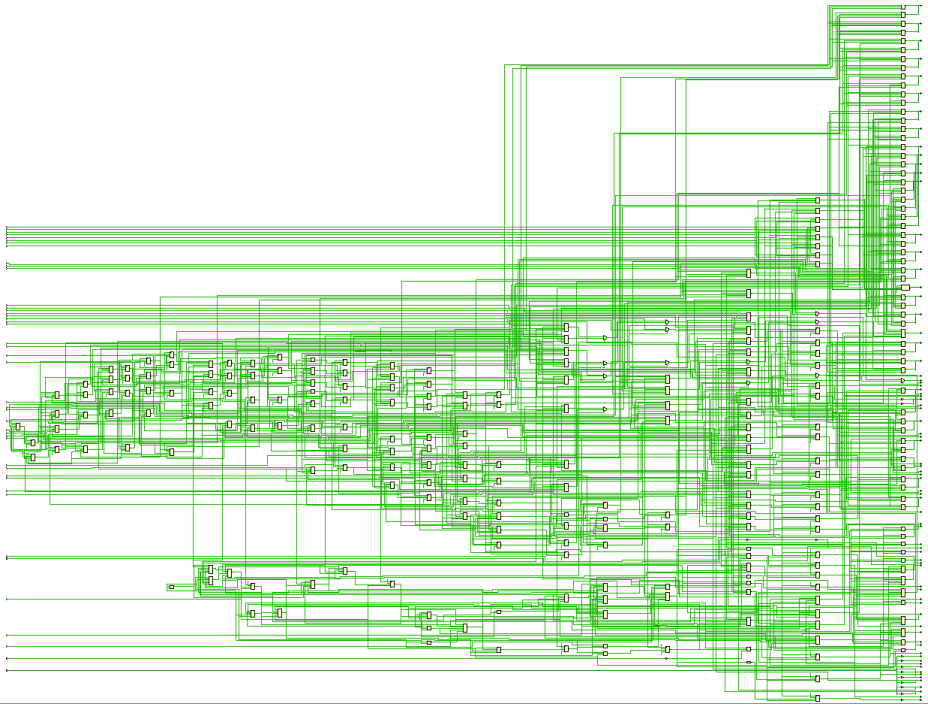
Picture 12: resource estimation

Part 3: 64 elements

1. Schematics.

Picture 13: the schematics of our design

2. Synthesis.



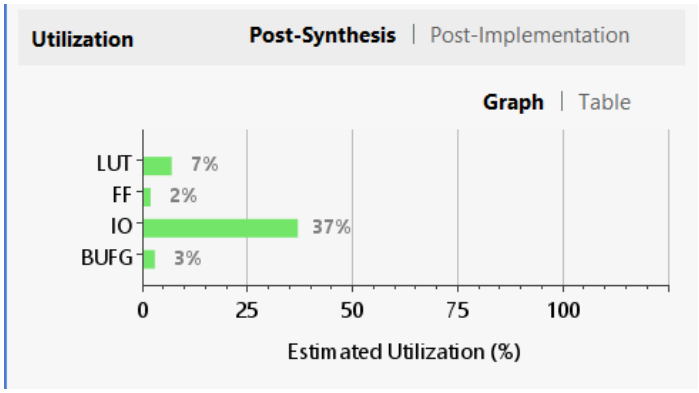
Picture 14: synthesis design

3. Resource estimation and timing estimation

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.196 ns	Worst Hold Slack (WHS):	0.129 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:	1051	Total Number of Endpoints:	1051	Total Number of Endpoints:	532

All user specified timing constraints are met.

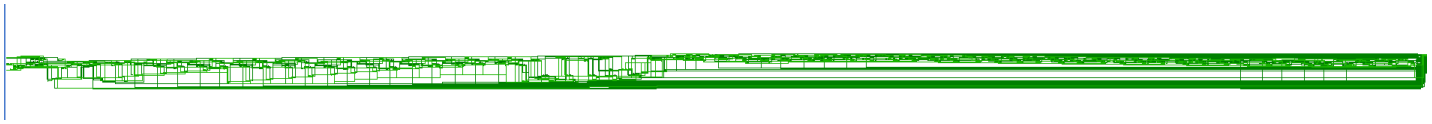
Picture 15: timing estimation



Picture 16: resource estimation

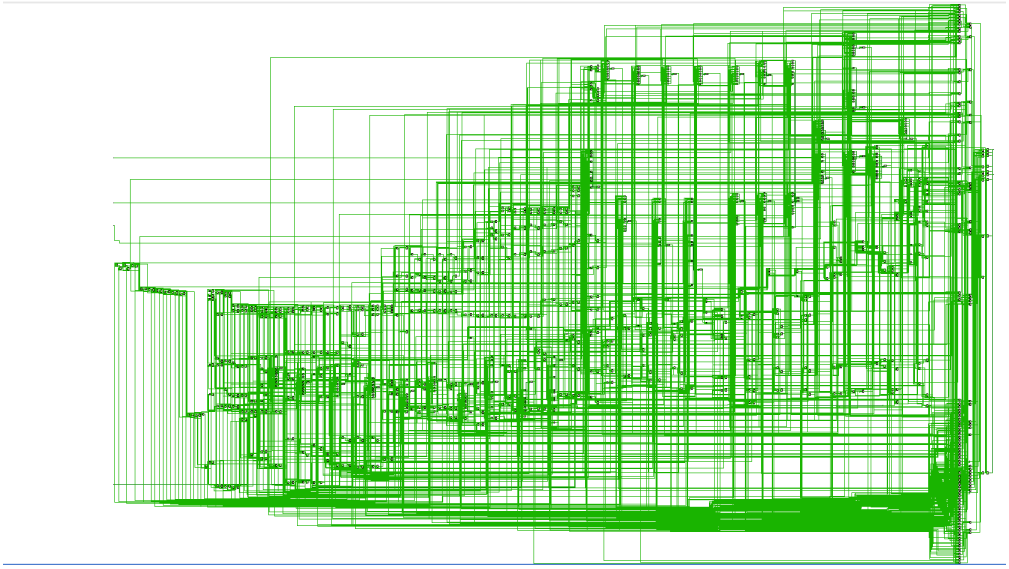
Part 4: 128 elements

1. Schematics.



Picture 17: the schematics of our design

2. Synthesis.



Picture 14: synthesis design

3. Resource estimation and timing estimation

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.887 ns	Worst Hold Slack (WHS):	0.052 ns	Worst Pulse Width Slack (WPWS):	4.500 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:	2075	Total Number of Endpoints:	2075	Total Number of Endpoints:	1044

All user specified timing constraints are met.

Picture 15: timing estimation

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	2162	14400	15.01
FF	1043	28800	3.62
IO	20	54	37.04
BUFG	1	32	3.13

Picture 16: resource estimation

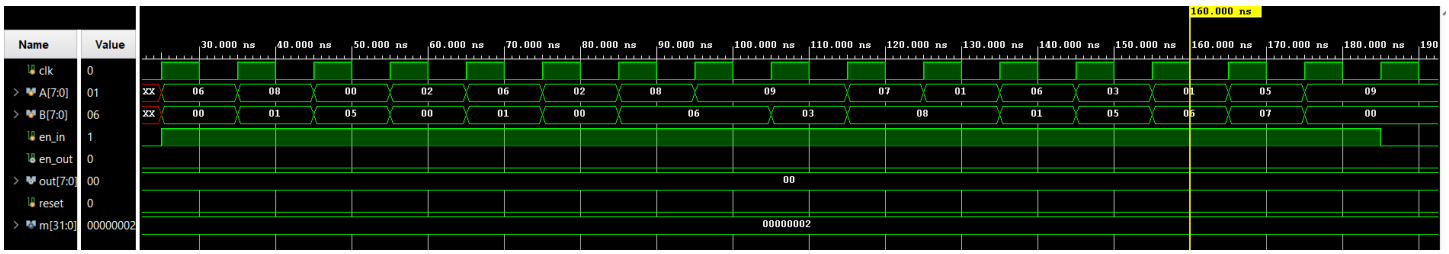
Dense Matrix – Matrix Multiplication

M_mult_1.v, adder.v, multiply.v, MulandAddTree.v and M_mult_tb.v have been attached.

M_mult_2.v is used to test the largest number of parallel MulandAddTrees.

BRAM is not used in this design. All the matrix data get from input in several clock cycle.

1. Simulation



Picture 17: the waveform of input

The input is generated by A <= $urandom%10; B <= $urandom%10; Both A and B are random unsigned int from 0 to 10.

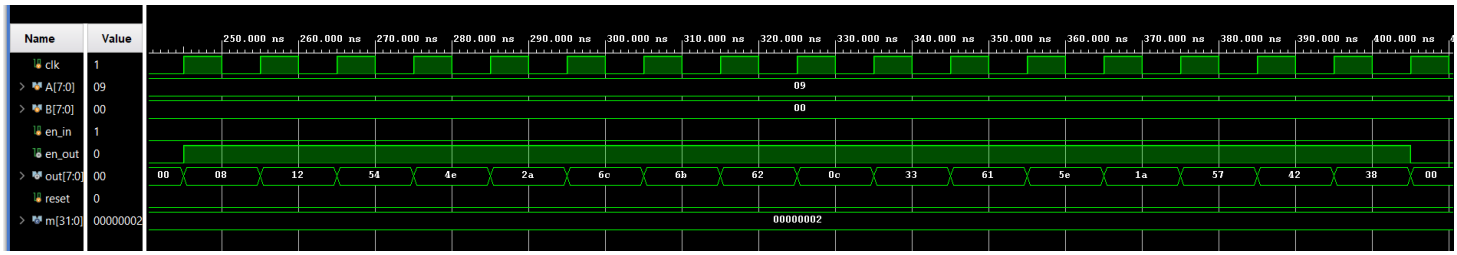
As we can see, the Matrixes are:

A:

06	08	00	02
06	02	08	09
09	07	01	06
03	01	05	09

B:

00	01	03	05
01	00	08	06
05	06	08	07
00	06	01	00



Picture 18: waveform of output

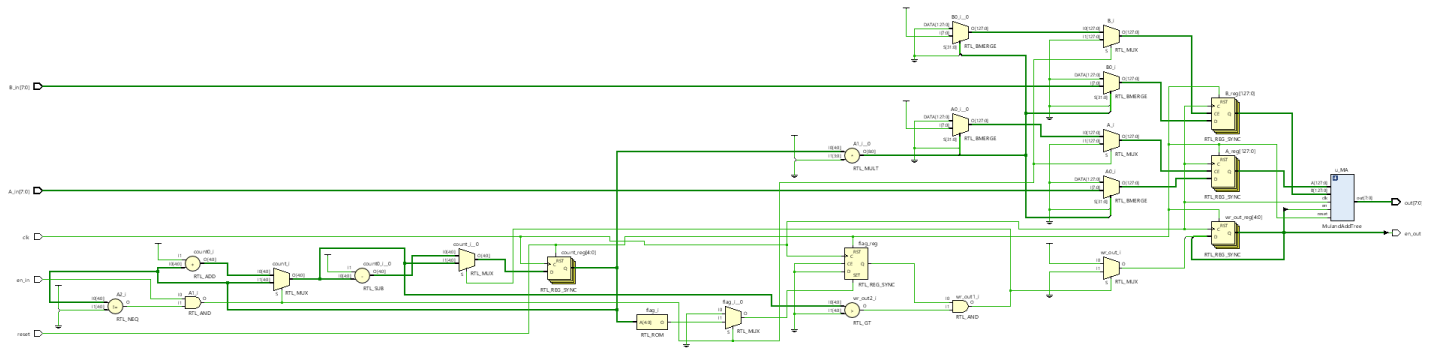
As we can see, the result Matrix is:

Out:

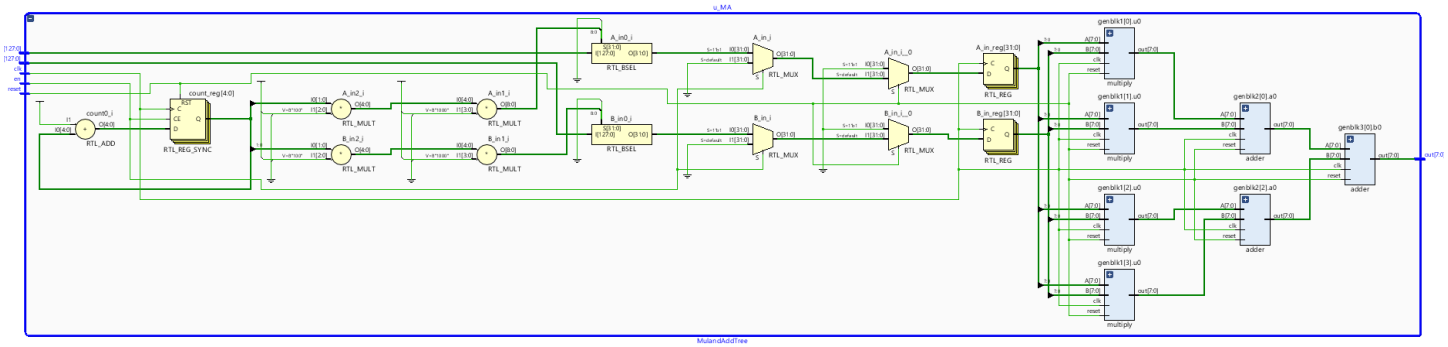
8	12	54	4e
2a	6c	6b	62
0c	33	61	5e
1a	57	42	38

This result is what we expected.

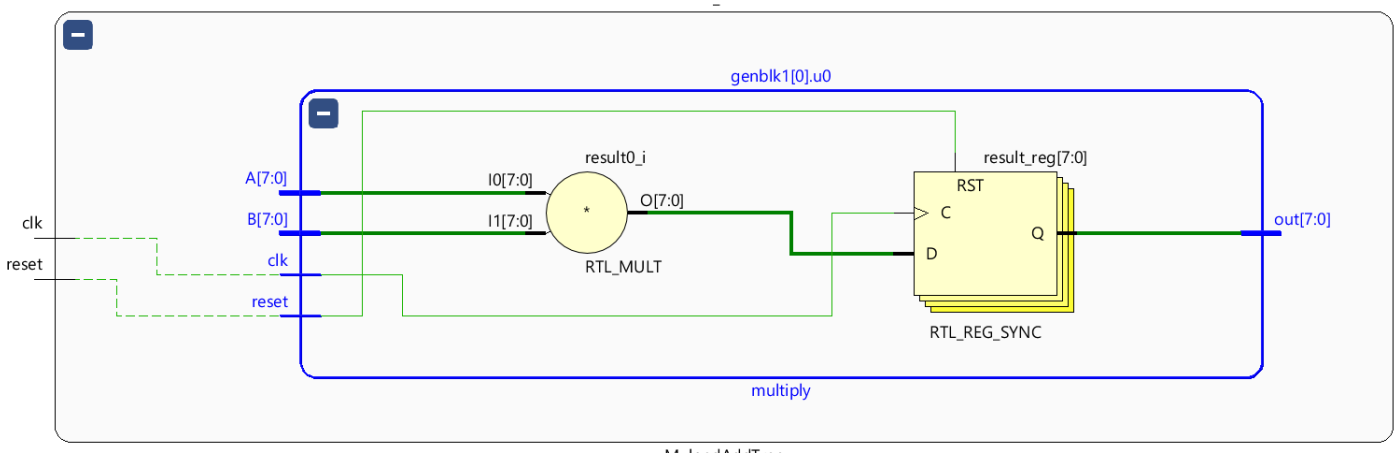
2. Schematics:



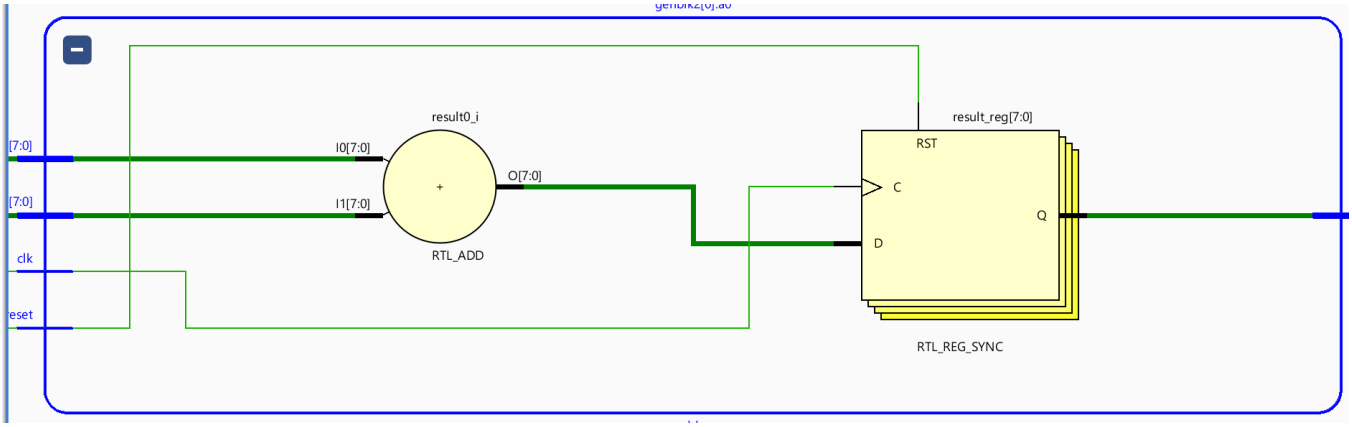
Picture 19: the schematics of our design



Picture 20: MulandAddTree



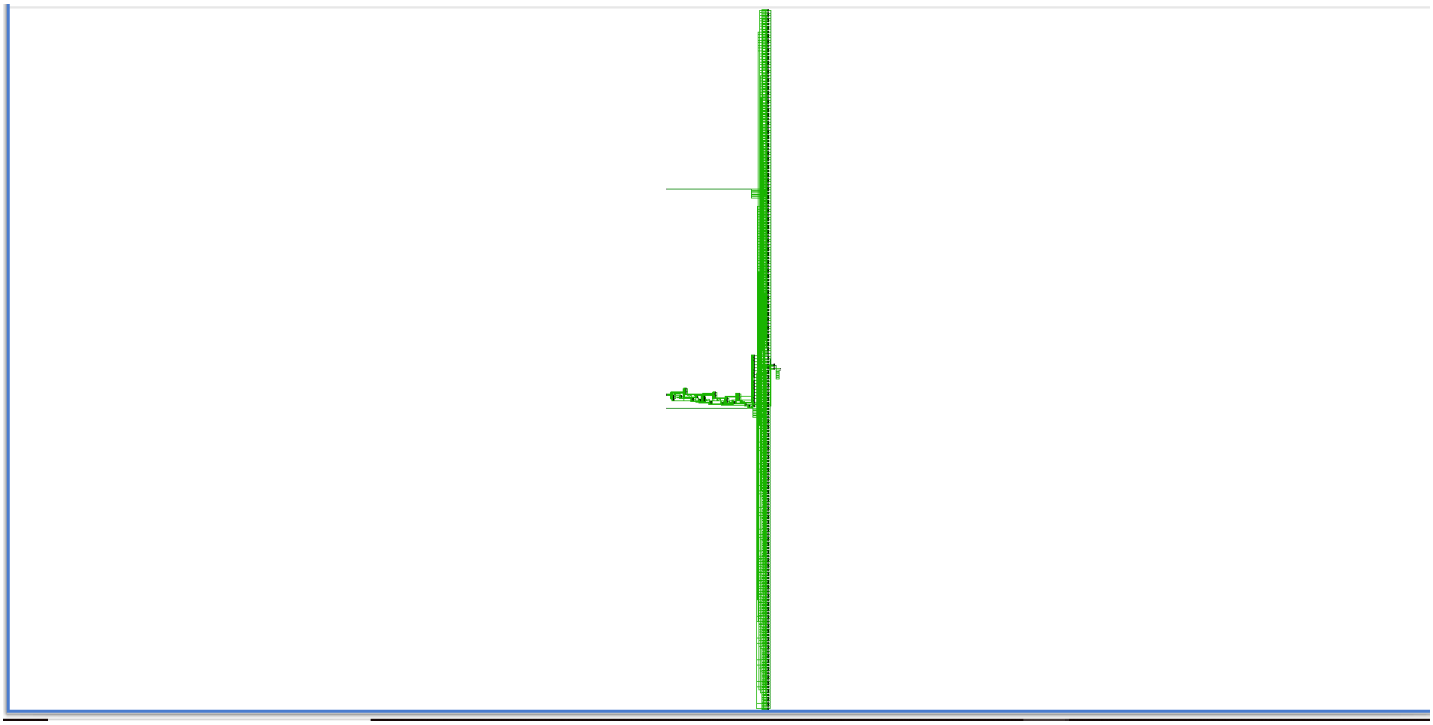
Picture 21: multiply



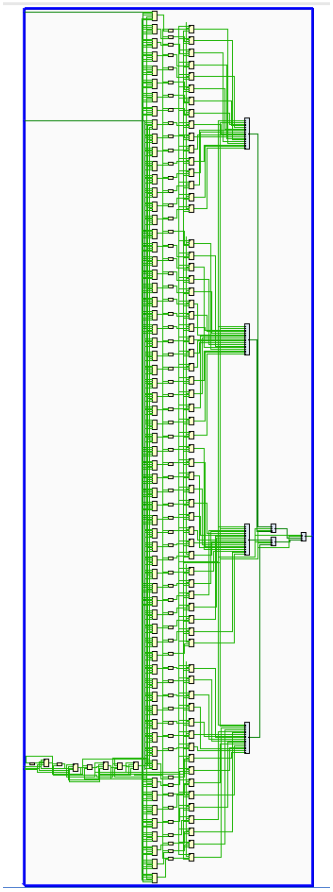
Picture 22: adder

Part 1: 4*4

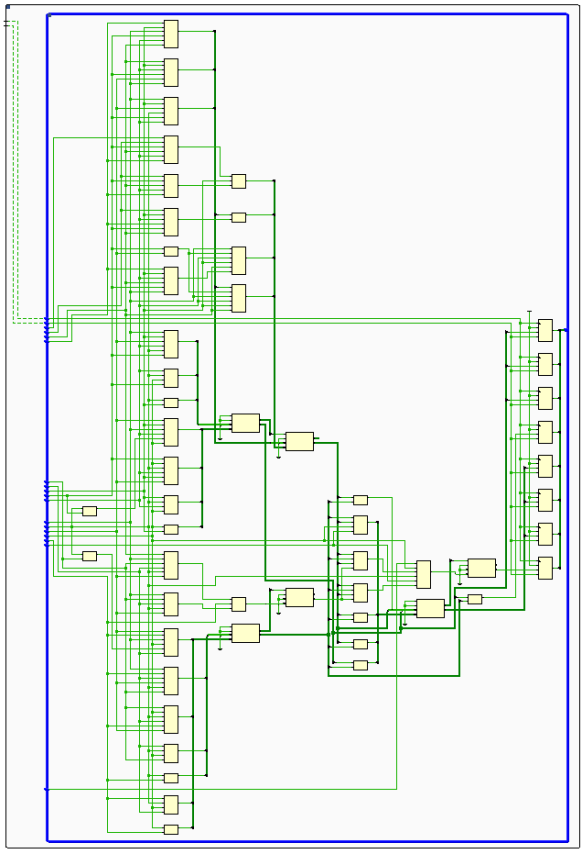
3. Synthesis



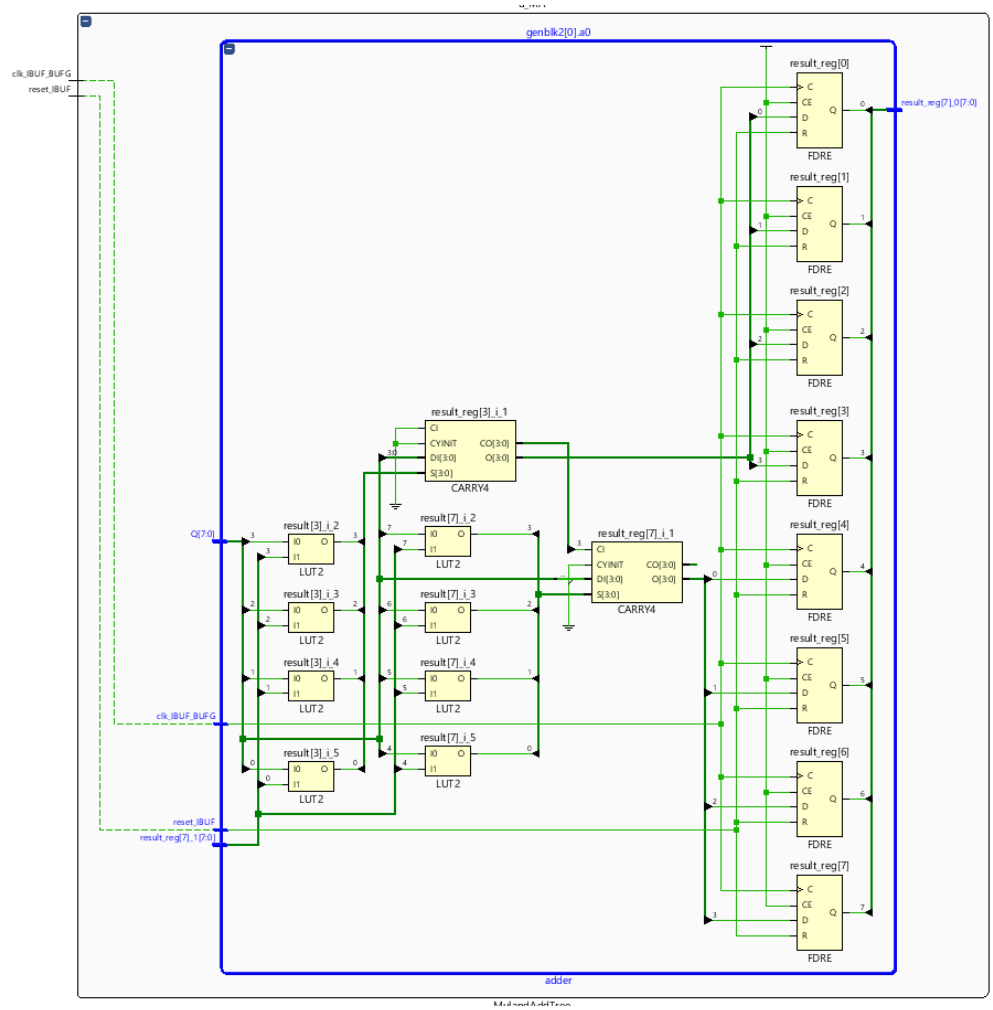
Picture 23: whole schematics of design



Picture 24: MulandAddTree



Picture 25: multiply



Picture 26: adder

4. estimation reports

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	247	14400	1.72
FF	327	28800	1.14
IO	28	54	51.85
BUFG	1	32	3.13

Power Summary | On-Chip

Total On-Chip Power:	**0.105 W**
Junction Temperature:	**26.2 °C**
Thermal Margin:	58.8 °C (5.0 W)
Effective ϑJA:	11.5 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Implemented Power Report

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.286 ns	Worst Hold Slack (WHS):	0.218 ns	Worst Pulse Width Slack (WPWS):	4.500 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:	395	Total Number of Endpoints:	395	Total Number of Endpoints:	392

All user specified timing constraints are met.

5. the largest number of parallel MulandAddTrees.

The largest number of parallel MulandAddTrees is 256, M_multi_2.v is used to do the test. The following table is our resource utilization.

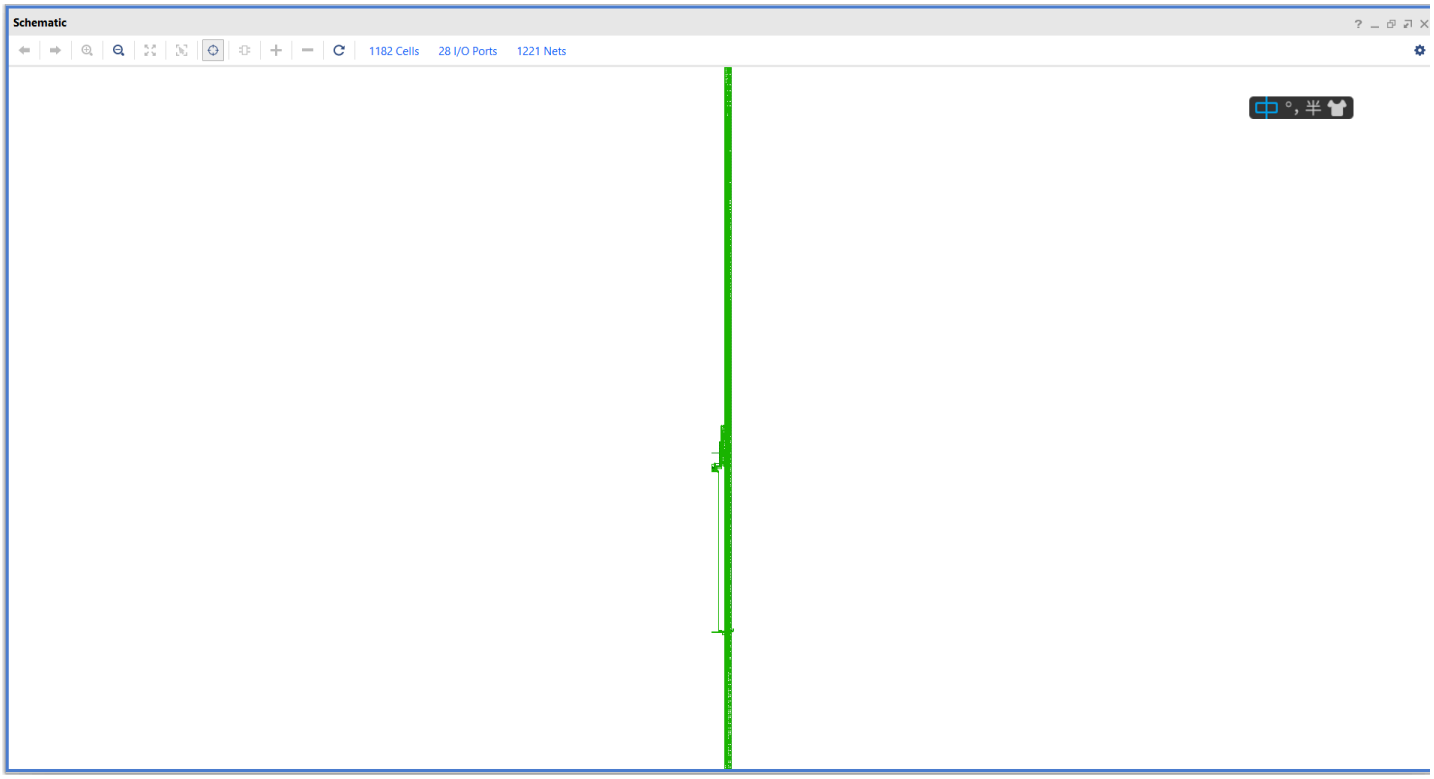
Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

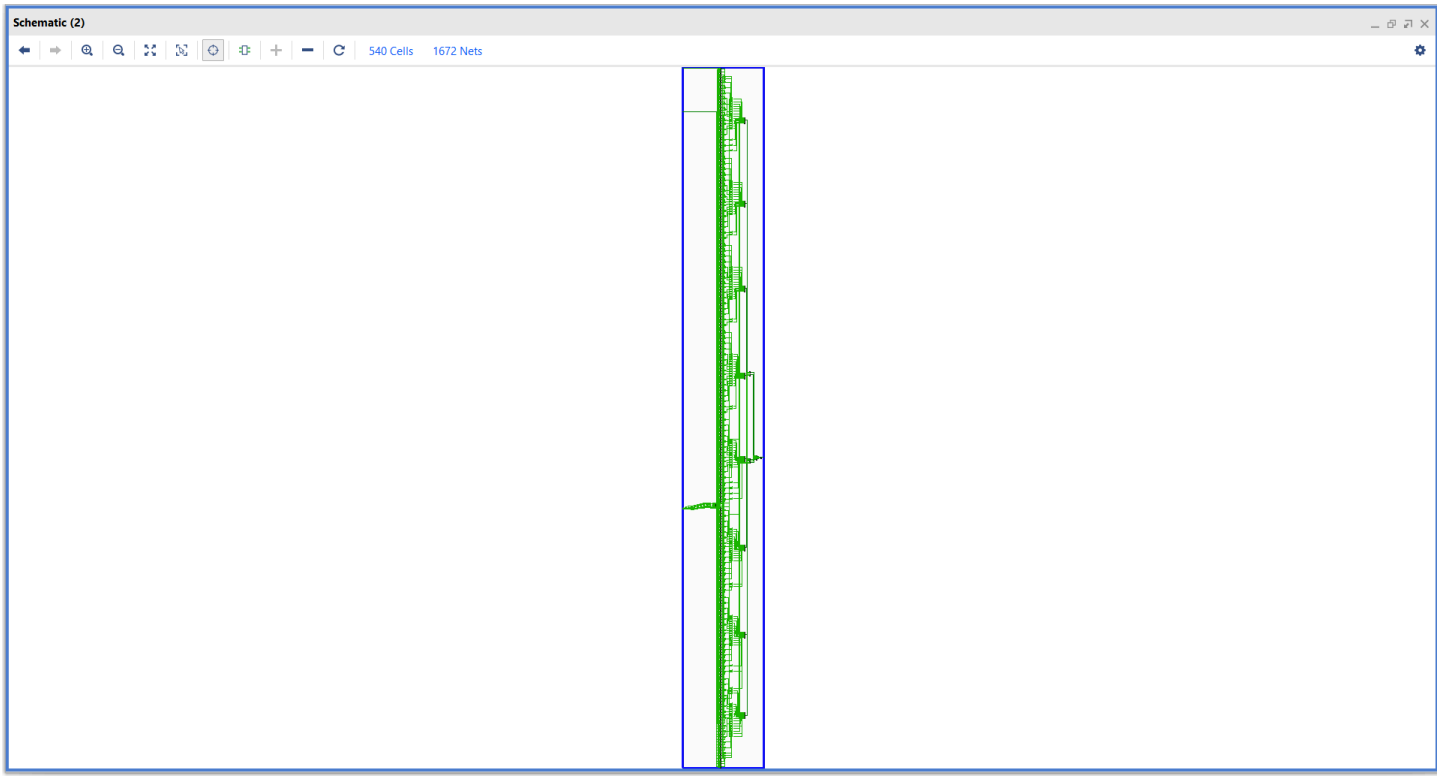
Resource	Utilization	Available	Utilization %
LUT	13475	14400	93.58
FF	6107	28800	21.20
IO	33	54	61.11
BUFG	1	32	3.13

Part 2: 8*8

1. Synthesis



Picture 27: whole schematics of design



Picture 28: MulandAddTree

For multiply and adder module, their schematics same as 4*4 ones.

2. estimation reports

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	793	14400	5.51
FF	1292	28800	4.49
IO	28	54	51.85
BUFG	1	32	3.13

Power **Summary** | On-Chip

Total On-Chip Power:	**0.111 W**
Junction Temperature:	**26.3 °C**
Thermal Margin:	58.7 °C (5.0 W)
Effective ϑJA:	11.5 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Implemented Power Report

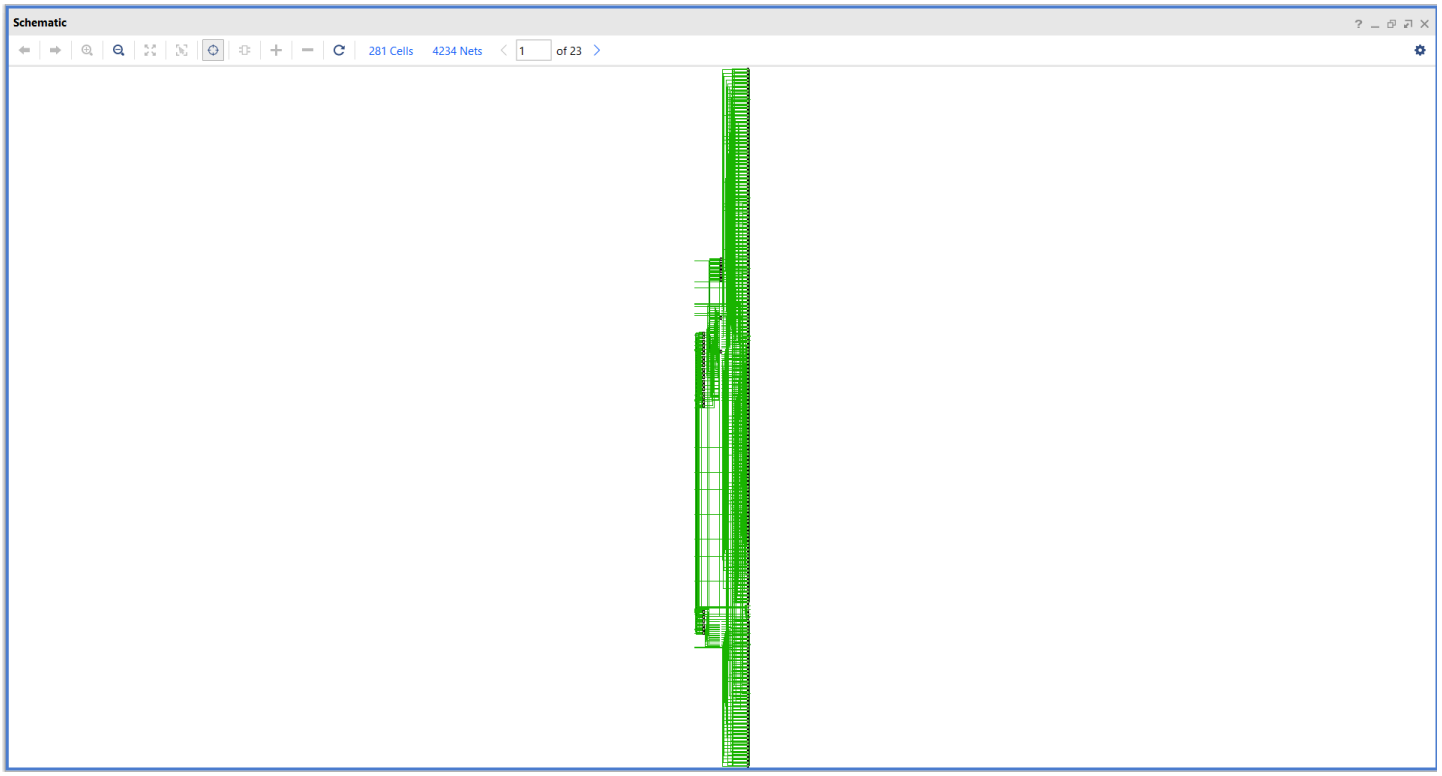
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.357 ns	Worst Hold Slack (WHS):	0.158 ns	Worst Pulse Width Slack (WPWS):	4.500 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:	1298	Total Number of Endpoints:	1298	Total Number of Endpoints:	1293

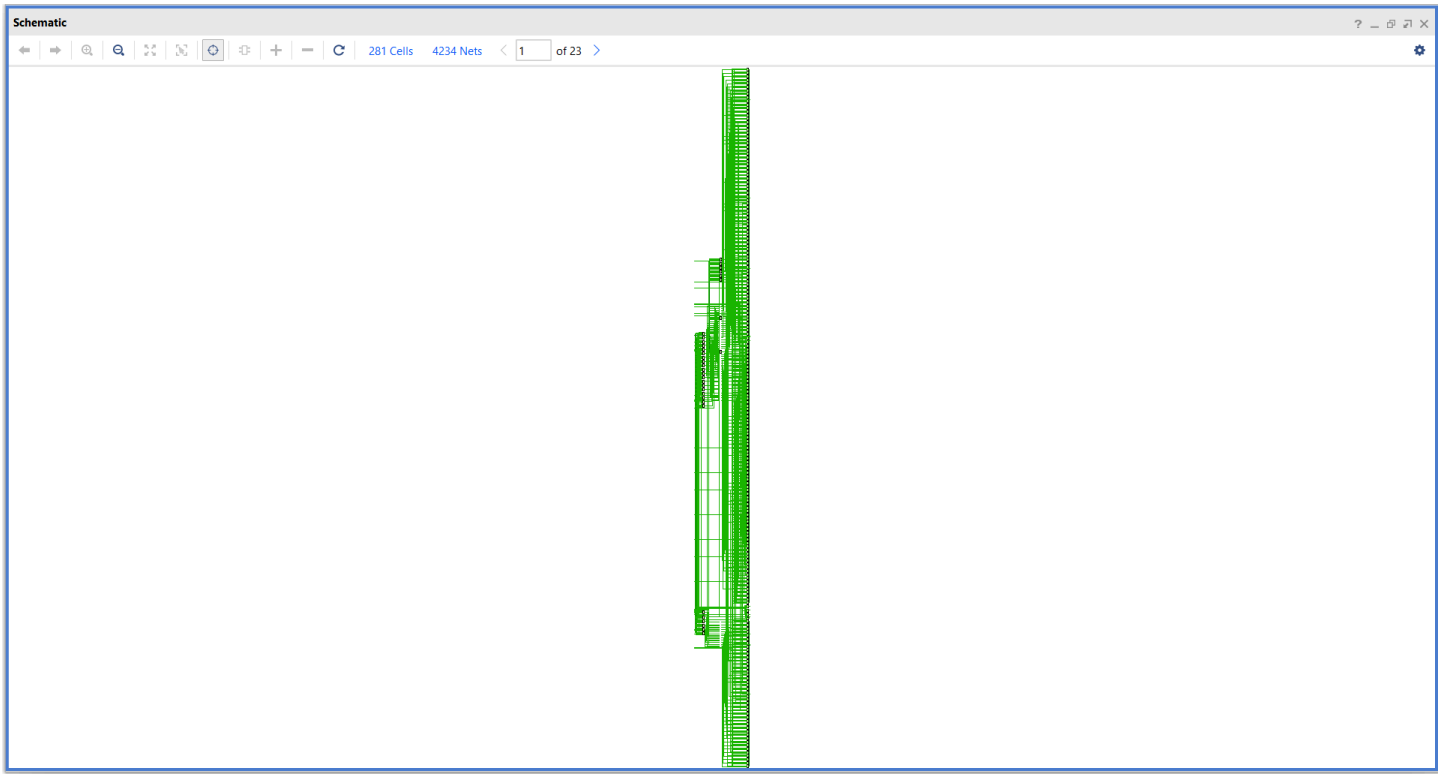
All user specified timing constraints are met.

Part 3: 16*16

1. **Synthesis**



Picture 29: whole schematics of design



Picture 30: MulandAddTree

For multiply and adder module, their schematics same as 4*4 ones.

2. estimation reports

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	3262	14400	22.65
FF	4667	28800	16.20
IO	28	54	51.85
BUFG	1	32	3.13

Power **Summary** | On-Chip

Total On-Chip Power:	**0.133 W**
Junction Temperature:	**26.5 °C**
Thermal Margin:	58.5 °C (5.0 W)
Effective ϑJA:	11.5 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Implemented Power Report

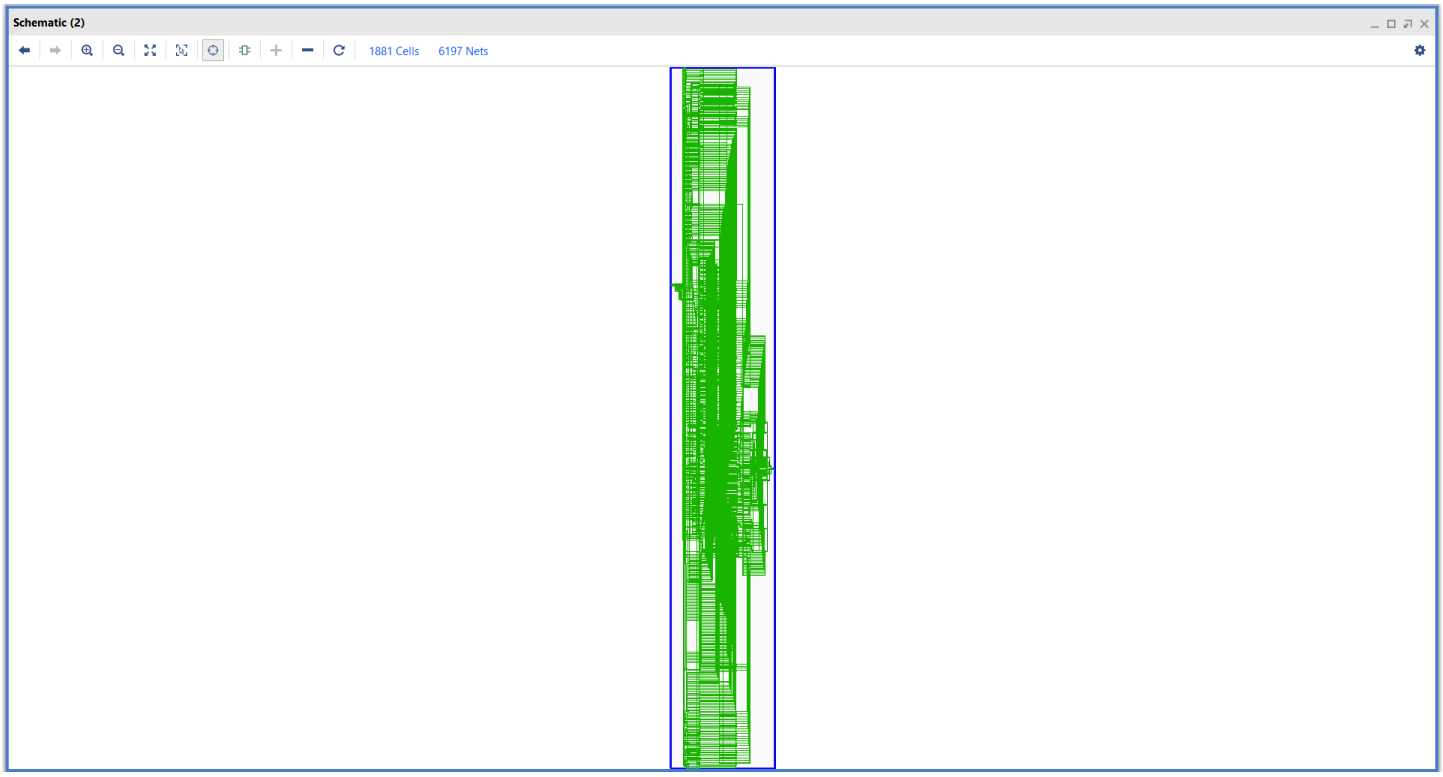
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.639 ns	Worst Hold Slack (WHS):	0.072 ns	Worst Pulse Width Slack (WPWS):	4.500 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:	8791	Total Number of Endpoints:	8791	Total Number of Endpoints:	4668

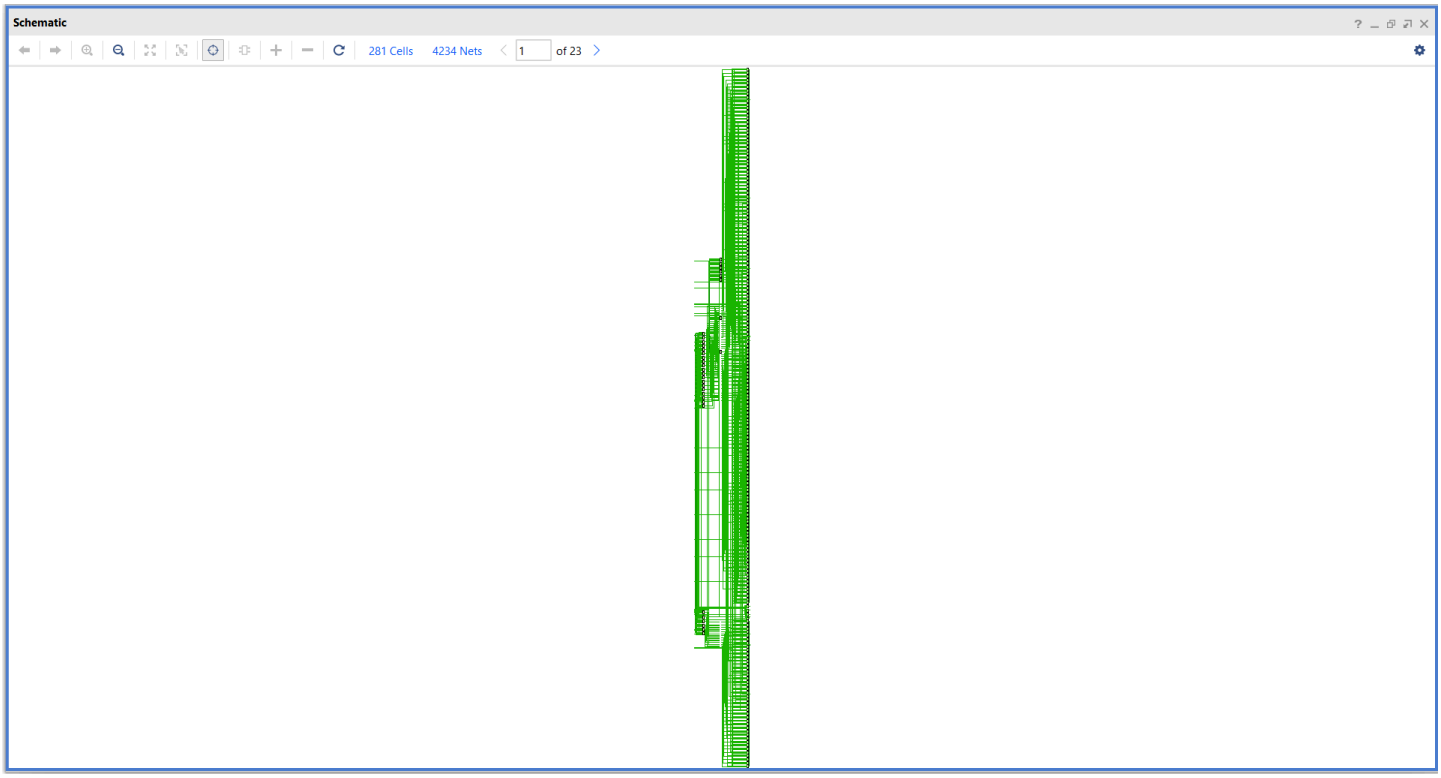
All user specified timing constraints are met.

Part 4: 32*32

1. **Synthesis**



Picture 31: whole schematics of design



Picture 30: MulandAddTree

For multiply and adder module, their schematics same as 4*4 ones.

2. estimation reports

the 32 * 32 matrixs cannot be implement on our FPGA, because the FPGA do not have enough resource.

Reason: the main reason why I cannot implement 32*32 Matrix on my FPGA is because in the design, all of the data will get from input 8 bit 1 clock, one by one, and then store them into registers. So, it is a big consumption. But I think it may not a bad design. First of all, if we want all of 32*32*8 bit data input at the same time, FPGA do not have enough pins, it is imposible. Second, if we use BRAM, it will consume a lot of time to read and write into BRAM, it is not first enough. So, at least for small matrixs, such as 4*4, my design is good enough.