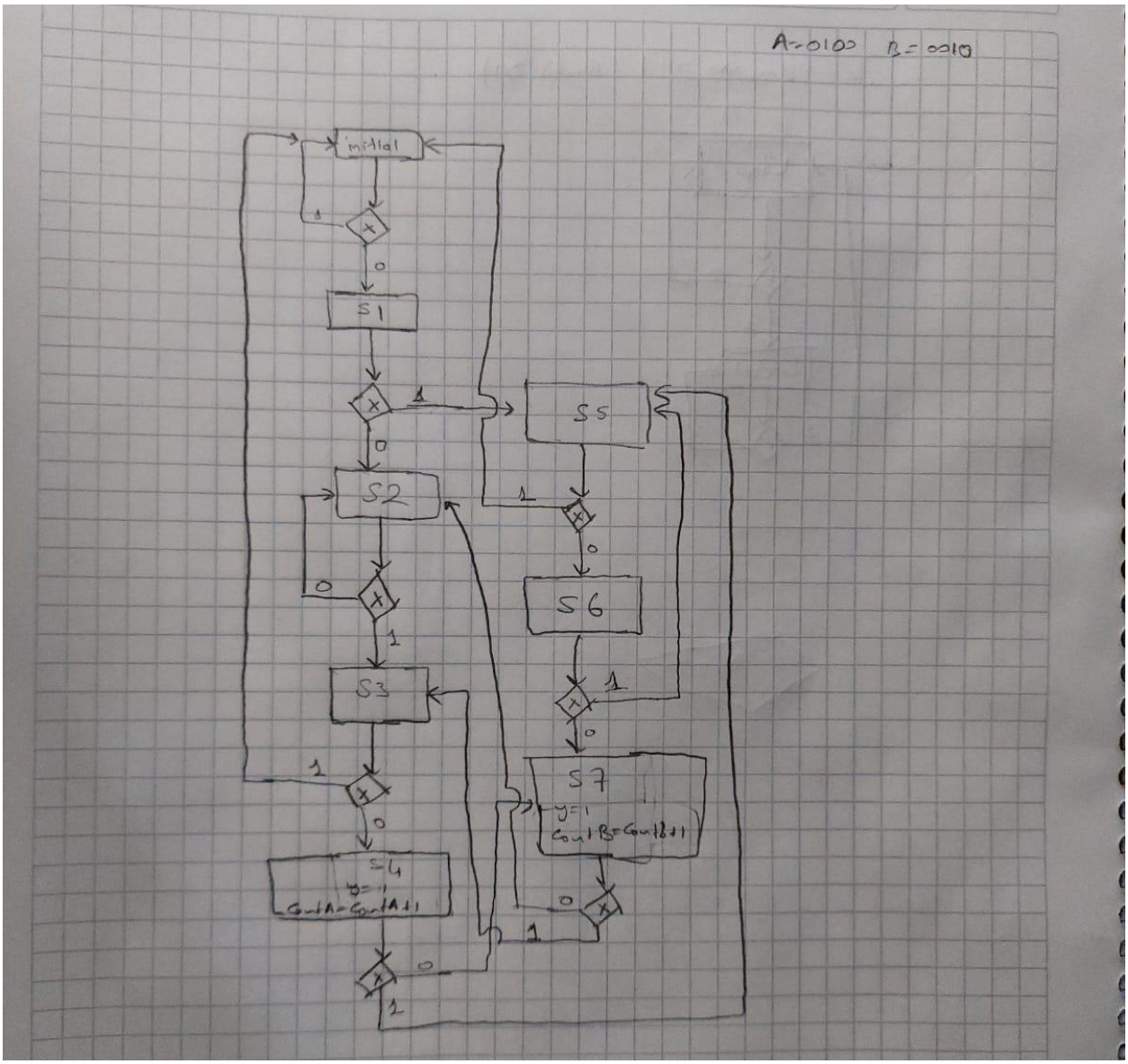
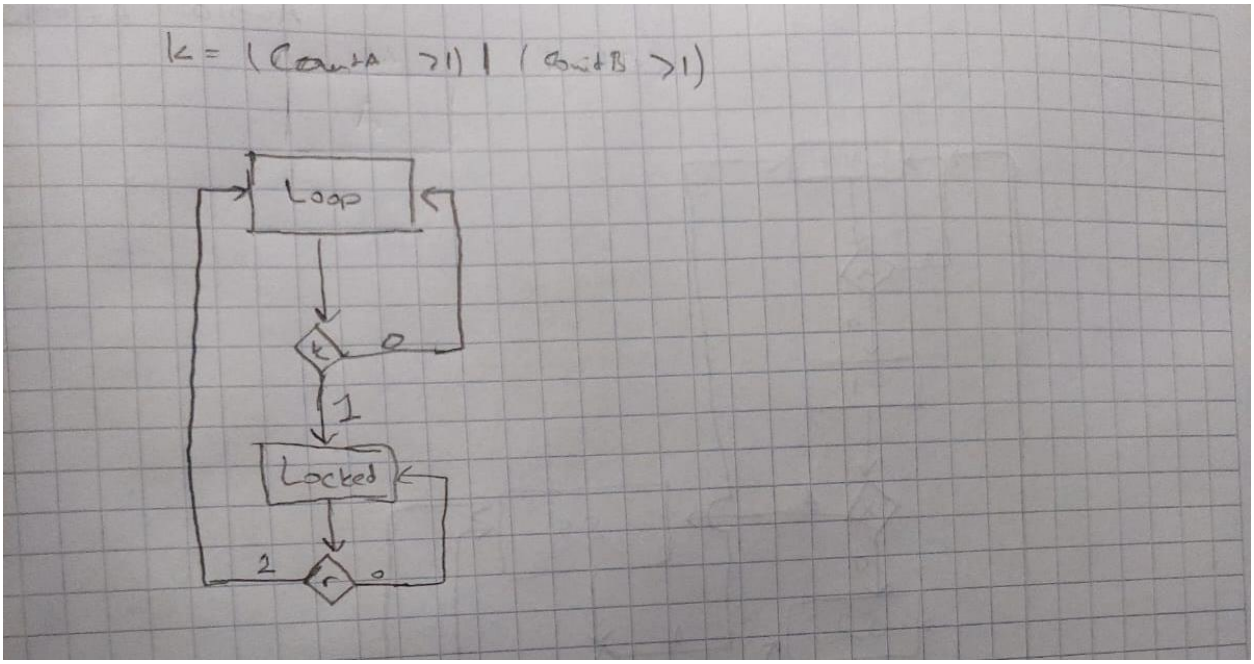
Ömer Faruk SERT

040170244

State Diagram and The Way of Approach





The way I approach to design is making states for input and at the end I have infinite loop that always has a next state even if it is itself so I can determine the outputs from this state diagram. Also, I made one more diagram for locking the output.

Encoding States

Initial = 000 S1=001 S2=011 S3=010 S4=110 S5=101 S6=100

S7=111

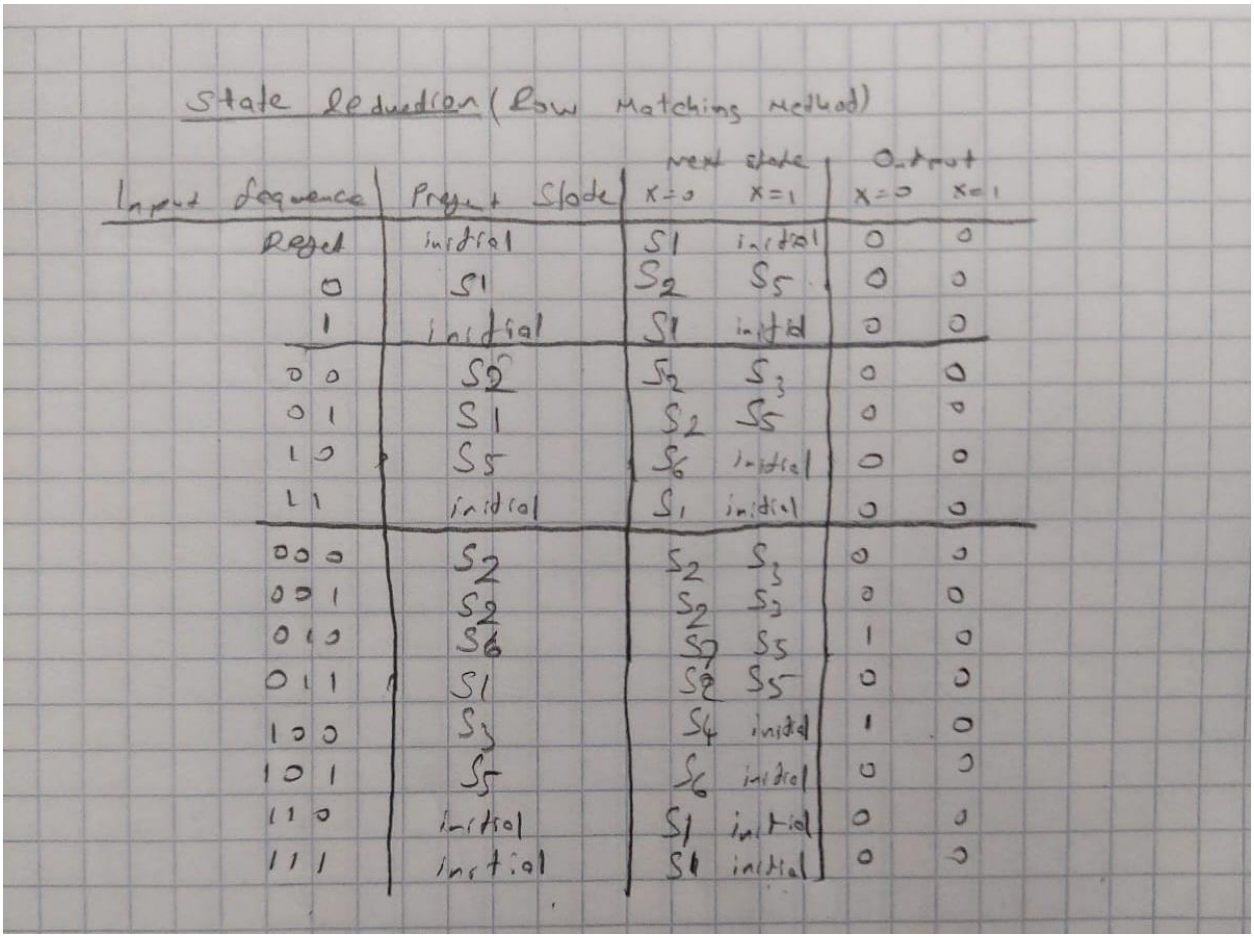
To make my encoding power efficient I needed less switching. According to this I tried to encode my states. I wanted to use gray

encoding but on the way because of the sideways it may not be seeable.

State Reduction

Row Matching Method

We write the table with input sequences, present states, next states and outputs, if we see same next state and same output in states then it means we can use one state instead of plural. This method does not give the best solution. There might be some unreduced states after applying this method.



State Reduction (Row Matching Method)

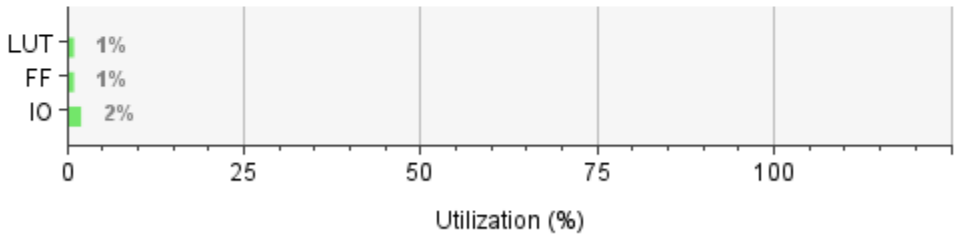
Input Sequence	Present State	Next state x=0	Next state x=1	Output x=0	Output x=1
Reset	initial	S1	initial	0	0
0	S1	S2	S5	0	0
1	initial	S1	initial	0	0
0 0	S2	S2	S3	0	0
0 1	S1	S2	S5	0	0
1 0	S5	S6	initial	0	0
1 1	initial	S1	initial	0	0
0 0 0	S2	S2	S3	0	0
0 0 1	S2	S2	S3	0	0
0 1 0	S6	S7	S5	1	0
0 1 1	S1	S8	S5	0	0
1 0 0	S3	S4	initial	1	0
1 0 1	S5	S6	initial	0	0
1 1 0	initial	S1	initial	0	0
1 1 1	initial	S1	initial	0	0

My state diagram is already reduced while I am designing it because I did not use a new state for every input, if there is suitable and available state I used this state again.

Utilization Report

Summary

Resource	Utilization	Available	Utilization %
LUT	5	63400	0.01
FF	7	126800	0.01
IO	4	210	1.90



Timing Report

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0,476 ns	Worst Hold Slack (WHS):	0,213 ns	Worst Pulse Width Slack (WPWS):	0,025 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:	7	Total Number of Endpoints:	7	Total Number of Endpoints:	8

All user specified timing constraints are met.

Maximum Clock Frequency

Maximum clock frequency is 459 MHz, I found it by changing the created clock frequency till I approach to 0 negative slack as can be seen.

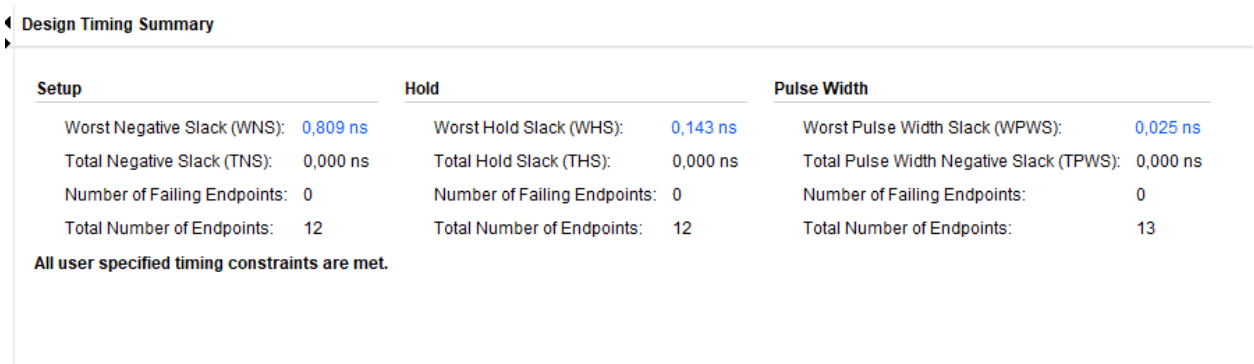
With Minimum Delay Strategy

I tried various synthesis and implementation strategies and timing constrain but I could not get a better clock frequency because of Worst Pulse Width Slack(It is caused by clock buffer), although I get a better Worst Negative Slack.

Strategies I found the best is

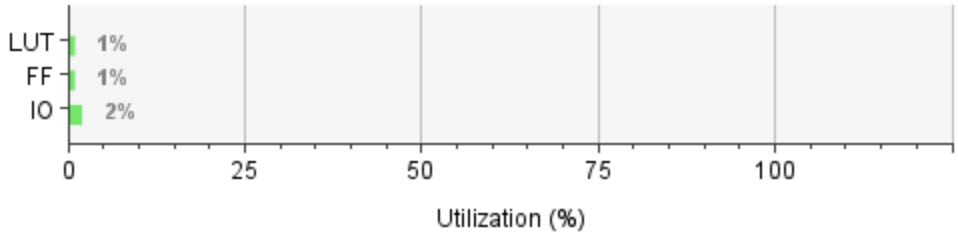
For synthesis Flow_perfoptimized_high

For implementation Performance_netdelay_low



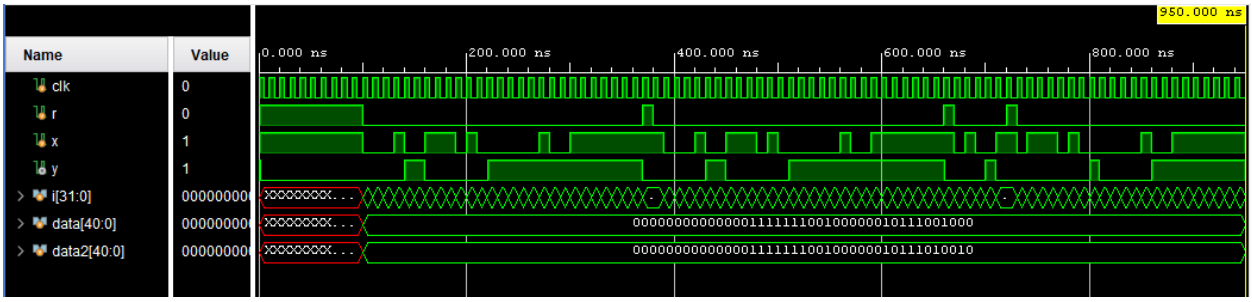
Summary

Resource	Utilization	Available	Utilization %
LUT	13	63400	0.02
FF	12	126800	0.01
IO	4	210	1.90



Simulation Results

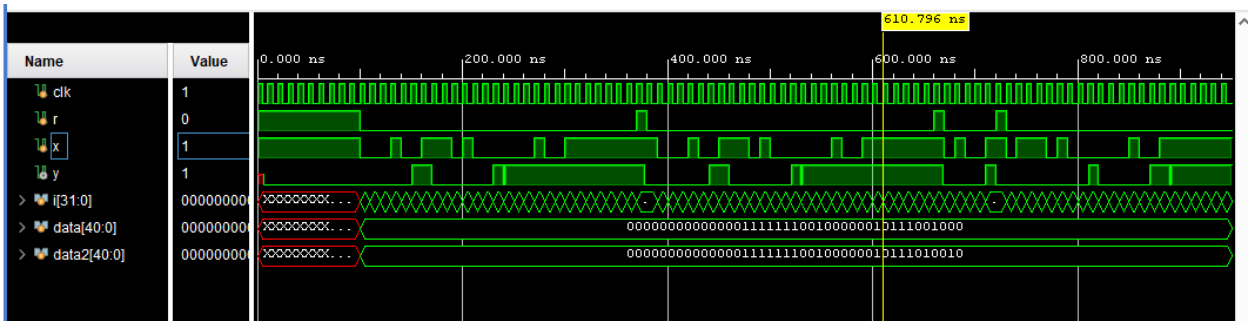
Post Implementation Functional Simulation



As can be seen in the figure machine detects first B(0100) then A(0010) and B overlaps A. After that there is 0010 and it locks its output to one. After reset high it resets its value and counter, same data sequence till the reset, after third reset it sees only one sequence and it resets

again, this scenario is to show that it can reset correctly in single occurrence situation, after fourth reset it sees 0010 and makes y=1 for one clock cycle. Then sees 0100 and 0010 overlapping each other and after that it locks its value to y=1 because it saw 0010 twice.

Post Implementation Timing Simulation



The result of the simulation is same as functional except some unwanted zeros at y=1 lock cases but that is okay because it doesn't occur at rising edge of the clock and the reason of this unwanted zeros is delays, because it don't see a sequence and that's why it want to make the output y=0, if condition makes the output y=1 in lock situation and the delay of the if condition is higher, that's why output becomes zero for a short while.

Machine Type

My machine is a Moore machine because it makes the output with only states not with the input. There is no hazardous outputs, in timing simulation unwanted zeros

are not hazardous because it makes the output correct at the rising edge of clock.