DIGITAL SYSTEM DESIGN APPLICATIONS

(CRN: 11275)

THE REPORT OF EXPERIMENT – 3

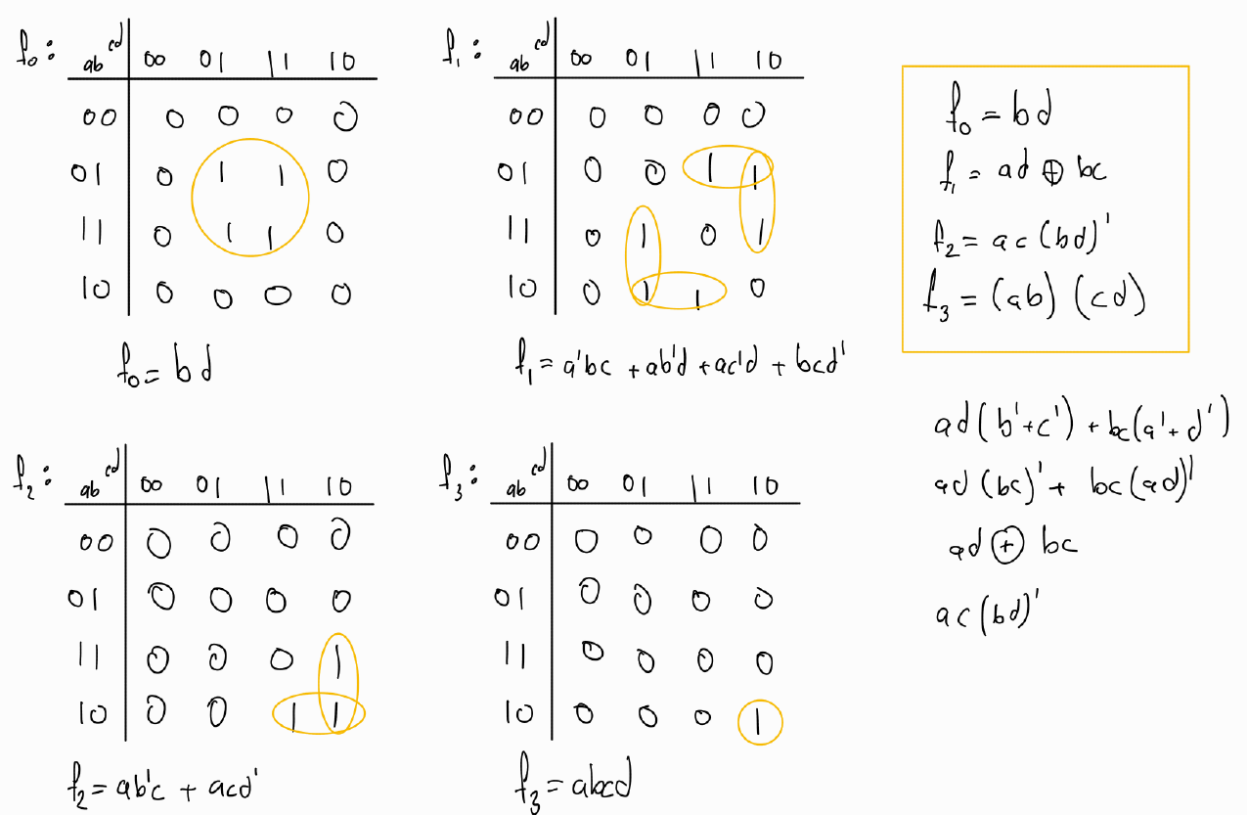
Faculty of Electrical and Electronics Engineering

Electronics and Communication Engineering

Yusuf Tekin – 040200043

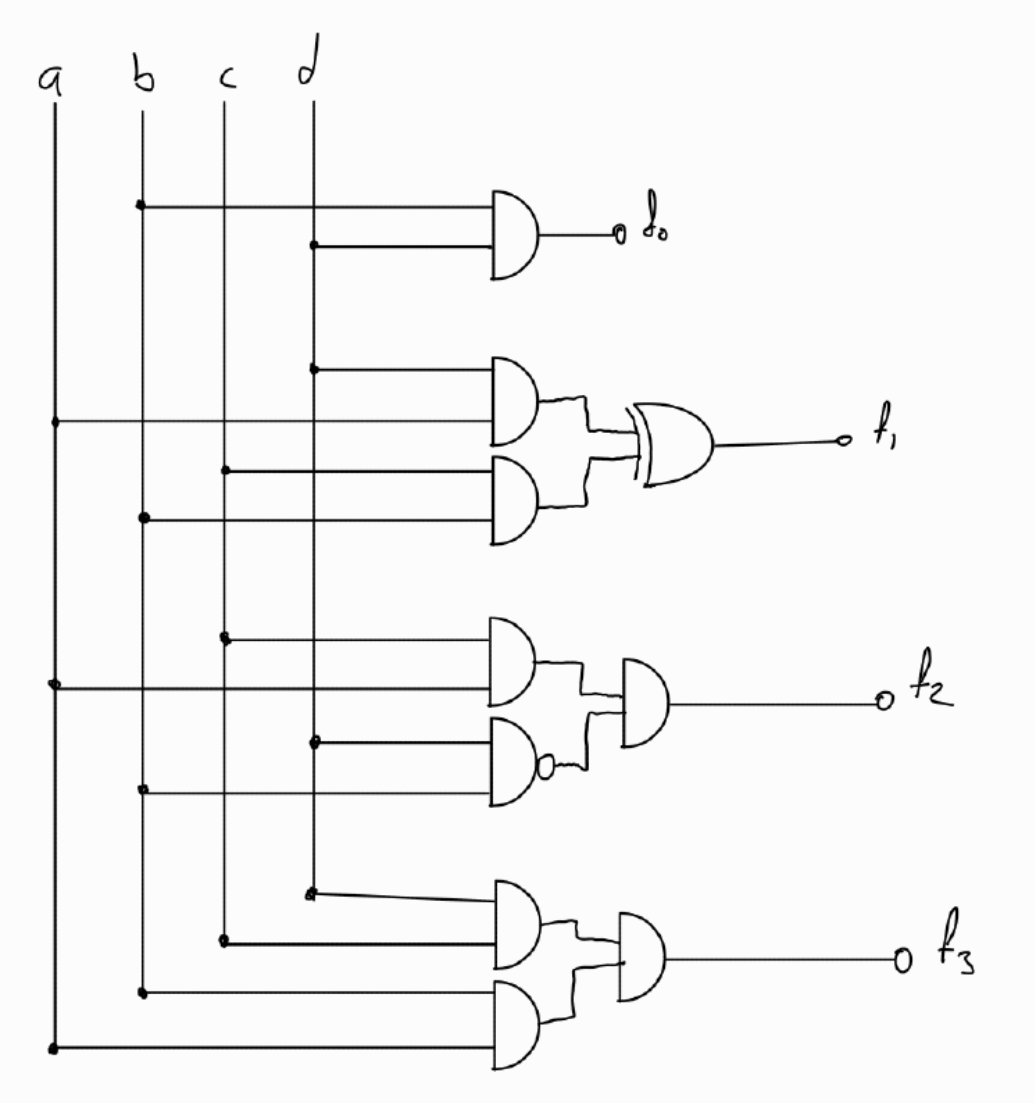
Yusuf Tekin
040200043

1. Realization with SSI Library



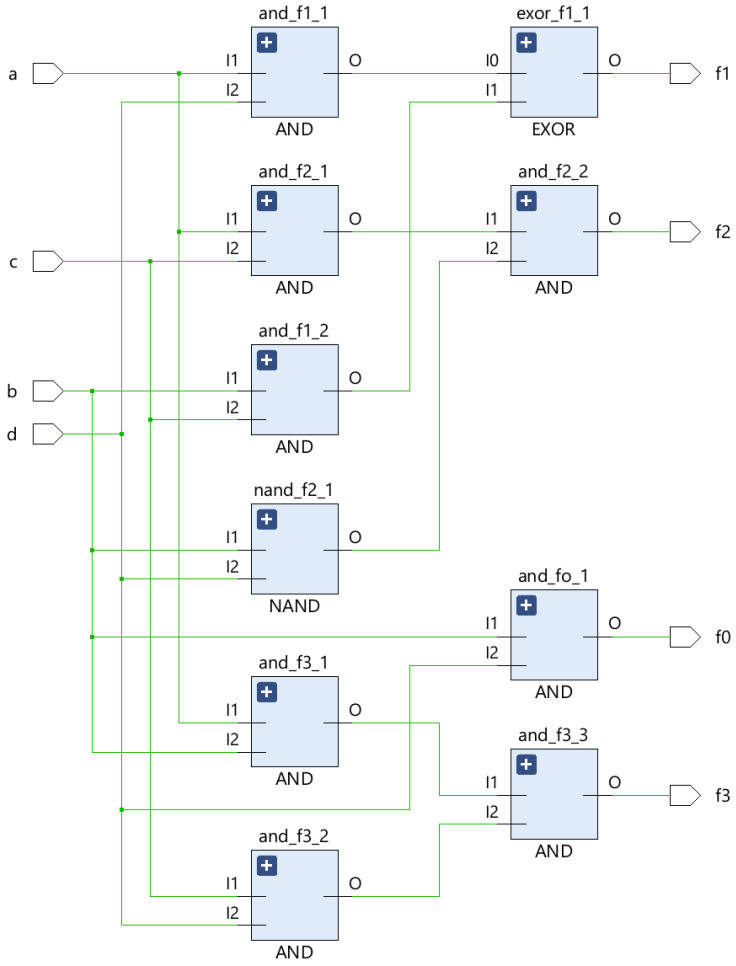
Karnaugh Maps and Boolean Expressions (with simplifications)

Yusuf Tekin
040200043



Gate Level Circuit Schematic

Yusuf Tekin
040200043



RTL Schematics of The Functions

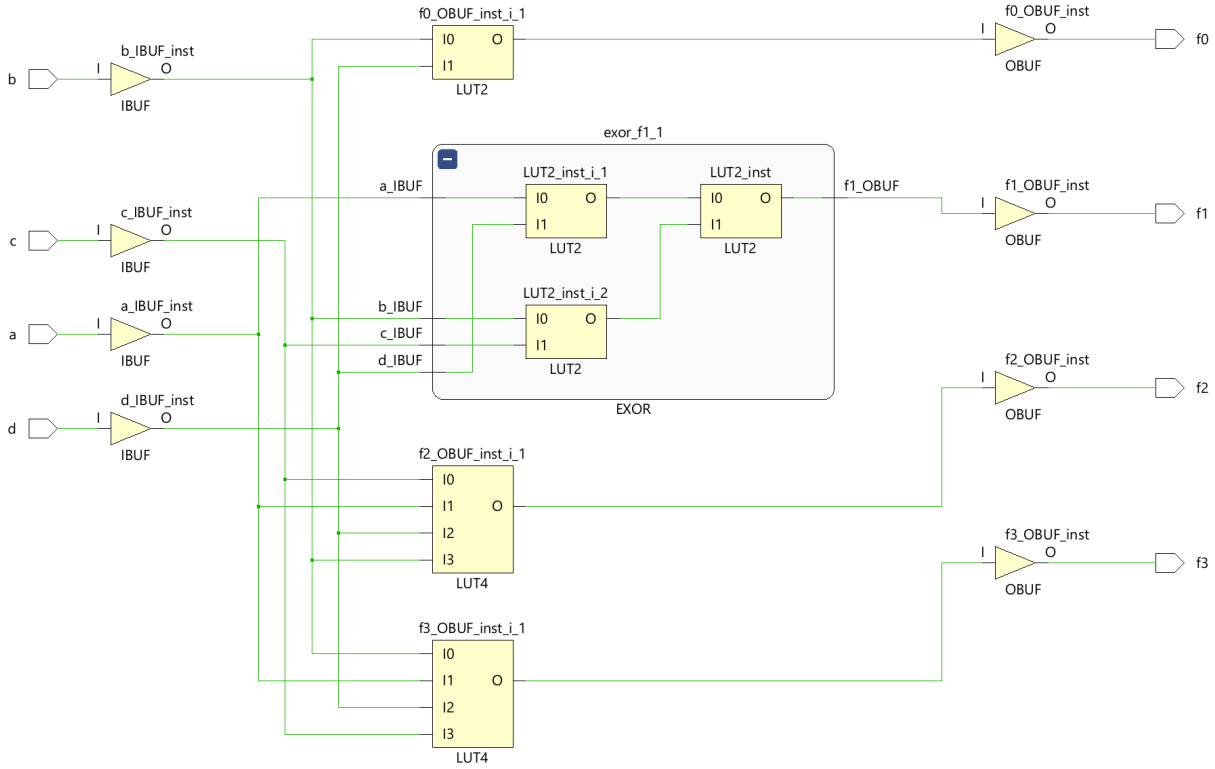
Yusuf Tekin
040200043

```
Time resolution is 1 ps
source experiment3_tb.tcl
{a,b,c,d}=0000 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0001 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0010 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0011 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0100 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0101 => {f3,f2,f1,f0} = 0001 -- TRUE
{a,b,c,d}=0110 => {f3,f2,f1,f0} = 0010 -- TRUE
{a,b,c,d}=0111 => {f3,f2,f1,f0} = 0011 -- TRUE
{a,b,c,d}=1000 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=1001 => {f3,f2,f1,f0} = 0010 -- TRUE
{a,b,c,d}=1010 => {f3,f2,f1,f0} = 0100 -- TRUE
{a,b,c,d}=1011 => {f3,f2,f1,f0} = 0110 -- TRUE
{a,b,c,d}=1100 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=1101 => {f3,f2,f1,f0} = 0011 -- TRUE
{a,b,c,d}=1110 => {f3,f2,f1,f0} = 0110 -- TRUE
{a,b,c,d}=1111 => {f3,f2,f1,f0} = 1001 -- TRUE
$finish called at time : 800 ns : File "C:/Users/jsphtkn/Vivado
Projects/sstu_experiment_3/sstu_experiment_3.srcs/sim_1/imports/Experiment_3/experiment3_tb.v" Line 52
INFO: [USF-XSim-96] XSim completed. Design snapshot 'experiment3_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:01 ; elapsed = 00:00:08 . Memory (MB): peak = 2459.781 ; gain = 40.254
```

Behavioral Simulation Console Message

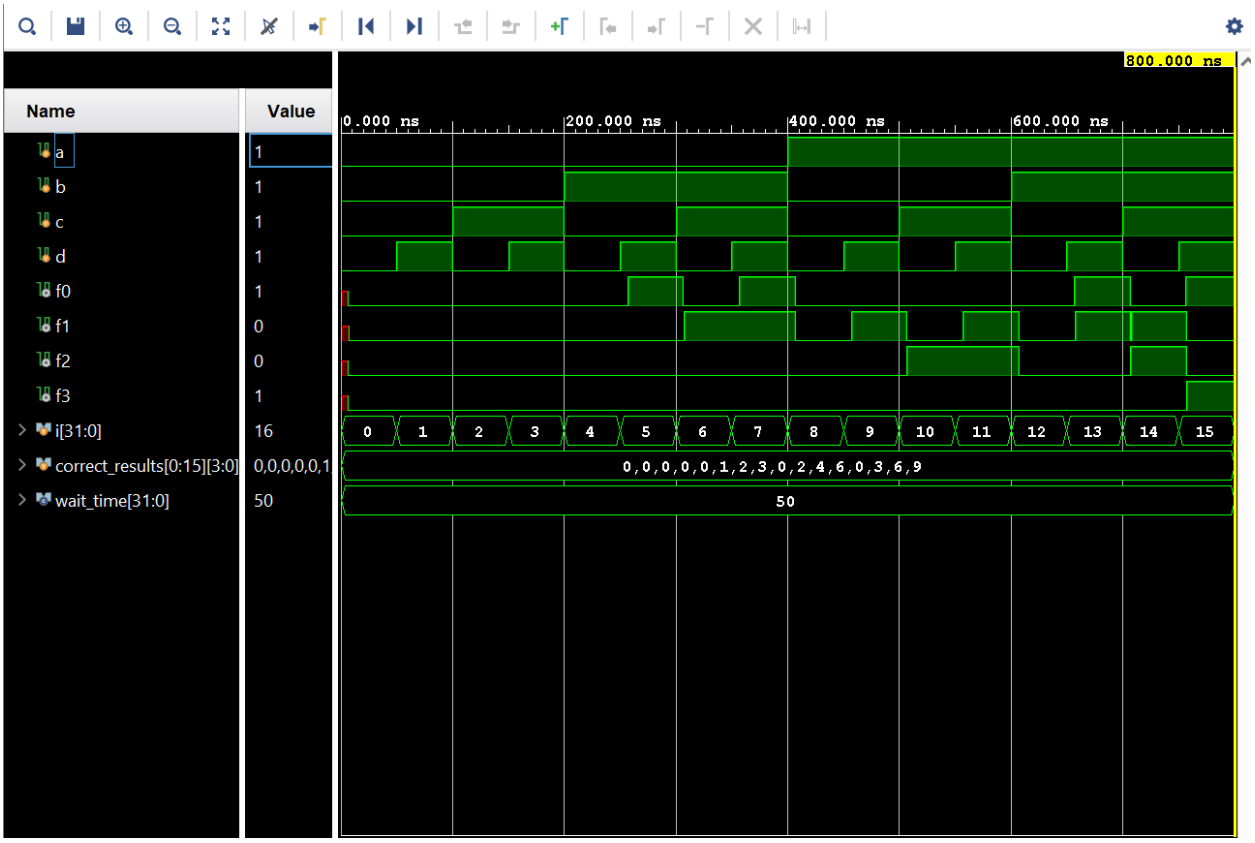
If it considered that the 4-bit output as a decimal number and two 2-bit input {a,b} and {c,d} as decimal numbers, the output is zero whenever either one of the input value is zero. Such behavior can be seen in a multiplication function. Where the inputs are named as {a,b}= X and {c,d}= Y, the multiplication function can be expressed as $F = X \times Y$. Therefore this function is a multiplication function.

Yusuf Tekin
040200043



SSI Realization Technology Schematic

Yusuf Tekin
040200043



SSI Realization Post-Synthesis Timing Simulation

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	4	3	4	d	f1	9.826	5.260	4.567	∞	input port clock			0.000
Path 2	∞	3	2	4	d	f2	9.712	5.354	4.359	∞	input port clock			0.000
Path 3	∞	3	2	4	d	f0	9.277	5.117	4.160	∞	input port clock			0.000
Path 4	∞	3	2	4	d	f3	9.218	5.119	4.099	∞	input port clock			0.000

SSI Realization Path Delays - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	∞	3	2	4	b	f3	2.497	1.524	0.973	-∞	input port clock			0.000
Path 6	∞	3	2	4	b	f2	2.684	1.590	1.094	-∞	input port clock			0.000
Path 7	∞	3	2	4	b	f0	2.705	1.522	1.183	-∞	input port clock			0.000
Path 8	∞	4	3	4	b	f1	2.779	1.586	1.193	-∞	input port clock			0.000

SSI Realization Path Delays - Hold

Yusuf Tekin
040200043

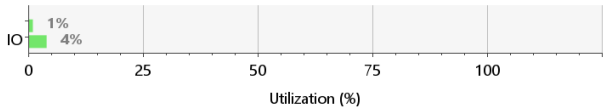
Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
a	f1	9.225	SLOW	2.784	FAST
a	f2	8.870	SLOW	2.701	FAST
a	f3	8.408	SLOW	2.507	FAST
b	f0	8.942	SLOW	2.705	FAST
b	f1	9.239	SLOW	2.779	FAST
b	f2	8.930	SLOW	2.684	FAST
b	f3	8.435	SLOW	2.497	FAST
c	f1	9.370	SLOW	2.818	FAST
c	f2	9.445	SLOW	2.858	FAST
c	f3	8.949	SLOW	2.669	FAST
d	f0	9.277	SLOW	2.894	FAST
d	f1	9.826	SLOW	3.046	FAST
d	f2	9.712	SLOW	3.017	FAST
d	f3	9.218	SLOW	2.826	FAST

SSI Realization Combinational Path Delays

Summary

Resource	Utilization	Available	Utilization %
LUT	5	32600	0.02
IO	8	210	3.81



SSI Realization Utilization Summary

Yusuf Tekin
040200043

Timing Constraint Design:

Other Path Groups - **default** - input port clock - - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	0.048	3	2	4	d	f0	8.952	5.117	3.835	9.0	input port clock		MaxDelay Path 9.000ns	0.000
Path 2	0.141	3	2	4	b	f2	8.859	5.141	3.718	9.0	input port clock		MaxDelay Path 9.000ns	0.000
Path 3	0.149	4	3	4	d	f1	8.851	5.260	3.591	9.0	input port clock		MaxDelay Path 9.000ns	0.000
Path 4	0.596	3	2	4	b	f3	8.404	5.135	3.269	9.0	input port clock		MaxDelay Path 9.000ns	0.000

SSI Realization Path Delays - Setup (9 ns max delay constraint)

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	∞	3	2	3	c	f3	2.292	1.510	0.782	-∞	input port clock			0.000
Path 6	∞	3	2	3	c	f2	2.532	1.517	1.015	-∞	input port clock			0.000
Path 7	∞	4	3	3	c	f1	2.536	1.572	0.964	-∞	input port clock			0.000
Path 8	∞	3	2	4	b	f0	2.554	1.522	1.032	-∞	input port clock			0.000

SSI Realization Path Delays - Hold (9 ns max delay constraint)

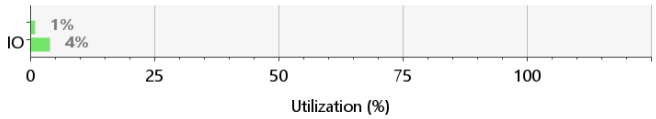
partial_input_delay

Name	Severity
Ports with partial input delay (4)	High
a	High
b	High
c	High
d	High

SSI Realization Partial Input Delay (9 ns max delay constraint)

Summary

Resource	Utilization	Available	Utilization %
LUT	6	32600	0.02
IO	8	210	3.81



SSI Realization Utilization Summary (9 ns max delay constraint)

Yusuf Tekin
040200043

set_max_delay constraint mainly focuses on setting the delays of the paths on the setup section to 9 ns. It can be seen in the constrained path delays figure above that the objective is achieved. By doing so is increased the input delays yet increased the LUTs it uses by one. Thats because while relocating the LUTs to be able to reduce the delays between them, the input delays rises due to the higher path lengths.

LOC Constraint Design:

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	3	2	4	d	f3	11.941	5.345	6.596	∞	input port clock			0.000
Path 2	∞	3	2	4	d	f2	10.836	5.126	5.710	∞	input port clock			0.000
Path 3	∞	3	2	4	d	f0	10.547	5.117	5.430	∞	input port clock			0.000
Path 4	∞	4	3	4	d	f1	9.940	5.260	4.680	∞	input port clock			0.000

SSI Realization Path Delays - Setup (LOC Constraint)

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	∞	4	3	3	a	f1	2.474	1.592	0.882	-∞	input port clock			0.000
Path 6	∞	3	2	3	a	f2	2.957	1.538	1.420	-∞	input port clock			0.000
Path 7	∞	3	2	4	b	f0	3.206	1.522	1.684	-∞	input port clock			0.000
Path 8	∞	3	2	3	a	f3	3.417	1.595	1.822	-∞	input port clock			0.000

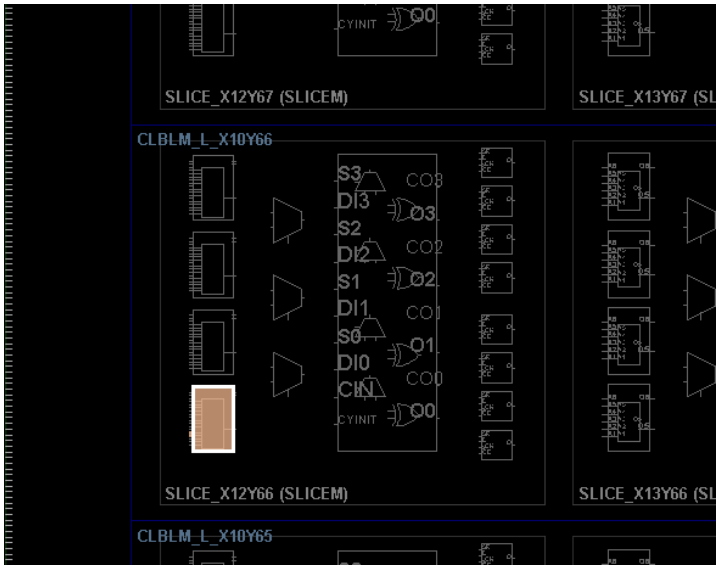
SSI Realization Path Delays - Hold (LOC Constraint)

Combinational Delays

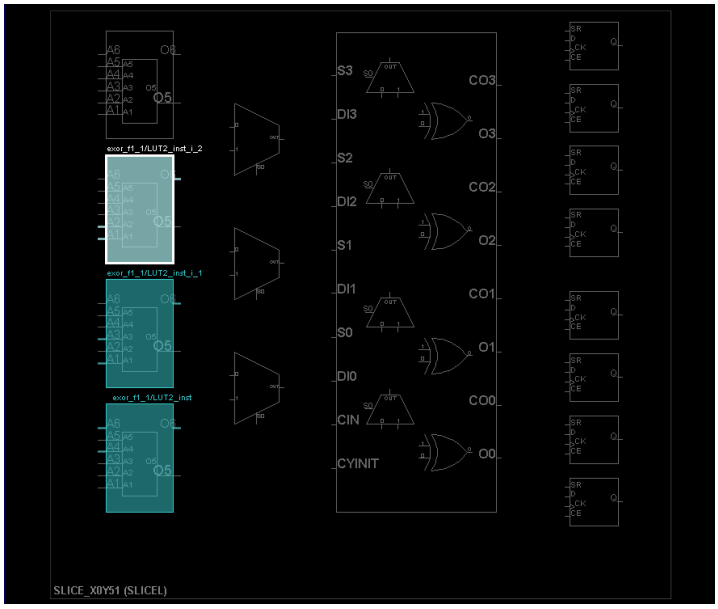
From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
a	f1	8.477	SLOW	2.474	FAST
a	f2	9.353	SLOW	2.957	FAST
a	f3	10.460	SLOW	3.417	FAST
b	f0	9.902	SLOW	3.206	FAST
b	f1	8.696	SLOW	2.589	FAST
b	f2	9.451	SLOW	3.031	FAST
b	f3	10.526	SLOW	3.496	FAST
c	f1	9.027	SLOW	2.698	FAST
c	f2	10.058	SLOW	3.226	FAST
c	f3	11.167	SLOW	3.687	FAST
d	f0	10.547	SLOW	3.578	FAST
d	f1	9.940	SLOW	3.162	FAST
d	f2	10.836	SLOW	3.658	FAST
d	f3	11.941	SLOW	4.119	FAST

SSI Realization - Combinational Delays (LOC Constraint)

Yusuf Tekin
040200043



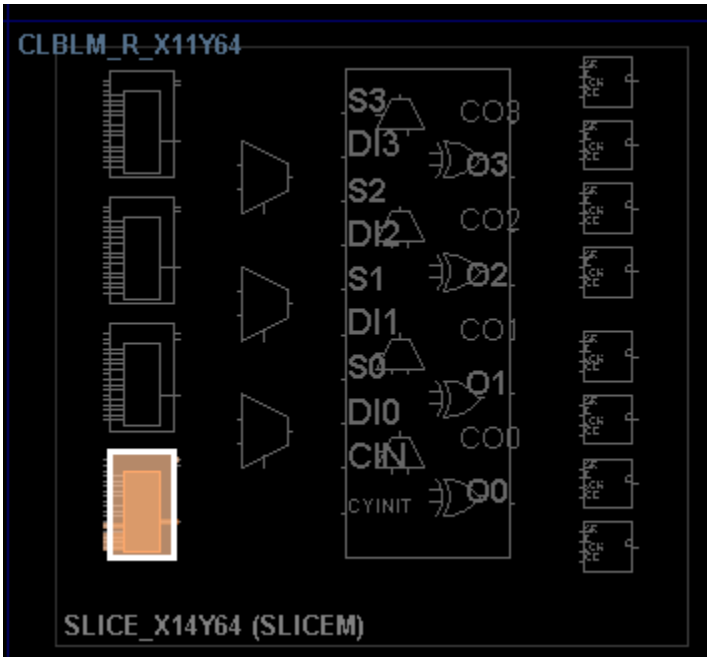
f0 relocated LUT



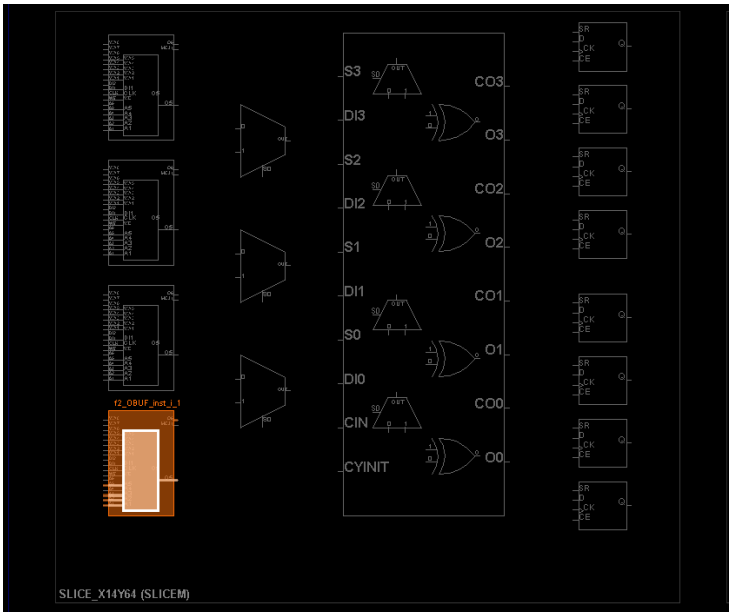
f1 LUT

Output *f1* is the output function which reduced into a Boolean expression that includes a exor gate. As it turns out that in SSI_Library the design of exor gate uses three LUT2 blocks instead of a LUT4. Therefore, it cannot be relocated to SLICE_X12Y67 position.

Yusuf Tekin
040200043



f2 relocated LUT



f3 relocated LUT

In the figures above, both f2 and f3 use the same LUTs. That's because of the similar circuitry they have. There is only a gate difference between the outputs *f2* and *f3*. If the *Gate Level Circuit Schematic (p.2)* is checked, it can be seen that the difference between these outputs is AND/NAND gates. After the implementation, Vivado (2024.1) decides that there is no

different LUTs need for that little difference and just puts a NOT gate to create such difference. This occurrence might be different for the other versions of the program.

Both LOC and Timing Constrained Design:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-1.599 ns	Worst Hold Slack (WHS):	inf	Worst Pulse Width Slack (WPWS):	NA
Total Negative Slack (TNS):	-3.151 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	NA
Number of Failing Endpoints:	3	Number of Failing Endpoints:	0	Number of Failing Endpoints:	NA
Total Number of Endpoints:	4	Total Number of Endpoints:	4	Total Number of Endpoints:	NA

Timing constraints are not met.

SSI Realization Timing Summary (with all constraints)

Other Path Groups - **default - input port clock - - Setup**

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	-1.599	3	3	c	f3	10.599	5.350	5.248	9.0	input port clock		MaxDelay Path 9.000ns	0.000
Path 2	-0.922	3	4	b	f0	9.922	5.132	4.790	9.0	input port clock		MaxDelay Path 9.000ns	0.000
Path 3	-0.630	3	4	d	f2	9.630	5.126	4.504	9.0	input port clock		MaxDelay Path 9.000ns	0.000
Path 4	0.300	4	4	d	f1	8.700	5.260	3.440	9.0	input port clock		MaxDelay Path 9.000ns	0.000

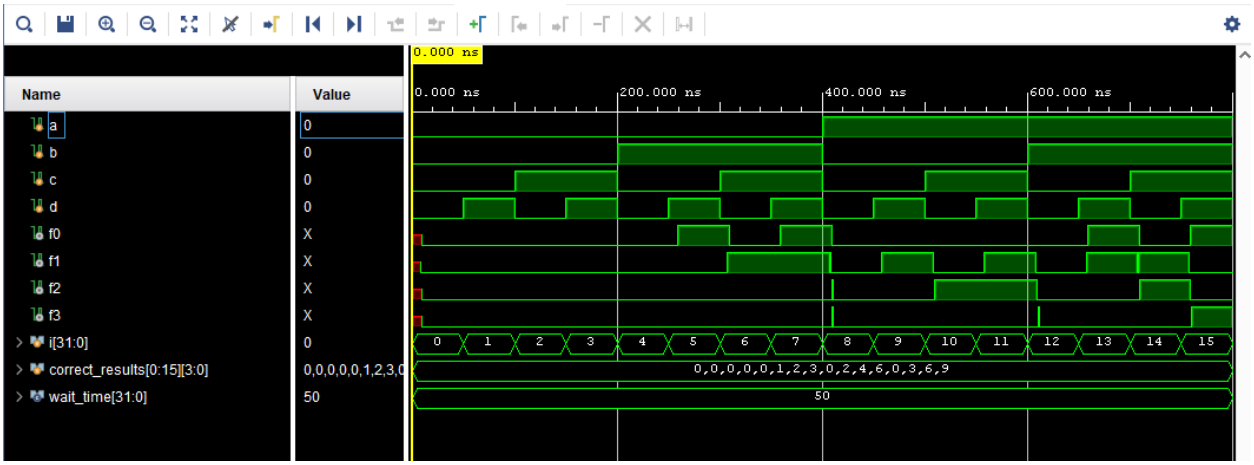
SSI Realization Path Delays - Setup (with all constraints)

Even though the *Post-Implementation Simulation* gives the same results as pre-constrained design, the tool could not manage to meet the desired design. For *Path 1, Path 2* and *Path 3,* the required max delays are not achieved.

In terms of combinational delay, out of all four designs, the best one is the design with the max delay constraint. Relocating the LUTs has resulted worse and combining both relocated and max delay constrained design has resulted the worst and even the requirements have not met.

To conclude, it is not advisable to use placement and routing manually. The default algorithm of Vivado does the job better. But that's not the case for the delay and timing modulations. For the right conditions and the reasons, the max delay constraint could be very useful.

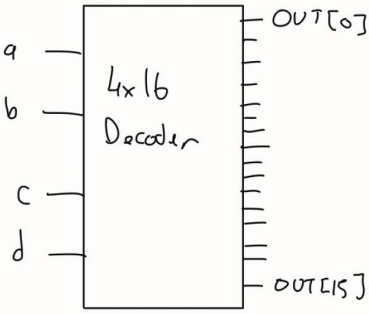
Yusuf Tekin
040200043



SSI Realization Post-Implementation Simulation (with all constraints)

There are significant output delays and glitches throughout the *Post-Implementation Simulation* which have not been observed in *Behavioral Simulation*. These glitches are mostly caused by the output nets which do not meet the requirements of the constraints and non-clocked design.

2. Realization with Decoder



Minterms:

$f_0 = bd = abcd + a'bcd + abc'd + a'bc'd$ (15, 7, 13, 5)

$f_1 = a'bc + ab'd + ac'd + bcd'$

$= a'bcd + a'bcd' + ab'c'd + ab'cd + abc'd + abcd'$ (7, 6, 9, 11, 13, 14)

$f_2 = ab'cd' + ab'cd + abcd'$ (10, 11, 14)

$f_3 = abcd$ (15)

IN3	IN2	IN1	IN0	O15	O14	O13	O12	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0	Minterm
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	a'b'c'd'
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	a'b'c'd
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	a'b'cd'
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	a'b'cd
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	a'bc'd'
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	a'bc'd
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	a'bcd'
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	a'bcd
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	ab'c'd'
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	ab'c'd
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	ab'cd'
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	ab'cd
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	abc'd'
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	abc'd
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	abcd'
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	abcd

$f_0 = OUT[5] + OUT[7] + OUT[13] + OUT[15]$

$f_1 = OUT[6] + OUT[7] + OUT[9] + OUT[11] + OUT[13] + OUT[14]$

$f_2 = OUT[10] + OUT[11] + OUT[14]$

$f_3 = OUT[15]$

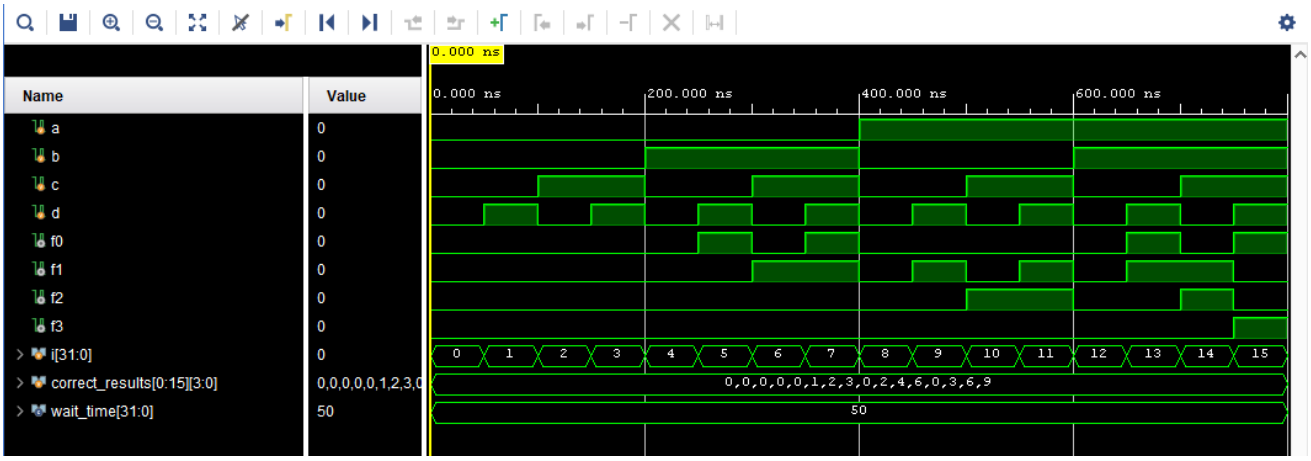
Decoder Minterms, Schematic and Logical Functions

Yusuf Tekin
040200043

```
{a,b,c,d}=0000 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0001 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0010 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0011 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0100 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0101 => {f3,f2,f1,f0} = 0001 -- TRUE
{a,b,c,d}=0110 => {f3,f2,f1,f0} = 0010 -- TRUE
{a,b,c,d}=0111 => {f3,f2,f1,f0} = 0011 -- TRUE
{a,b,c,d}=1000 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=1001 => {f3,f2,f1,f0} = 0010 -- TRUE
{a,b,c,d}=1010 => {f3,f2,f1,f0} = 0100 -- TRUE
{a,b,c,d}=1011 => {f3,f2,f1,f0} = 0110 -- TRUE
{a,b,c,d}=1100 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=1101 => {f3,f2,f1,f0} = 0011 -- TRUE
{a,b,c,d}=1110 => {f3,f2,f1,f0} = 0110 -- TRUE
{a,b,c,d}=1111 => {f3,f2,f1,f0} = 1001 -- TRUE
$finish called at time : 800 ns : File "C:/Users/jsphtkn/Vivado
Projects/sstu_experiment_3/sstu_experiment_3.srcs/sim_1/imports/Experiment_3/experiment3_tb.v" Line 52
INFO: [USF-XSim-96] XSim completed. Design snapshot 'experiment3_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:01 ; elapsed = 00:00:05 . Memory (MB): peak = 2673.414 ; gain = 0.000
```

Decoder Realization TCL Console - Simulation

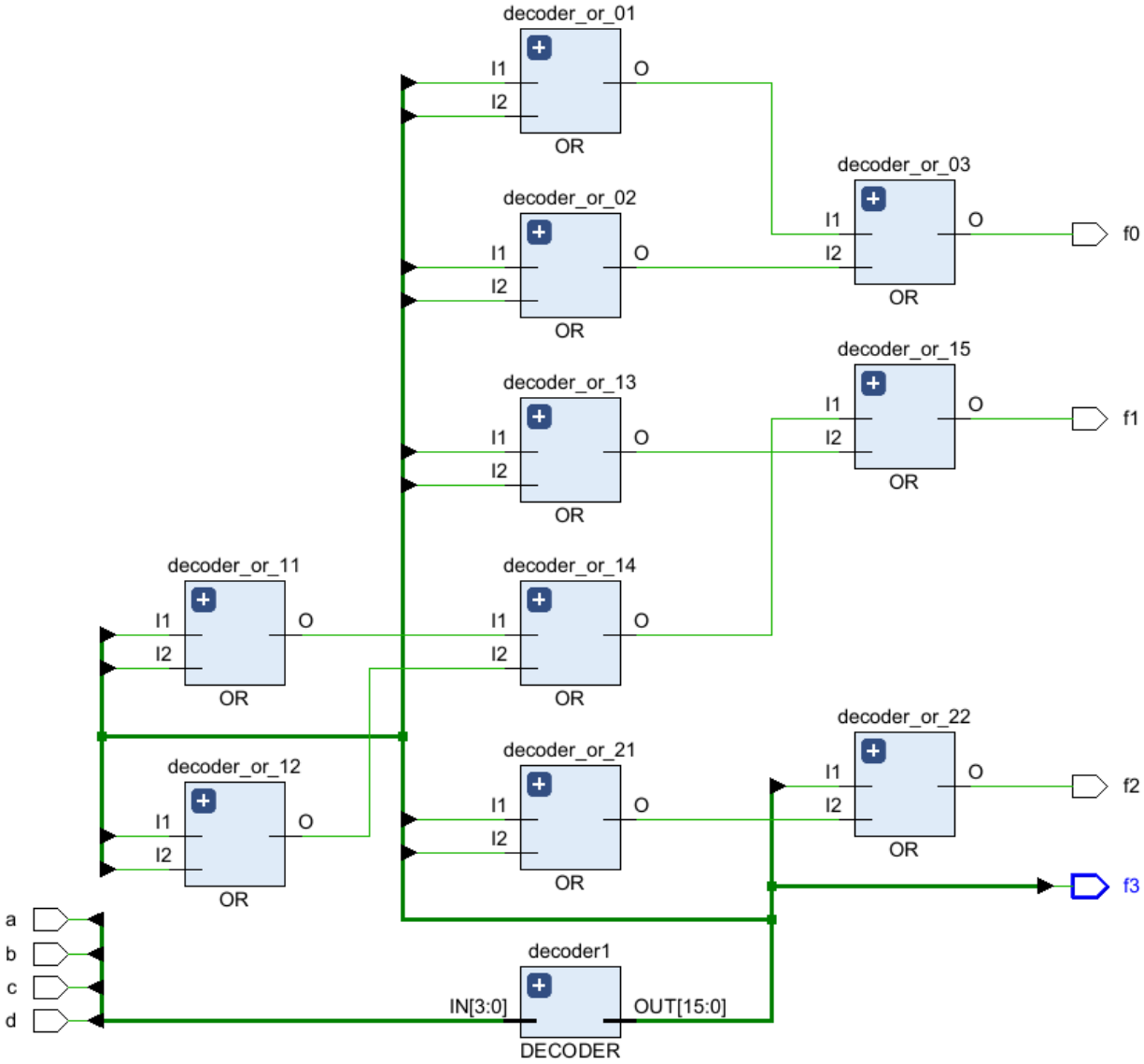
Yusuf Tekin
040200043



Decoder Realization Behavioral Simulation

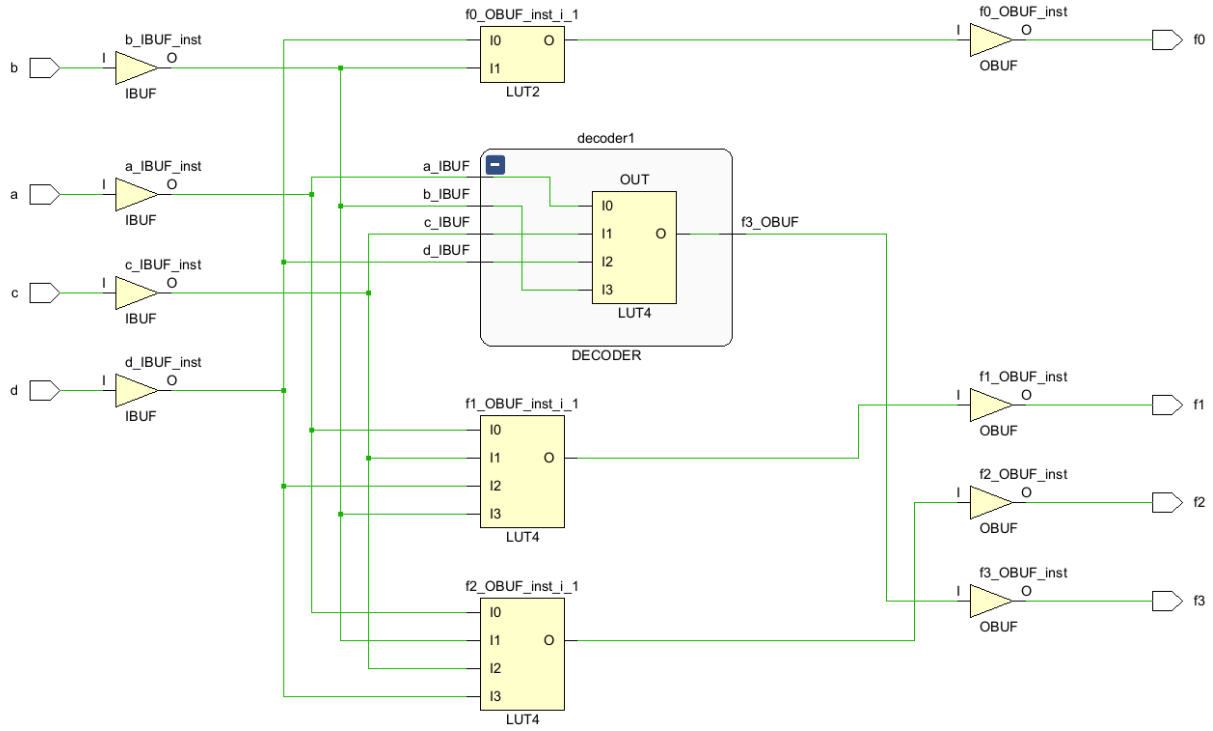
As can be both seen in the *TCL Console* and *Behavioral Simulation* graph, the design works as intended which is a multiplication block.

Yusuf Tekin
040200043



Decoder Realization RTL Schematic

Yusuf Tekin
040200043



Decoder Realization Technology Schematic

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
a	f1	8.566	SLOW	2.578	FAST
a	f2	8.647	SLOW	2.631	FAST
a	f3	8.408	SLOW	2.511	FAST
b	f0	9.059	SLOW	2.743	FAST
b	f1	8.568	SLOW	2.534	FAST
b	f2	8.617	SLOW	2.589	FAST
b	f3	8.589	SLOW	2.544	FAST
c	f1	8.798	SLOW	2.621	FAST
c	f2	8.846	SLOW	2.680	FAST
c	f3	8.596	SLOW	2.548	FAST
d	f0	9.703	SLOW	3.045	FAST
d	f1	9.421	SLOW	2.894	FAST
d	f2	9.468	SLOW	2.951	FAST
d	f3	9.203	SLOW	2.819	FAST

Decoder Realization Path Delays - Combinational Delays

Yusuf Tekin
040200043

Unconstrained Paths - NONE - NONE - Setup

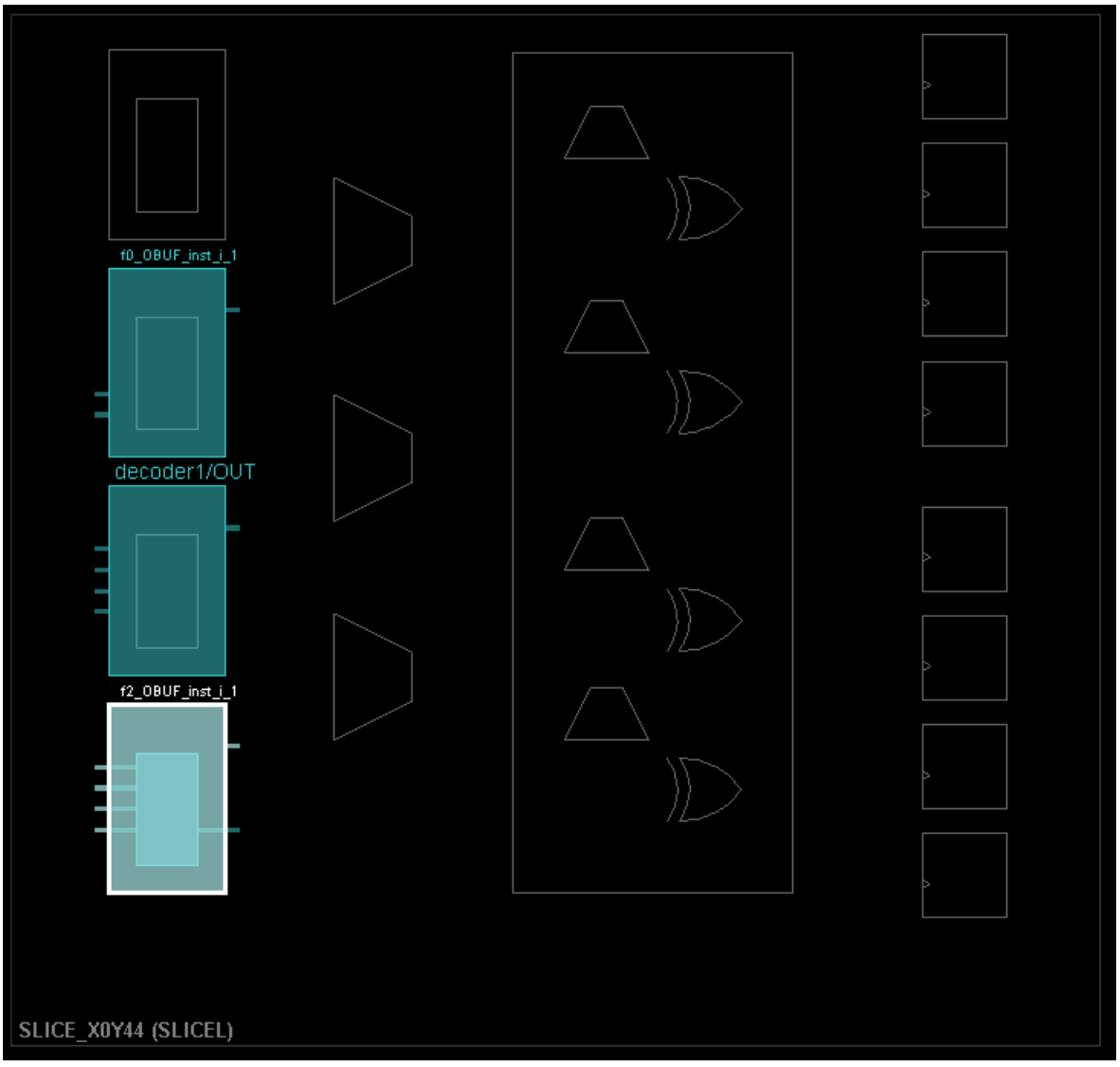
Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	3	2	4	d	f0	9.703	5.117	4.586	∞	input port clock			0.000
Path 2	∞	3	2	4	d	f2	9.468	5.126	4.342	∞	input port clock			0.000
Path 3	∞	3	2	4	d	f1	9.421	5.366	4.055	∞	input port clock			0.000
Path 4	∞	3	2	4	d	f3	9.203	5.119	4.084	∞	input port clock			0.000

Decoder Realization Path Delays - Setup

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	∞	3	2	3	a	f3	2.511	1.531	0.980	-∞	input port clock			0.000
Path 6	∞	3	2	4	b	f1	2.534	1.604	0.930	-∞	input port clock			0.000
Path 7	∞	3	2	4	b	f2	2.589	1.531	1.058	-∞	input port clock			0.000
Path 8	∞	3	2	4	b	f0	2.743	1.522	1.221	-∞	input port clock			0.000

Decoder Realization Path Delays - Hold

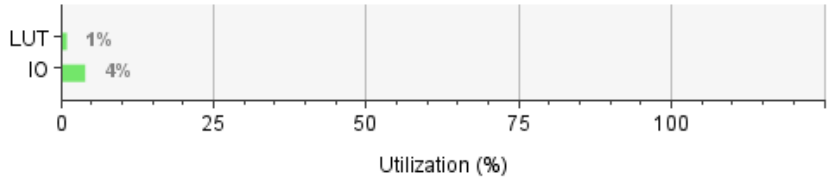


FPGA Device Overview of the Placed Design

Yusuf Tekin
040200043

Between this decoder design and previous SSI design there is no difference in terms of the used cell type but in this design the LUTs are not distributed as previous ones.

Resource	Utilization	Available	Utilization %
LUT	3	32600	0.01
IO	8	210	3.81



Decoder Realization - Utilization Summary

As it can be seen in the figures, the amount of LUTs in-use is drop to 3 from 5 (or 6 depending on the constraints). The number of I/O ports does not change as expected. Because there are still four inputs and four outputs as the first design.

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
a	f1	8.324	SLOW
a	f2	8.481	SLOW
a	f3	8.264	SLOW
b	f0	8.267	SLOW
b	f1	8.381	SLOW
b	f2	8.625	SLOW
b	f3	7.905	SLOW
c	f1	8.101	SLOW
c	f2	8.648	SLOW
c	f3	8.127	SLOW
d	f0	8.798	SLOW
d	f1	8.588	SLOW
d	f2	8.686	SLOW
d	f3	8.320	SLOW

Combination Delays (after 6ns max delay)

Yusuf Tekin
040200043

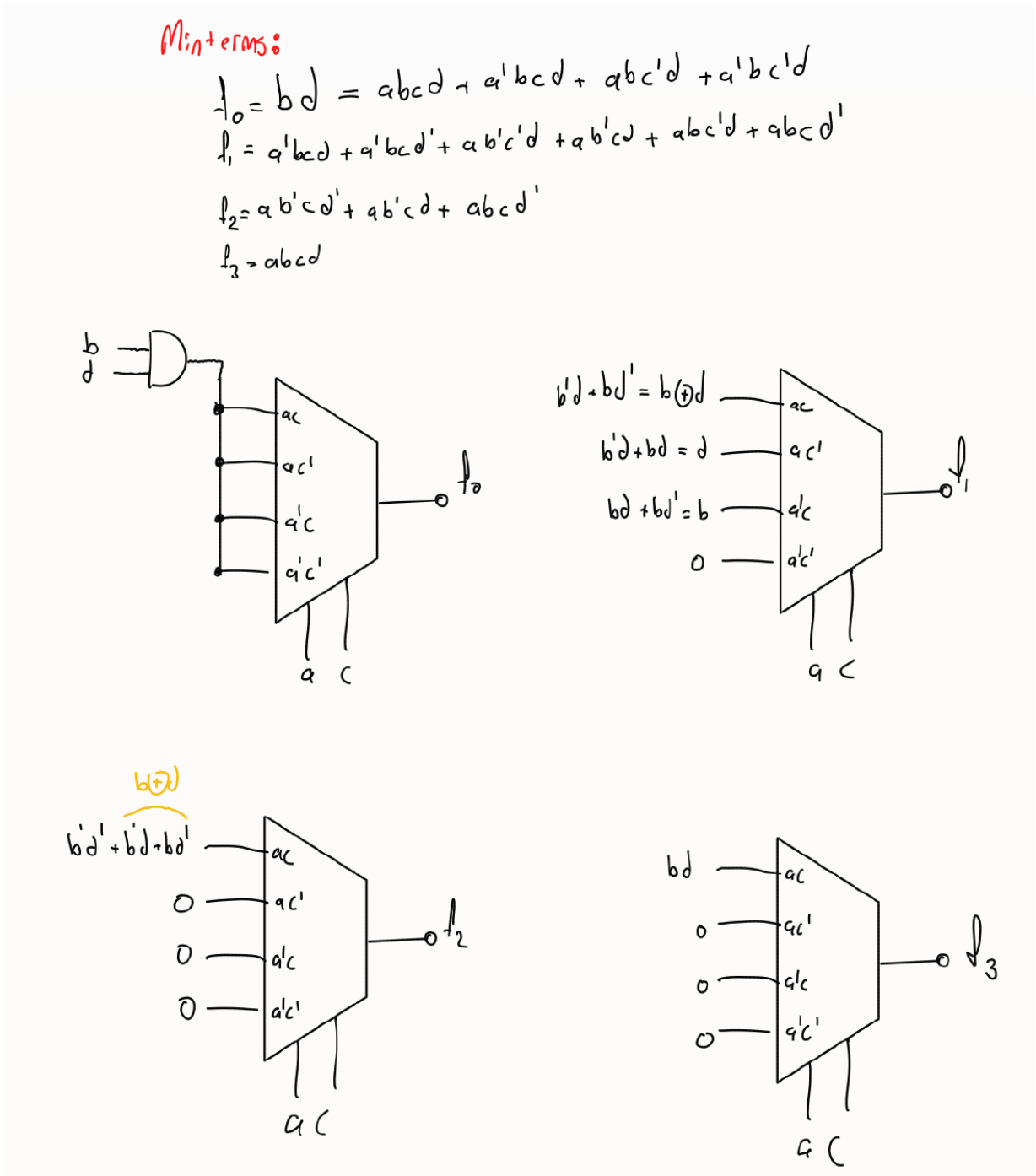
Other Path Groups - **default** - input port clock - - Setup

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	-2.798	3	2	4	d	f0	8.798	5.117	3.682	6.0	input port clock		MaxDelay Path 6.000ns	0.000
Path 2	-2.686	3	2	4	d	f2	8.686	5.126	3.560	6.0	input port clock		MaxDelay Path 6.000ns	0.000
Path 3	-2.588	3	2	4	d	f1	8.588	5.136	3.452	6.0	input port clock		MaxDelay Path 6.000ns	0.000
Path 4	-2.320	3	2	4	d	f3	8.320	5.119	3.201	6.0	input port clock		MaxDelay Path 6.000ns	0.000

Path Delays - Setup (after 6ns max delay)

Yusuf Tekin
040200043

3. Realization with MUX

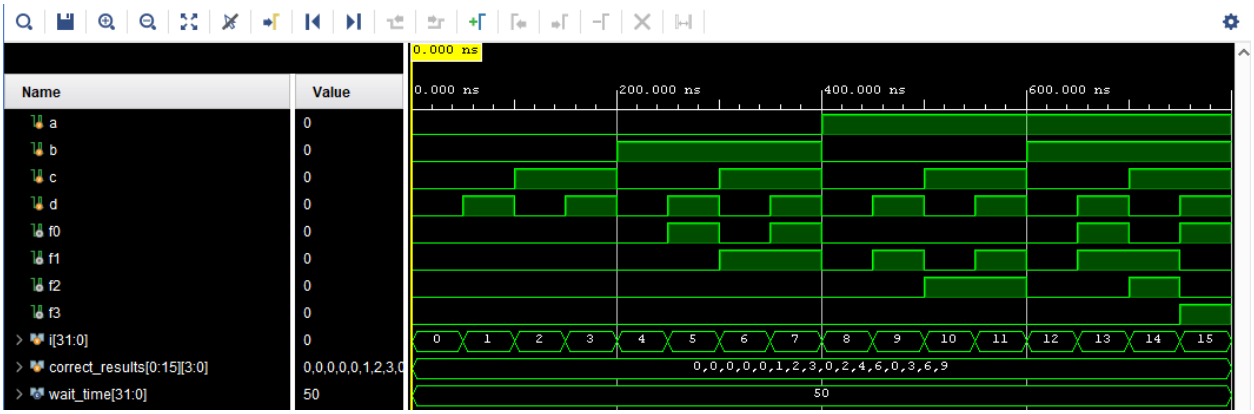


MUX Based Schematic and It's Minterms

Yusuf Tekin
040200043

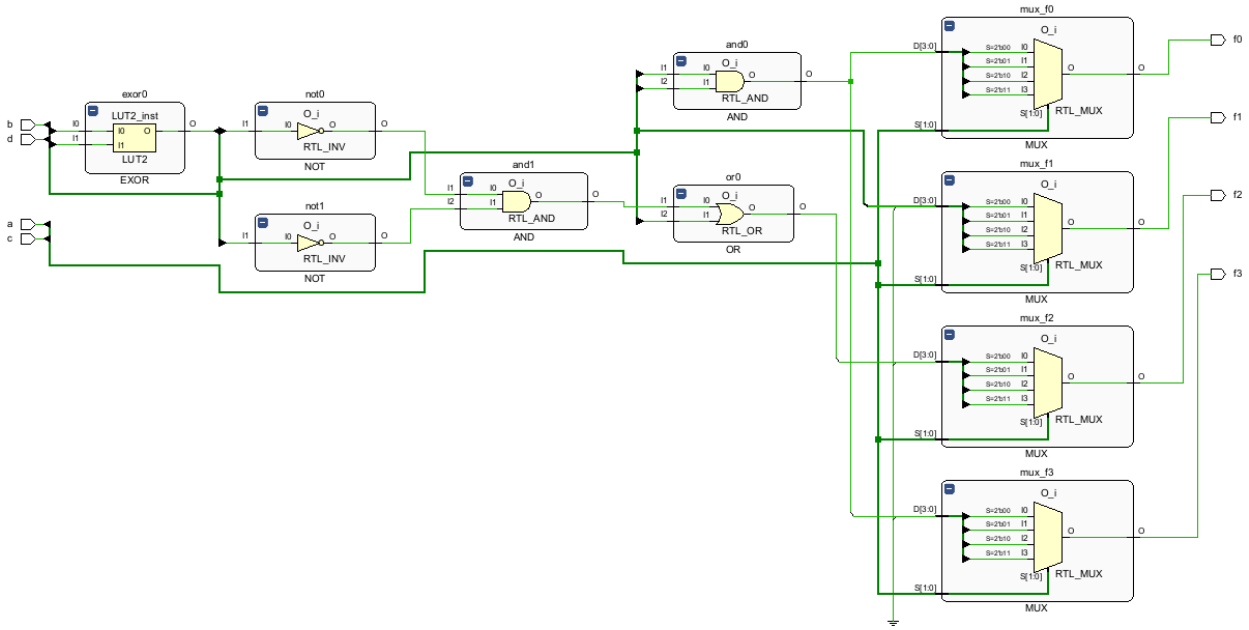
```
{a,b,c,d}=0000 => {f3,f2,f1,f0} = 0000 -- TRUE

{a,b,c,d}=0001 => {f3,f2,f1,f0} = 0000 -- TRUE

{a,b,c,d}=0010 => {f3,f2,f1,f0} = 0000 -- TRUE

{a,b,c,d}=0011 => {f3,f2,f1,f0} = 0000 -- TRUE

{a,b,c,d}=0100 => {f3,f2,f1,f0} = 0000 -- TRUE

{a,b,c,d}=0101 => {f3,f2,f1,f0} = 0001 -- TRUE

{a,b,c,d}=0110 => {f3,f2,f1,f0} = 0010 -- TRUE

{a,b,c,d}=0111 => {f3,f2,f1,f0} = 0011 -- TRUE

{a,b,c,d}=1000 => {f3,f2,f1,f0} = 0000 -- TRUE

{a,b,c,d}=1001 => {f3,f2,f1,f0} = 0010 -- TRUE

{a,b,c,d}=1010 => {f3,f2,f1,f0} = 0100 -- TRUE

{a,b,c,d}=1011 => {f3,f2,f1,f0} = 0110 -- TRUE

{a,b,c,d}=1100 => {f3,f2,f1,f0} = 0000 -- TRUE

{a,b,c,d}=1101 => {f3,f2,f1,f0} = 0011 -- TRUE

{a,b,c,d}=1110 => {f3,f2,f1,f0} = 0110 -- TRUE

{a,b,c,d}=1111 => {f3,f2,f1,f0} = 1001 -- TRUE

$finish called at time : 800 ns : File "C:/Users/jsphtkn/Vivado
Projects/sstu_experiment_3/sstu_experiment_3.srcs/sim_1/imports/Experiment_3/experiment3_tb.v" Line 52

INFO: [USF-XSim-96] XSim completed. Design snapshot 'experiment3_tb_behav' loaded.

INFO: [USF-XSim-97] XSim simulation ran for 1000ns

launch_simulation: Time (s): cpu = 00:00:01 ; elapsed = 00:00:05 . Memory (MB): peak = 3185.719 ; gain = 0.000
```

MUX Realization TCL Console (after behavioral sim)

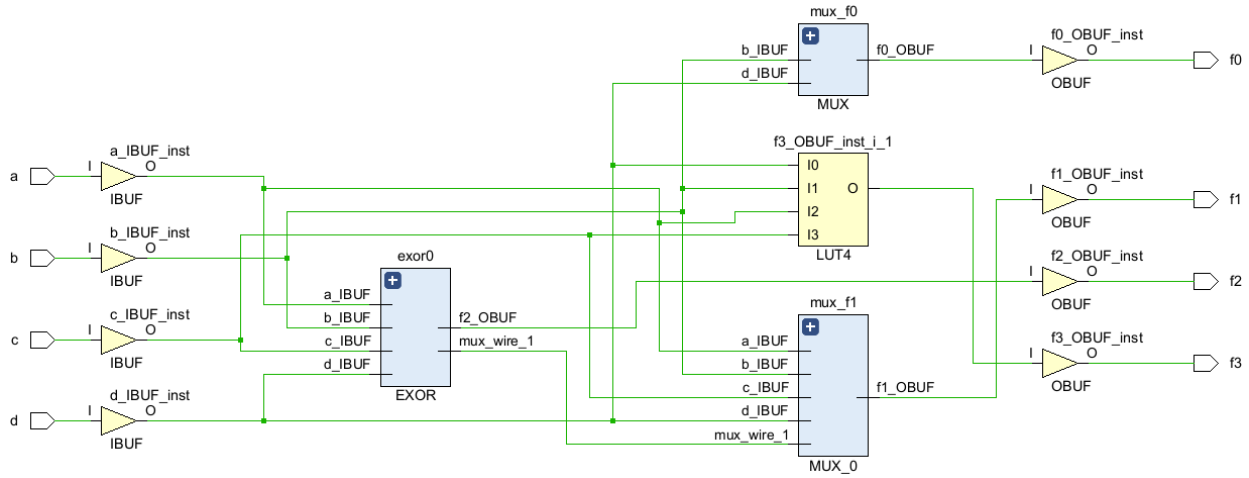


MUX Realization Behavioral Simulation

Yusuf Tekin
040200043



MUX Realization RTL Schematic



MUX Realization Technology Schematic

Yusuf Tekin
040200043

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
a	f1	8.454	SLOW	2.509	FAST
a	f2	8.502	SLOW	2.571	FAST
a	f3	8.356	SLOW	2.503	FAST
b	f0	8.813	SLOW	2.665	FAST
b	f1	8.899	SLOW	2.389	FAST
b	f2	9.313	SLOW	2.745	FAST
b	f3	8.620	SLOW	2.590	FAST
c	f1	8.636	SLOW	2.572	FAST
c	f2	8.828	SLOW	2.669	FAST
c	f3	8.845	SLOW	2.643	FAST
d	f0	9.442	SLOW	2.935	FAST
d	f1	9.713	SLOW	2.857	FAST
d	f2	10.127	SLOW	3.040	FAST
d	f3	9.256	SLOW	2.864	FAST

MUX Realization Combinational Delays

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	4	3	5	d	f2	10.127	5.250	4.877	∞	input port clock			0.000
Path 2	∞	4	3	5	d	f1	9.713	5.260	4.453	∞	input port clock			0.000
Path 3	∞	3	2	5	d	f0	9.442	5.117	4.325	∞	input port clock			0.000
Path 4	∞	3	2	5	d	f3	9.256	5.119	4.137	∞	input port clock			0.000

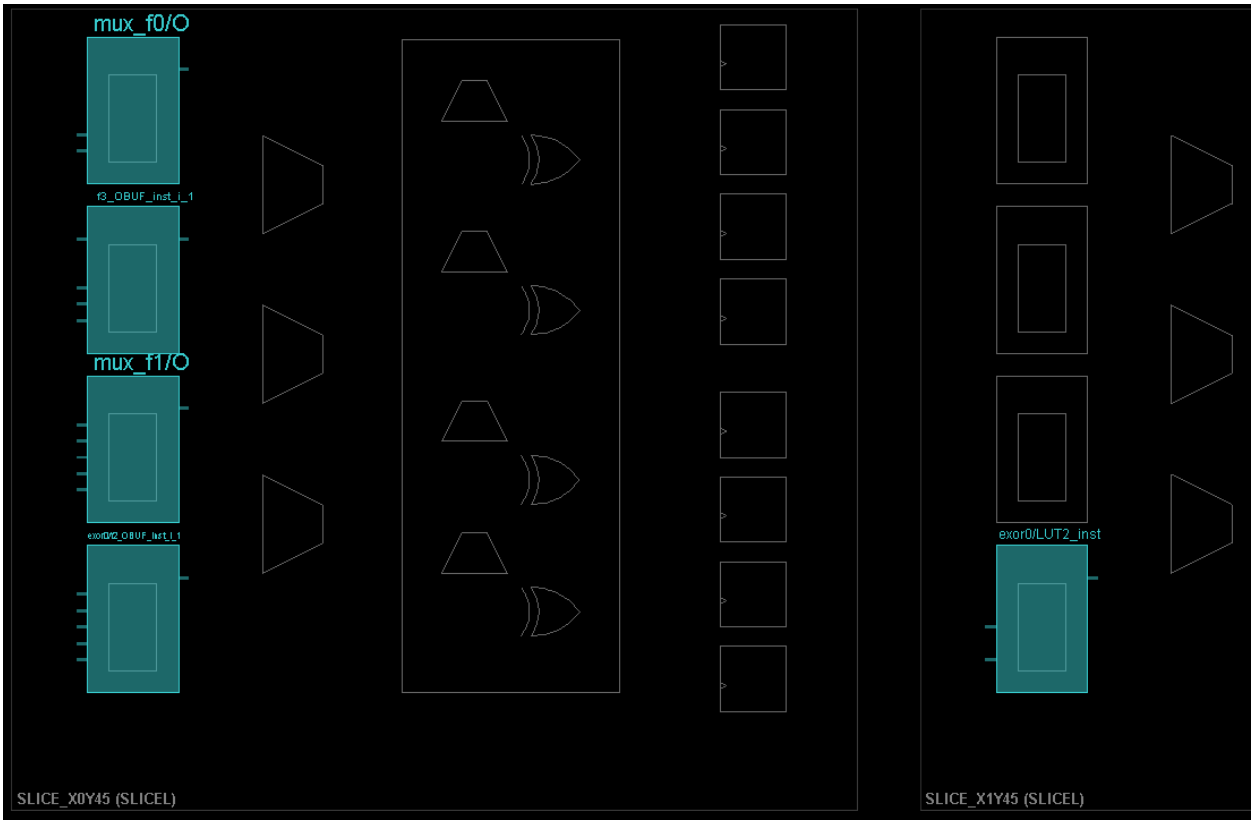
MUX Realization Path Delays – Setup

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	∞	3	2	5	b	f1	2.389	1.541	0.849	-∞	input port clock			0.000
Path 6	∞	3	2	3	a	f3	2.503	1.531	0.972	-∞	input port clock			0.000
Path 7	∞	3	2	3	a	f2	2.571	1.538	1.033	-∞	input port clock			0.000
Path 8	∞	3	2	5	b	f0	2.665	1.522	1.143	-∞	input port clock			0.000

MUX Realization Path Delays – Hold

Yusuf Tekin
040200043

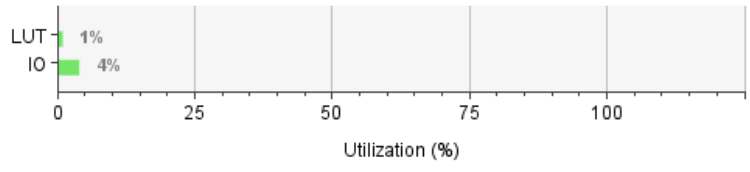


MUX Realization FPGA Device Overview

It is observable that in this design a LUT2 primitive and four LUT4 primitives are in use. There is not any LUT2 used for the decoder design nor SSI design.

Summary

Resource	Utilization	Available	Utilization %
LUT	5	32600	0.02
IO	8	210	3.81



MUX Realization Utilization Summary

Comparing the utilizations of both three designs, the MUX design is the least LUT used design of all, but the I/O usage is the same for both designs. Although, the design used the least

amount of LUTs, that does not mean it has the least delay. In terms of delay the ranking is actually backwards. Without any constraints applied, the MUX design has the greatest delay in the setup.

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
a	f1	8.174	SLOW
a	f2	8.327	SLOW
a	f3	8.130	SLOW
b	f0	8.271	SLOW
b	f1	8.395	SLOW
b	f2	8.567	SLOW
b	f3	8.082	SLOW
c	f1	8.270	SLOW
c	f2	8.782	SLOW
c	f3	7.908	SLOW
d	f0	8.820	SLOW
d	f1	8.741	SLOW
d	f2	8.835	SLOW
d	f3	7.929	SLOW

MUX Realization Combinational Delays (after 6ns max delay)

Other Path Groups - **default** - input port clock - - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	-2.835	3	2	5	d	f2	8.835	5.126	3.709	6.0	input port clock		MaxDelay Path 6.000ns	0.000
Path 2	-2.820	3	2	5	d	f0	8.820	5.117	3.703	6.0	input port clock		MaxDelay Path 6.000ns	0.000
Path 3	-2.741	4	3	5	d	f1	8.741	5.260	3.481	6.0	input port clock		MaxDelay Path 6.000ns	0.000
Path 4	-2.130	3	2	3	a	f3	8.130	5.141	2.989	6.0	input port clock		MaxDelay Path 6.000ns	0.000

MUX Realization Path Delays - Setup (after 6ns max delay)

Yusuf Tekin
040200043

Conclusion:

For the non-constrained designs, if the desired design requires less resources, the best choice would be the MUX design due to the 5 LUTs it uses yet it has the longest delays. Therefore, if the design is required to be precise in terms of delays and timing the best choice would be the decoder design.

Even though the rankings don't change, the delays and utilization change whenever a constraint is added. Most paths in the designs cannot meet the required max delay value but their delay drops remarkably. If the desire is the least delay the max-delay-constrained decoder design is slightly faster than the MUX design. Besides, all the time constrained designs have shorter delays.

4. Research

Simulation types and their differences:

Behavioral Simulation is the fastest and simplest simulation, but it only verifies the high-level logic. It's easy to code via Verilog or VHDL which makes it ideal for the early stages of design. There is no timing or netlist included.[1]

Post-Synthesis Functional Simulation confirms that the design logic is still correct after synthesis, using the synthesized netlist without timing information. This is useful to notice that synthesis hasn't changed the functionality.[1]

Post-Implementation Functional Simulation verifies that the placed-and-routed design (after physical layout on the FPGA) maintains the correct logical behavior. This is closer to the final design but still ignores timing.[1]

Post-Implementation Timing Simulation is the most accurate and realistic simulation, as it includes timing delays from the physical layout. This simulation checks both logic and timing, confirming that the design meets timing constraints before deploying it on the FPGA.[1]

FPGA Design Constraints:

Timing constraints, location constraints, I/O constraints, clock constraints and area constraints are the main types of constraints. By using these constraints, designers can use Vivado to produce a design that not only functions correctly but also meet the desired performance, power, and layout requirements.[2]

Yusuf Tekin
040200043

Synthesis Attributes:

In FPGA design there are special directives or properties that provide instructions to the synthesis tool on how to handle specific parts of the design during the synthesis process. They do not alter the functionality of the design but control how the design is implemented on the FPGA hardware.[3]

The **DONT_TOUCH** attribute is used to prevent the synthesis tool from optimizing or modifying a specific signal, register, or module. In Verilog or VHDL, you can apply the DONT_TOUCH attribute to a signal, register, or module instance as:

```
(* DONT_TOUCH = "true" *) reg [3:0] signal_name;
```

This way, this line of code is not optimized during the synthesis. Can lead to increased resource usage because the synthesis tool cannot optimize the protected elements.[3]

The **RAM_STYLE** attribute is used to control how the synthesis tool implements memory structures (like arrays or inferred RAM) in the FPGA. In FPGAs, memory can be implemented either using block RAM (BRAM), distributed RAM (using LUTs), or register-based memory. In Verilog or VHDL, you apply the RAM_STYLE attribute to arrays or inferred memory as:

```
(* ram_style = "block" *) reg [7:0] memory _name[0:15];
```

Choosing the wrong RAM style can lead to inefficient resource usage. For example, using block RAM for a very small memory structure wastes block RAM resources, while using distributed RAM for a large memory can consume too many LUTs. Using RAM_STYLE for such situations to avoid using unnecessary resources might result in a more efficient design.[3]

Yusuf Tekin
040200043

References:

1- Vivado Design Suite User Guide, Logic Simulation, Simulating with Vivado Simulator (UG900)
2- Vivado Design Suite User Guide, Using Constraints, UG903
3- Vivado Design Suite User Guide: Synthesis, Synthesis Attributes (UG901)