EHB436E

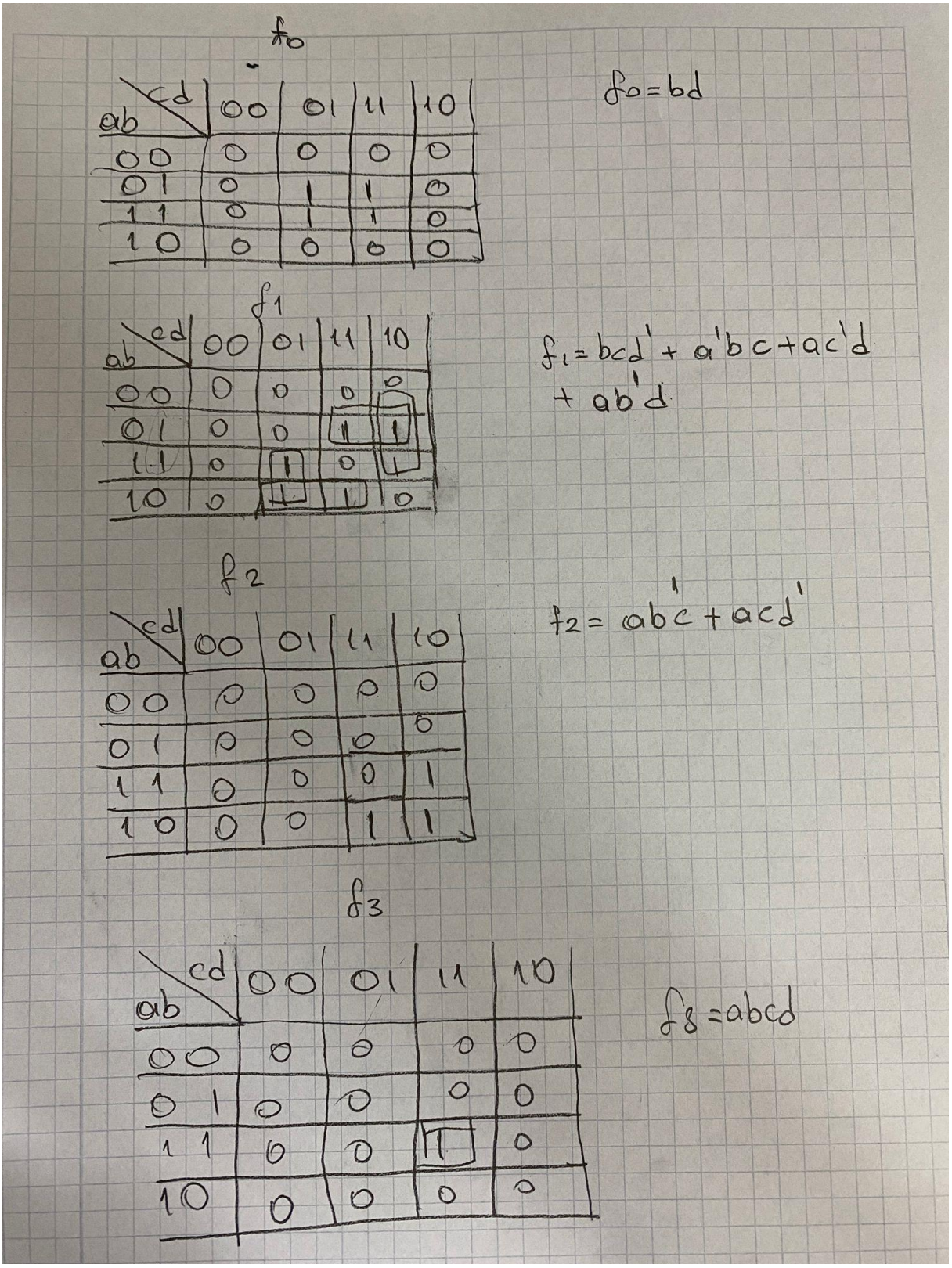
DIGITAL SYSTEM DESIGN APPLICATIONS

Muhammed Erkmen

040170049

EXPERIMENT-3 REPORT

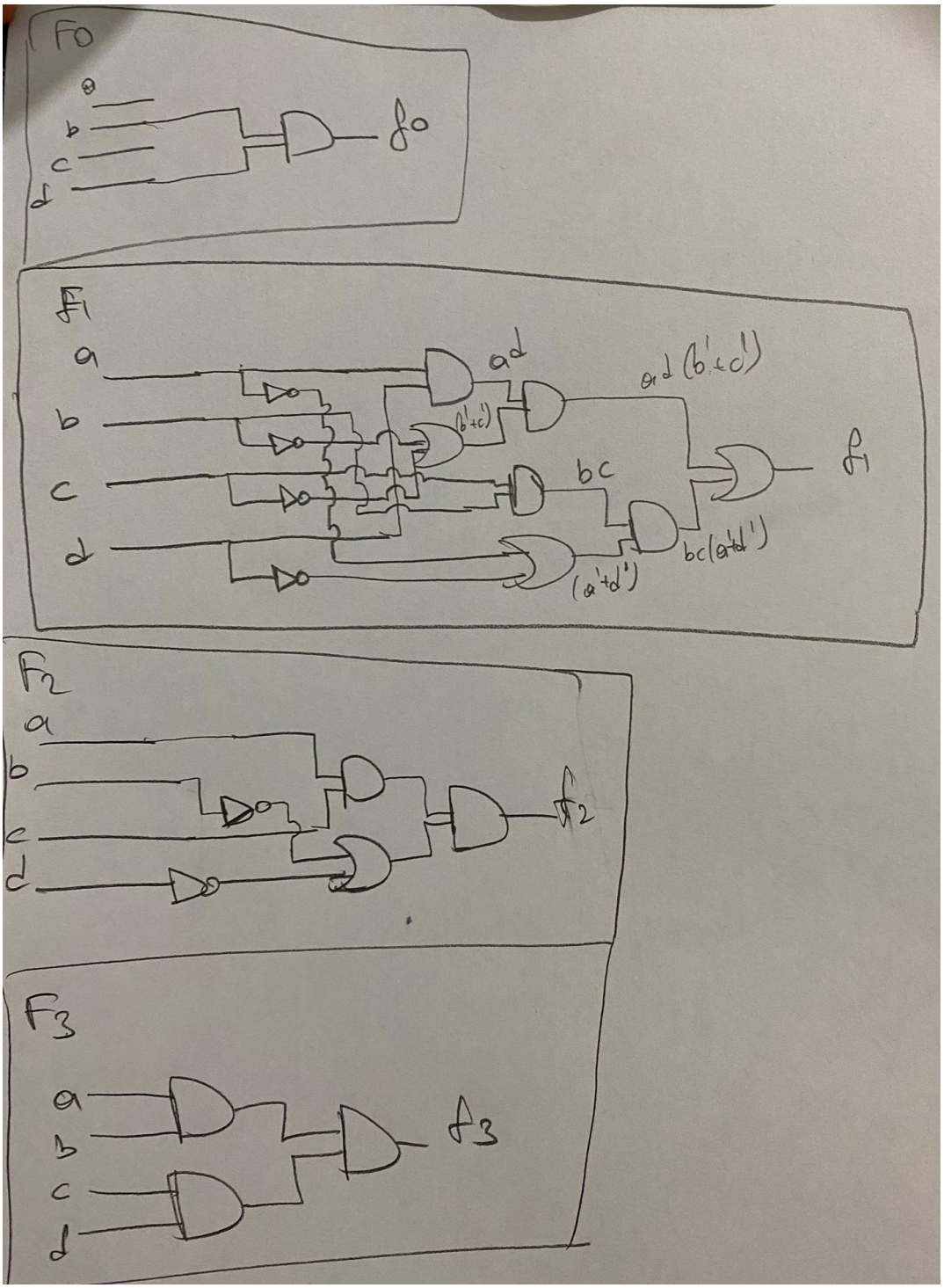
Question1



This is the karnaugh map solution of the functions

MULTI-LEVEL LOGIC

F0 = bd
F1 = bcd’ + a’bc + ac’d + ab’d = ad(c’+b’)+bc(a’+d’)
F2= ab’c+acd’ = ac(b’+d’)
F3 = (ab)(cd)



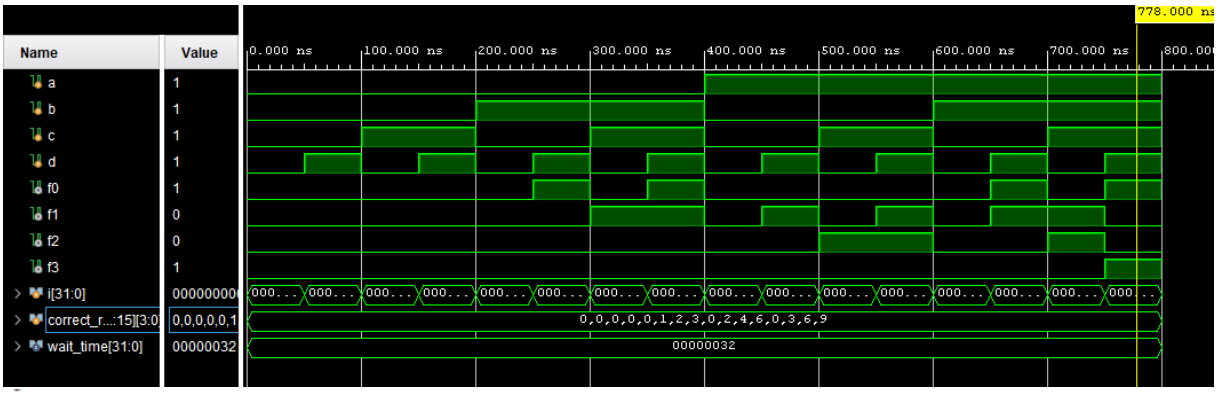
Source codes:

```
`timescale 1ns / 1ps
module behavioral(
input a,b,c,d,
output f0,f1,f2,f3
    );
wire a_not;
wire b_not;
wire c_not;
wire d_not;
// not calculations
not_gate not1 (.I(a),.O(a_not));
not_gate not2 (.I(b),.O(b_not));
not_gate not3(.I(c),.O(c_not));
not_gate not4 (.I(d),.O(d_not));
// f0 calculation
and_gate and1 (.I1(b),.I2(d),.O(f0));
// f1 calculation
wire element1_1;
wire element1_2;
wire element1;
wire element2_1;
wire element2_2;
wire element2;
and_gate and2 (.I1(a),.I2(d),.O(element1_1));                  // ad
or_gate or1 (.I1(c_not),.I2(b_not),.O(element1_2));            // b'+c'
and_gate and3 (.I1(element1_1),.I2(element1_2),.O(element1)); // ELEMENT1
and_gate and4 (.I1(b),.I2(c),.O(element2_1));
or_gate or2 (.I1(a_not),.I2(d_not),.O(element2_2));
and_gate and5 (.I1(element2_1),.I2(element2_2),.O(element2));
or_gate or3(.I1(element1),.I2(element2),.O(f1));

//f2 calculation
wire ele1;
wire ele2;
and_gate and6(.I1(a),.I2(c),.O(ele1));
or_gate or4 (.I1(b_not),.I2(d_not),.O(ele2));
and_gate and7 (.I1(ele1),.I2(ele2),.O(f2));

//f3 calculation
wire el1;
wire el2;

and_gate and8 (.I1(a),.I2(b),.O(el1));
and_gate and9 (.I1(c),.I2(d),.O(el2));
and_gate and10 (.I1(el1),.I2(el2),.O(f3));

endmodule
```

Testbench codes:

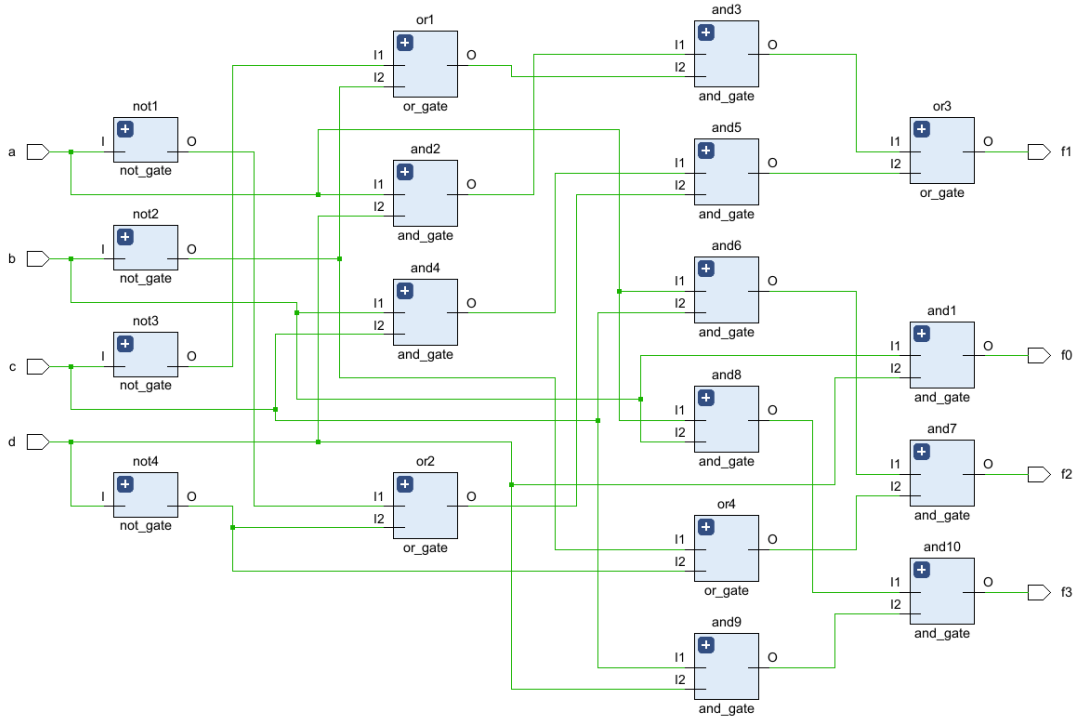
```
`timescale 1ns / 1ps

module experiment3_tb();
    // ----- Inputs & Outputs
    reg a,b,c,d;
    wire f0,f1,f2,f3;
    // ----- Testbench Parameters ----- //
    parameter wait_time = 50;
    integer i;

    reg [3:0]correct_results[0:15];
    initial begin
        correct_results[0]  = 0; correct_results[1]  = 0;
correct_results[2]  = 0 ;correct_results[3] = 0;
        correct_results[4]  = 0; correct_results[5]  = 1;
correct_results[6]  = 2 ;correct_results[7] = 3;
        correct_results[8]  = 0; correct_results[9]  = 2;
correct_results[10] = 4 ;correct_results[11] = 6;
        correct_results[12] = 0; correct_results[13] = 3;
correct_results[14] = 6 ;correct_results[15] = 9;
    end
    // ---------------------------------- //
    // ---------- UUT Instantiation -------------- //
    behavioral
    //with_decoder
    //with_MUX
    UUT (.a(a),
        .b(b),
        .c(c),
        .d(d),
        .f0(f0),
        .f1(f1),
        .f2(f2),
        .f3(f3));
    // ------- Test Procedure ------- //
    initial
    begin
        for(i=0;i<16;i=i+1)
        begin
            {a,b,c,d} = i;
            #(wait_time);
            $write("{a,b,c,d}=%d%d%d%d => {f3,f2,f1,f0} = %d%d%d%d --
",a,b,c,d,f3,f2,f1,f0);
            if({f3,f2,f1,f0} == correct_results[i])
                $display("TRUE");
            else
                $display("FALSE");
        end
        $finish();
    end
    // ------------------------------- //

endmodule
```

Simulation Results:

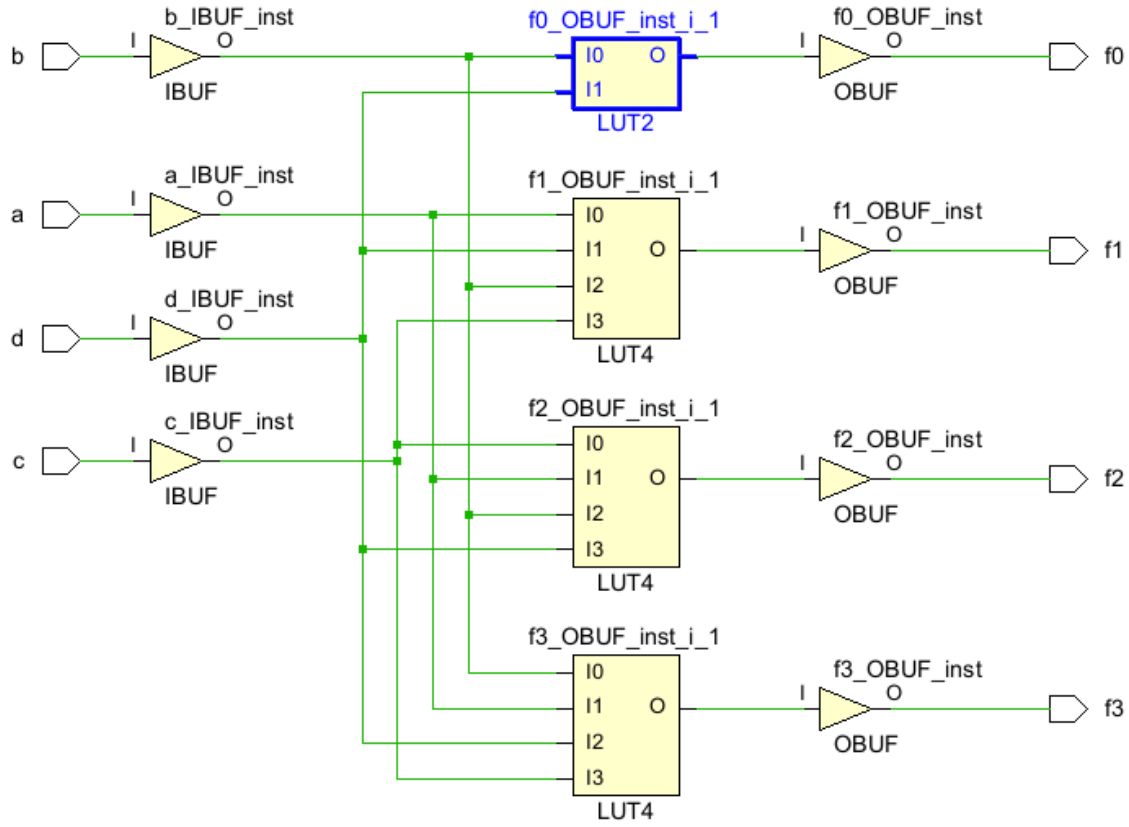



```
{a,b,c,d}=0000 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0001 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0010 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0011 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0100 => {f3,f2,f1,f0} = 0000 -- TRUE
{a,b,c,d}=0101 => {f3,f2,f1,f0} = 0001 -- TRUE
{a,b,c,d}=0110 => {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/Progg/
INFO: [USF-XSim-96] XSim completed. Design snapshot 'ex
```

RTL Schematic:



Technology schematic



Pad to pad delays:

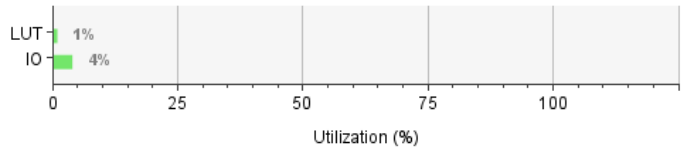
Combinational Delays

From Port	To Port	Ma	Max Process Corner	Min Delay	Min Process Corner
d	f0	9.437	SLOW	2.894	FAST
b	f0	9.416	SLOW	2.850	FAST
a	f2	9.263	SLOW	2.871	FAST
a	f1	9.131	SLOW	2.782	FAST
d	f2	9.027	SLOW	2.753	FAST
b	f2	9.016	SLOW	2.721	FAST
d	f1	8.897	SLOW	2.666	FAST
b	f1	8.886	SLOW	2.634	FAST
a	f3	8.867	SLOW	2.696	FAST
d	f3	8.641	SLOW	2.592	FAST
b	f3	8.620	SLOW	2.546	FAST
c	f2	8.292	SLOW	2.480	FAST
c	f1	8.128	SLOW	2.397	FAST
c	f3	7.908	SLOW	2.316	FAST

Utilization report:

Summary

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
IO	8	210	3.81



B5 LUT and A5 LUT are used in this application without constraint time codes.

After timing constraints

```
set_max_delay -from [get_ports {a b c d}] -to [get_ports {f0 f1 f2 f3}] 9.0
```

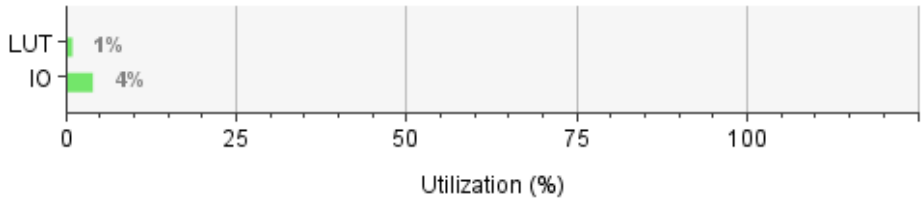
Pad to pad delays:

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
a	f2	8.914	SLOW
b	f2	8.805	SLOW
c	f2	8.643	SLOW
a	f1	8.592	SLOW
b	f0	8.500	SLOW
d	f1	8.337	SLOW
b	f1	8.249	SLOW
d	f2	8.220	SLOW
d	f0	8.220	SLOW
d	f3	8.121	SLOW
c	f1	8.070	SLOW
c	f3	8.068	SLOW
b	f3	7.947	SLOW
a	f3	7.934	SLOW

Utilization report after timing constraint:

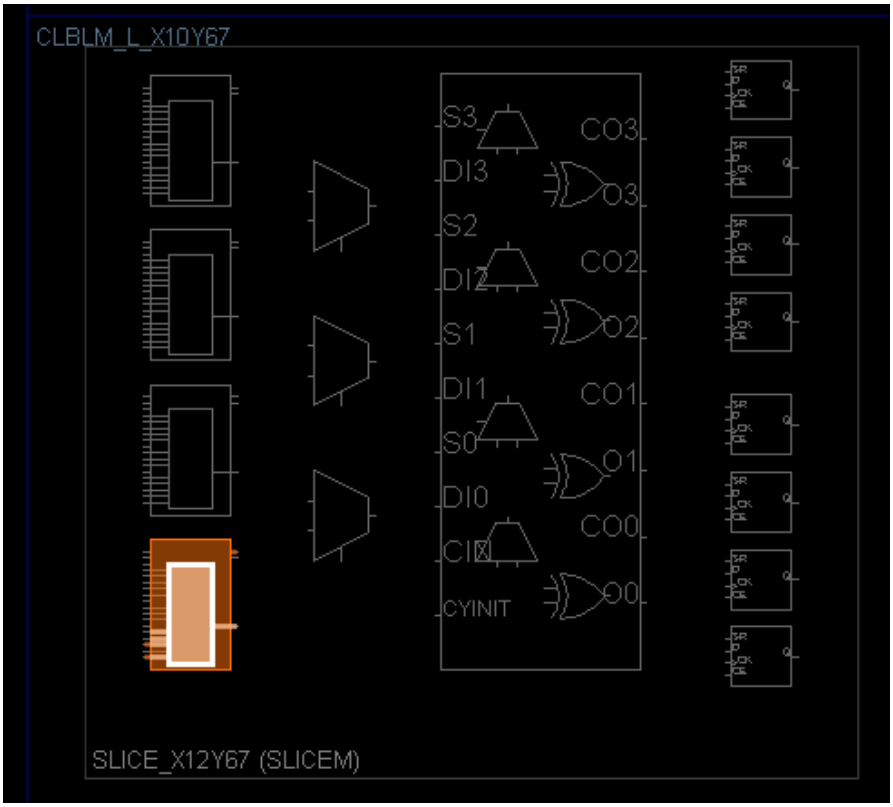
Resource	Utilization	Available	Utilization %
LUT	4	32600	0.01
IO	8	210	3.81



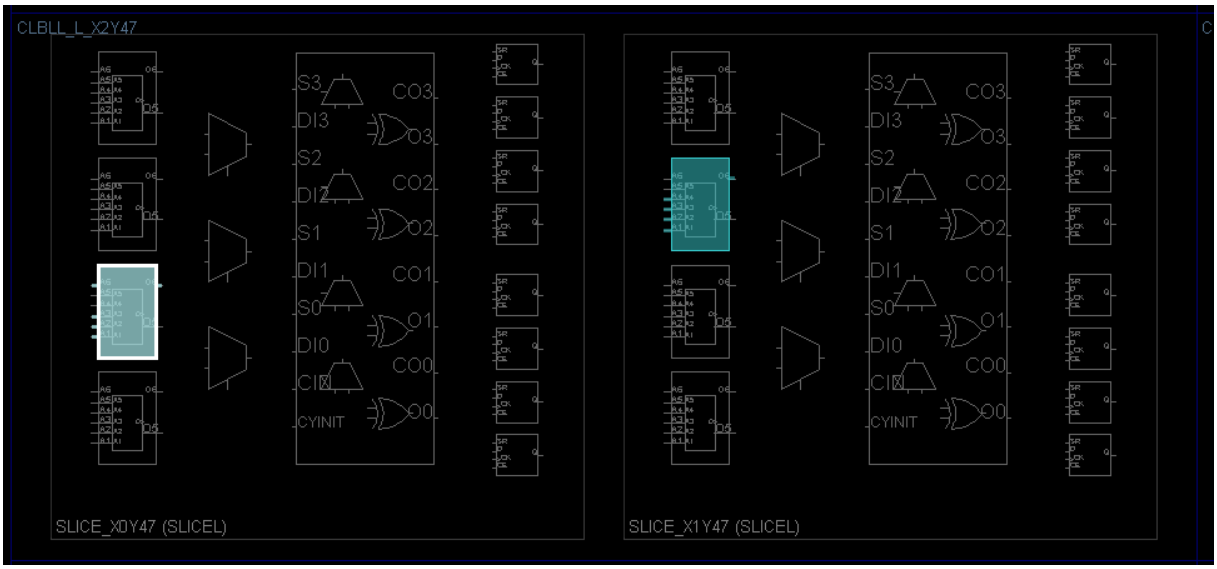
So there are 4 LUTs instead of 2. Places of LUTs changed.

Logic cell places before LOC constraints:

F3:



F2 and F1:

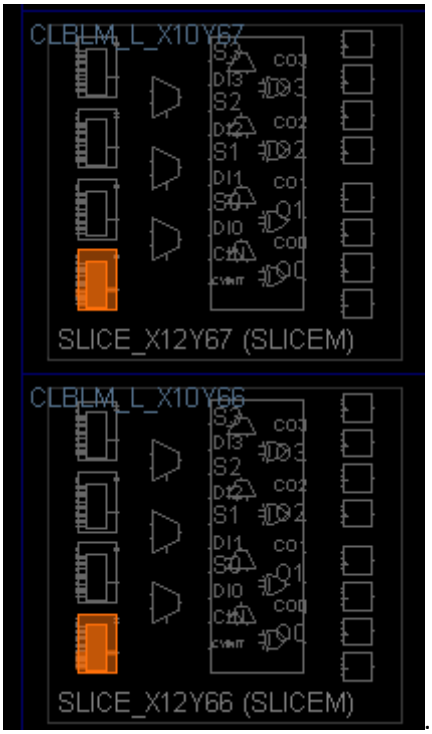


and f0 was in compeletly different clock region.

After LOC Constraints:

```
set_property LOC SLICE_X12Y67 [get_cells f3_OBUF_inst_i_1]
set_property LOC SLICE_X12Y66 [get_cells f2_OBUF_inst_i_1]
set_property LOC SLICE_X13Y67 [get_cells f1_OBUF_inst_i_1]
set_property LOC SLICE_X14Y64 [get_cells f0_OBUF_inst_i_1]
```

Vivado implemented f0 and f3 to SLICE_X12Y67, f1 and f2 to SLICE_X12Y66.

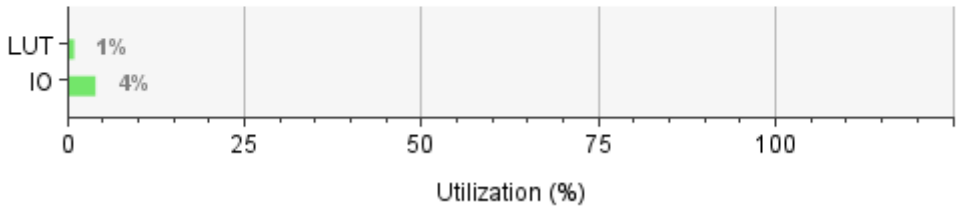


Pad to pad delays:

From Port	To Port	M a	Max Process Corner	Min Delay	Min Process Corner
a	f3	11.454	SLOW	3.849	FAST
c	f3	11.015	SLOW	3.670	FAST
b	f3	10.778	SLOW	3.617	FAST
d	f3	10.662	SLOW	3.520	FAST
a	f1	10.603	SLOW	3.515	FAST
a	f2	10.568	SLOW	3.454	FAST
b	f1	10.339	SLOW	3.414	FAST
b	f2	10.300	SLOW	3.353	FAST
d	f1	9.682	SLOW	3.143	FAST
d	f2	9.645	SLOW	3.081	FAST
c	f1	9.630	SLOW	3.150	FAST
b	f0	9.565	SLOW	3.086	FAST
c	f2	9.561	SLOW	3.093	FAST
d	f0	9.416	SLOW	2.991	FAST

Utilization report:

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
IO	8	210	3.81



UNCOMMENT TIMING:

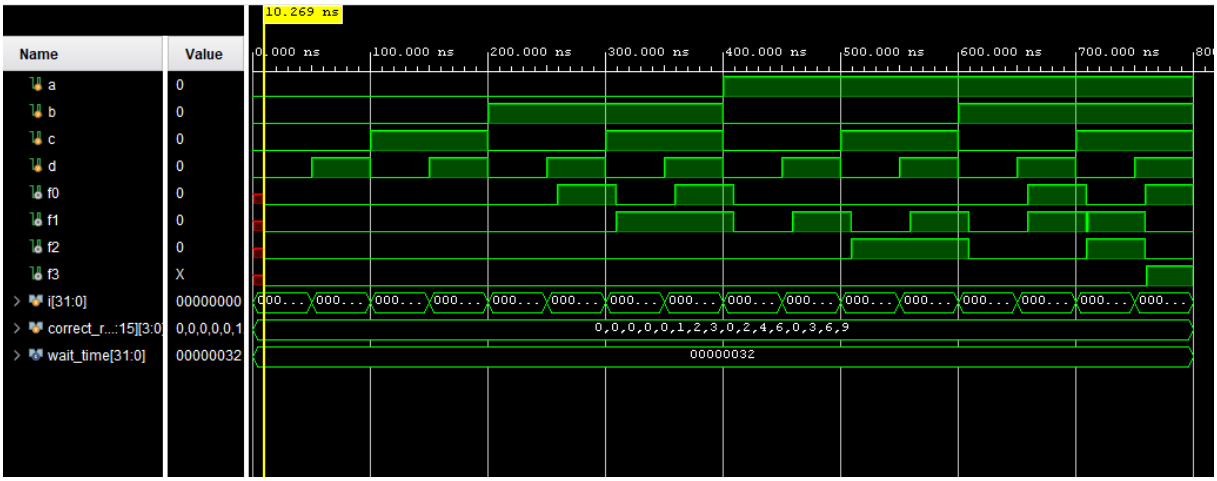
Pad to pad delays:

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
b	f3	10.752	SLOW
a	f3	10.580	SLOW
c	f3	10.476	SLOW
d	f3	10.298	SLOW
a	f1	10.060	SLOW
a	f2	9.991	SLOW
d	f1	9.889	SLOW
b	f1	9.836	SLOW
d	f2	9.821	SLOW
b	f2	9.765	SLOW
b	f0	9.747	SLOW
c	f1	9.634	SLOW
c	f2	9.531	SLOW
d	f0	9.294	SLOW

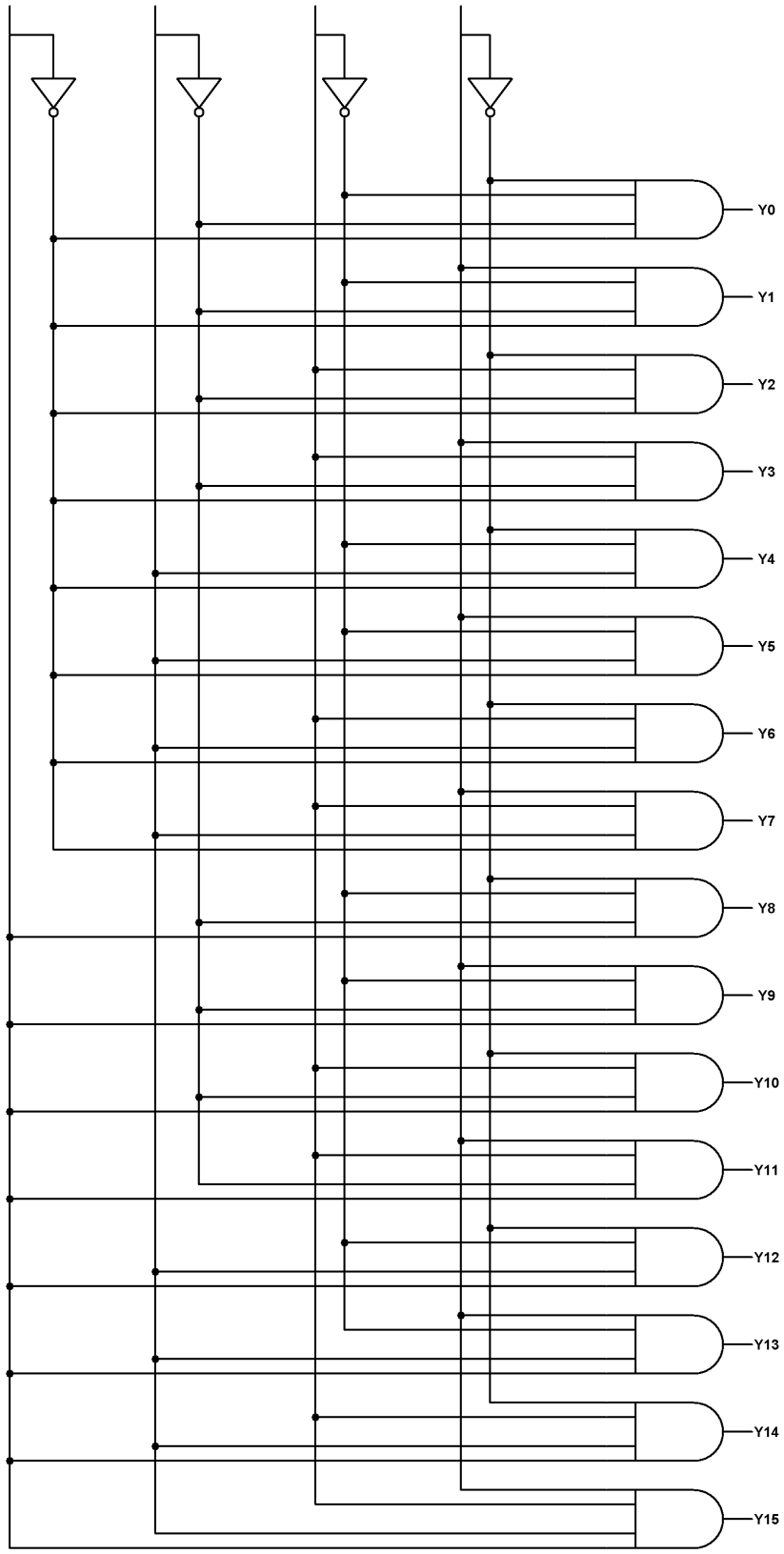
RESULTS:
To get best timing, only timing constraint codes implementation is the best one.

Post-implementation simulation results:

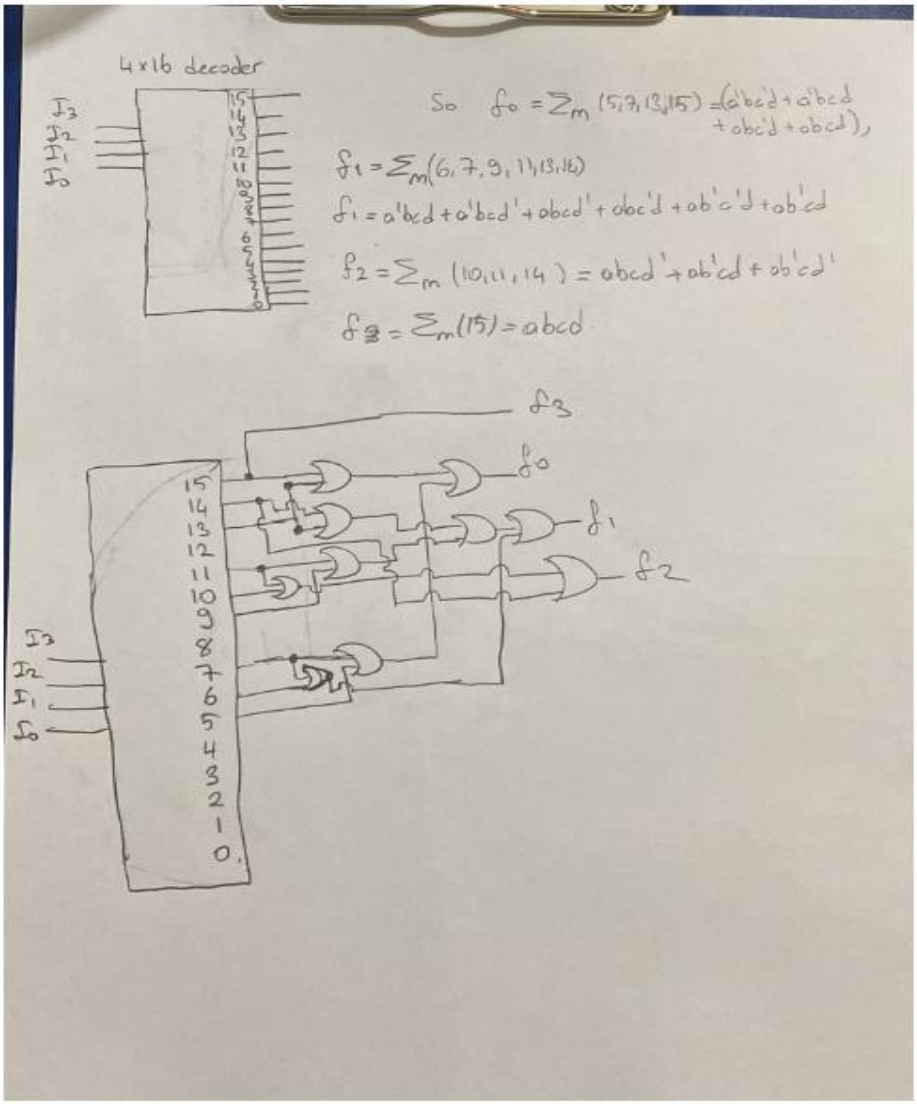


2) REALIZATION WITH DECODER

4x16 decoder:



Kaynak: https://www.elprocus.com/designing-4-to-16-decoder-using-3-to-8-decoder/

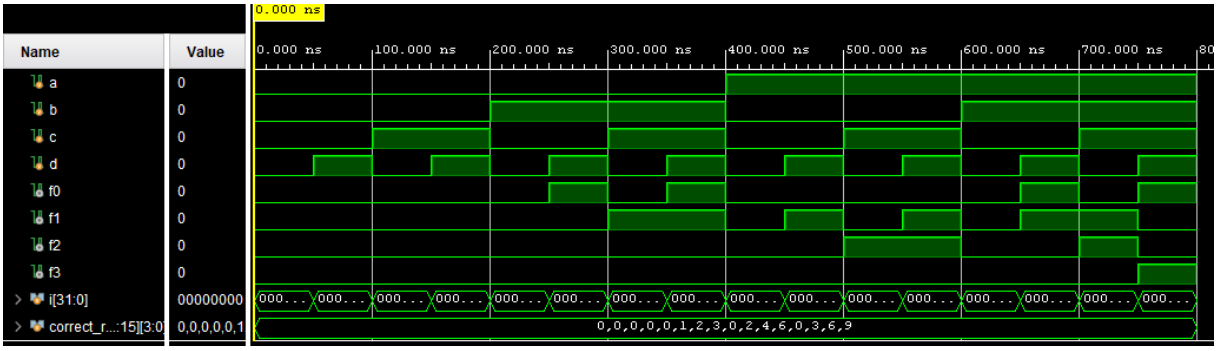


Source codes:

```
`timescale 1ns / 1ps

module with_decoder(
input a,b,c,d,
output f3,f2,f1,f0
    );

wire [15:0] minterms;
wire [3:0] inputs;



DECODER DLS (.IN({a,b,c,d}),.OUT(minterms));

wire f0_element_1;
wire f0_element_2;
//f0
or_gate or0_1 (.I1(minterms[5]),.I2(minterms[7]),.O(f0_element_1));
or_gate or0_2 (.I1(minterms[13]),.I2(minterms[15]),.O(f0_element_2));
or_gate or0_3 (.I1(f0_element_2),.I2(f0_element_1),.O(f0));

//f1
```

```
wire f1_element_1,f1_element_2,f1_element_3,f1_element_4;
or_gate or1_1 (.I1(minterms[6]),.I2(minterms[7]),.O(f1_element_1));
or_gate or1_2 (.I1(minterms[9]),.I2(minterms[11]),.O(f1_element_2));
or_gate or1_3 (.I1(minterms[13]),.I2(minterms[14]),.O(f1_element_3));
or_gate or1_4 (.I1(f1_element_2),.I2(f1_element_1),.O(f1_element_4));
or_gate or1_5 (.I1(f1_element_4),.I2(f1_element_3),.O(f1));

//f2
wire f2_element_1;
or_gate or2_1 (.I1(minterms[10]),.I2(minterms[11]),.O(f2_element_1));
or_gate or2_2 (.I1(minterms[14]),.I2(f2_element_1),.O(f2));

//f3

assign f3 = minterms[15];

endmodule
```

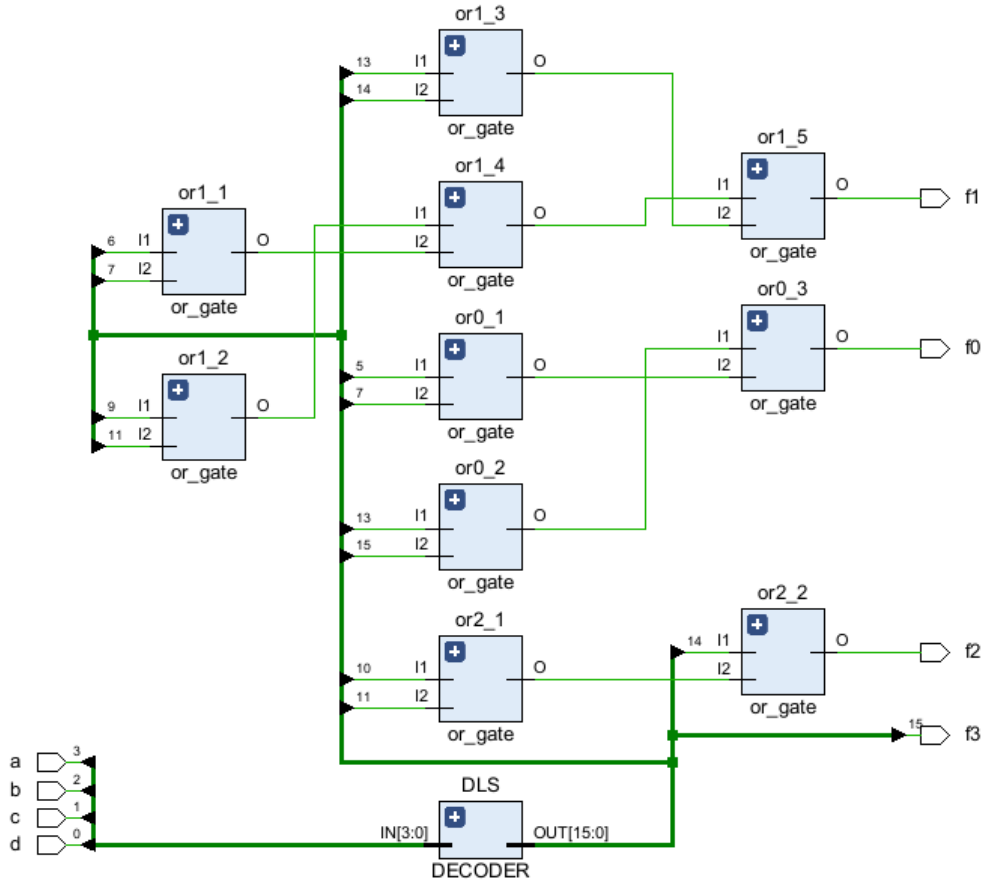
Behavioral schematic:



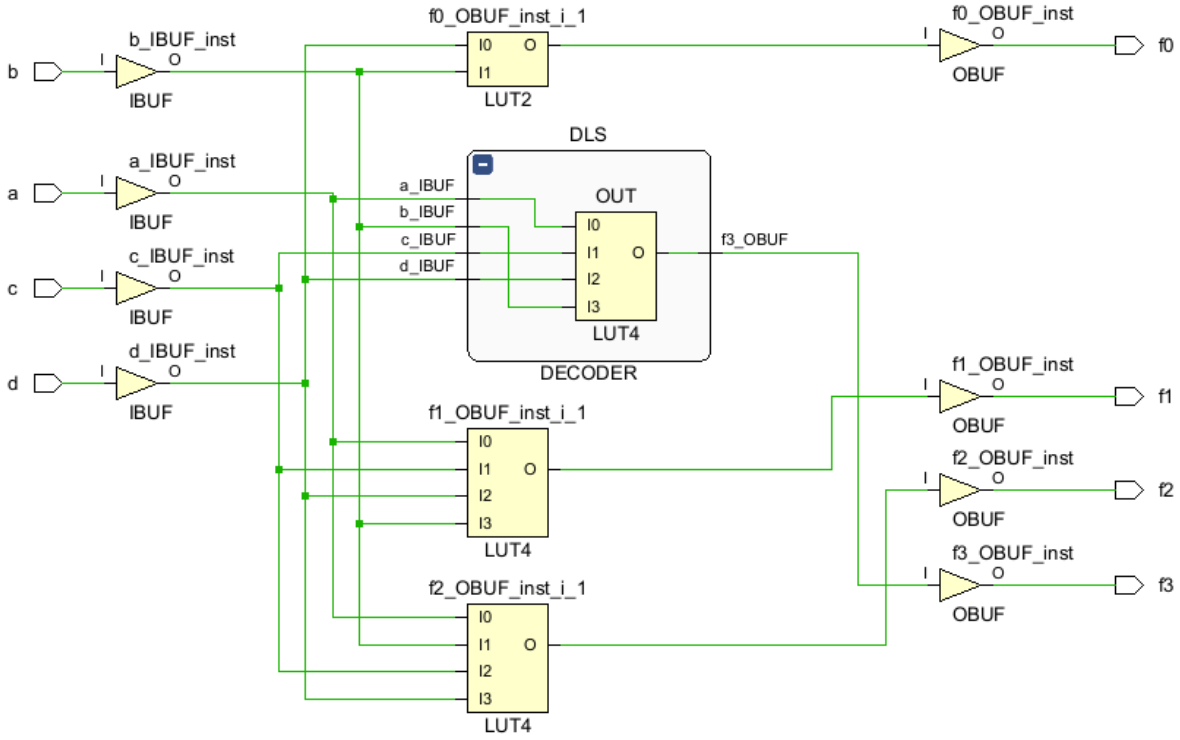
Console output:

```
# run 1000ns
{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
```

RTL Schematic:



Tech schematic:



Pad to Pad delays without constraint codes:

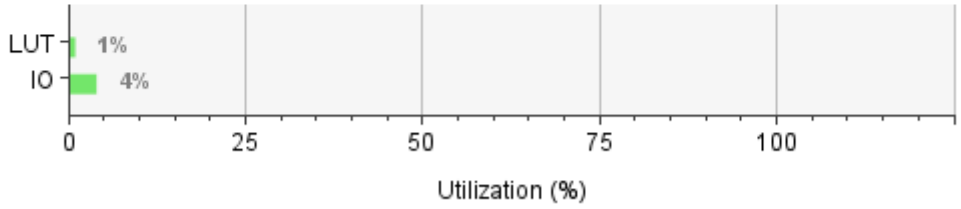
Combinational Delays

From Port	To Port	Ma	Max Process Corner	Min Delay	Min Process Corner
a	f2	9.468	SLOW	2.951	FAST
a	f1	9.421	SLOW	2.894	FAST
a	f3	9.068	SLOW	2.769	FAST
b	f0	8.875	SLOW	2.670	FAST
b	f2	8.863	SLOW	2.696	FAST
b	f1	8.814	SLOW	2.638	FAST
b	f3	8.737	SLOW	2.604	FAST
d	f2	8.658	SLOW	2.643	FAST
c	f2	8.604	SLOW	2.576	FAST
d	f1	8.577	SLOW	2.590	FAST
c	f3	8.576	SLOW	2.531	FAST
c	f1	8.555	SLOW	2.521	FAST
d	f3	8.420	SLOW	2.522	FAST
d	f0	8.073	SLOW	2.392	FAST

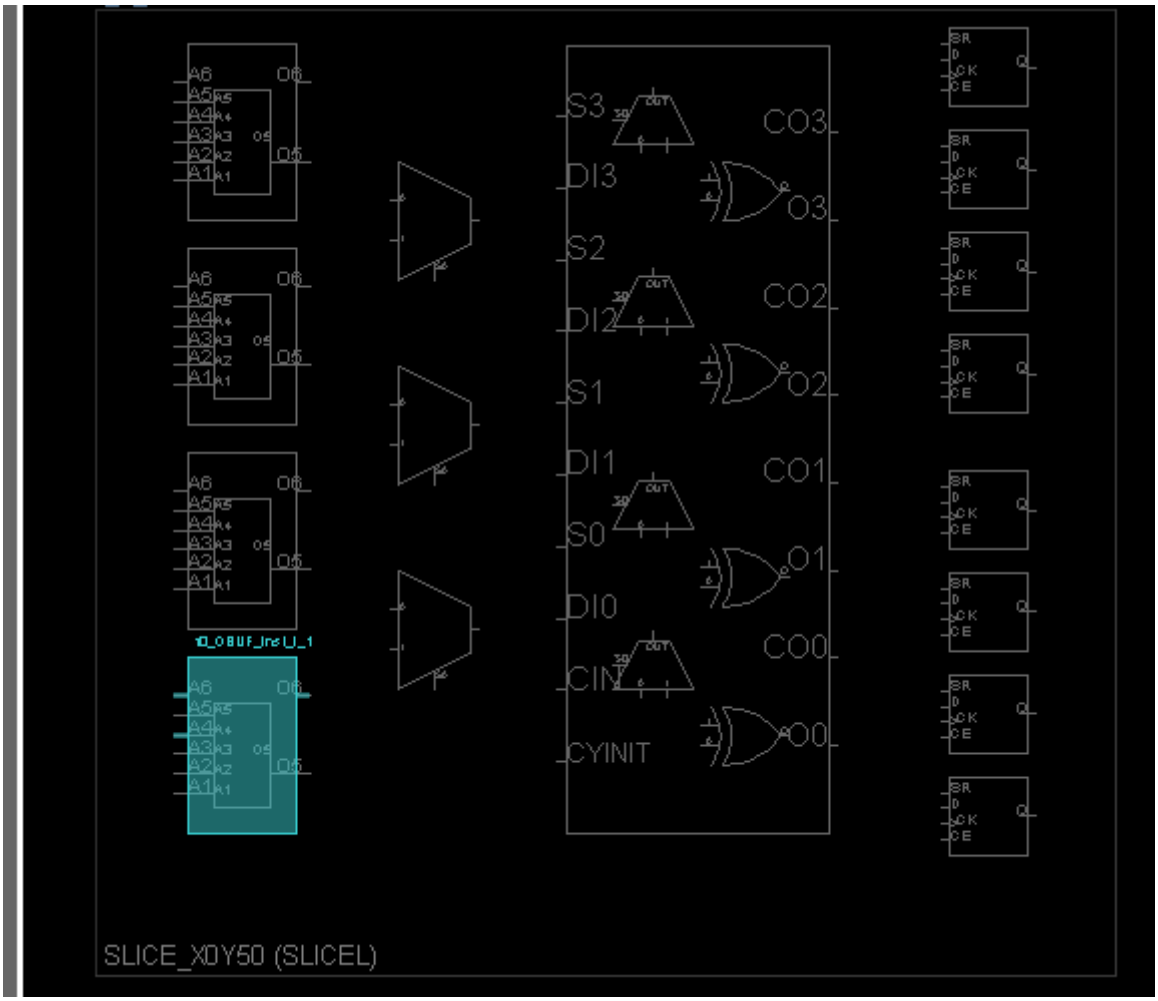
Utilization summary:

Summary

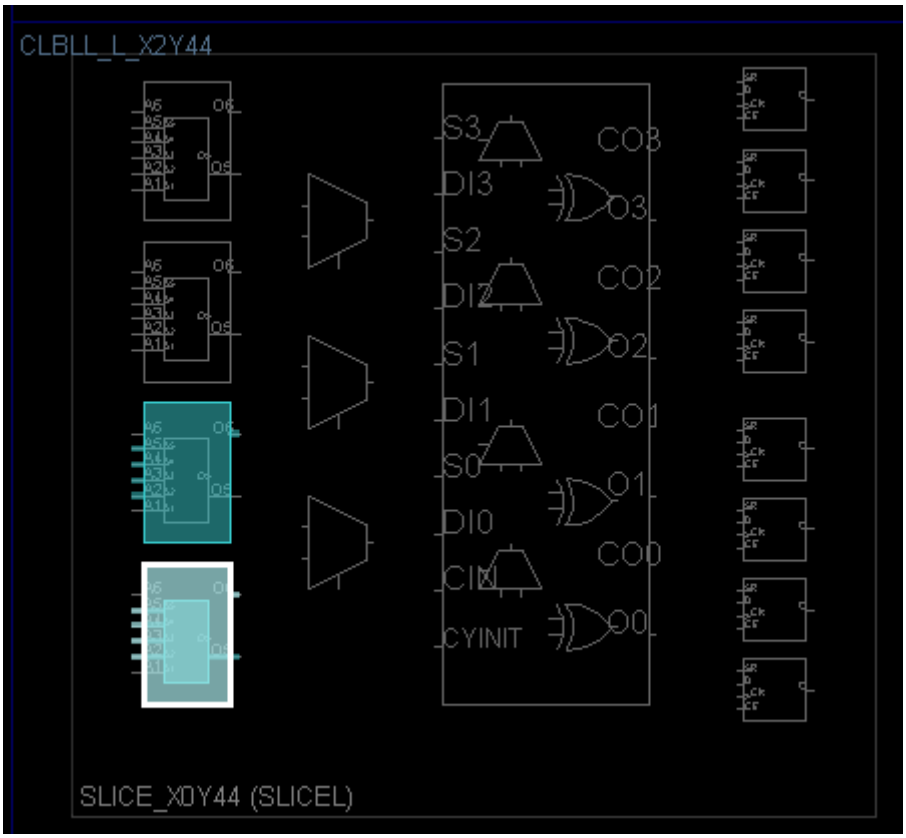
Resource	Utilization	Available	Utilization %
LUT	3	32600	0.01
IO	8	210	3.81



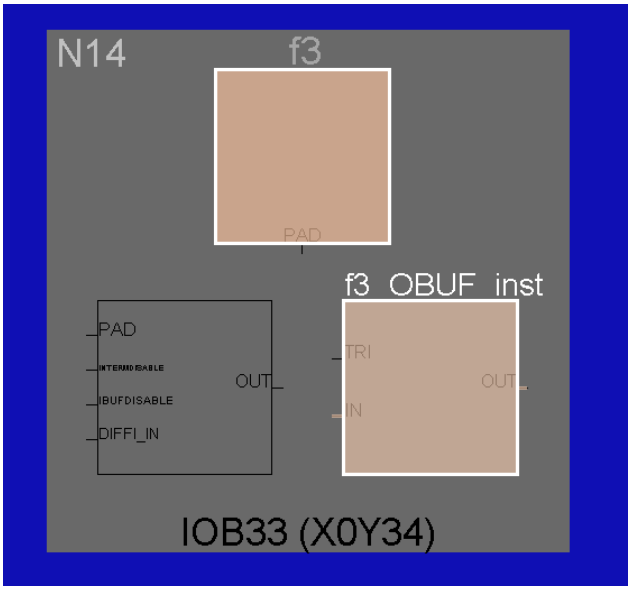
F0 PLACE:



F2 and F1 PLACE:



F3 PLACE:



As we see, there is no LUT defined for f3 and only IOB is defined for it.

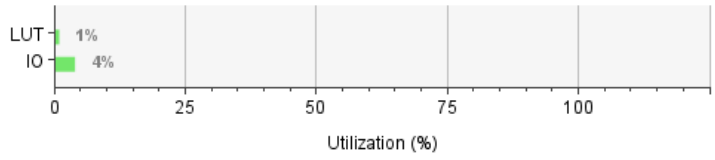
DECODER WITH CONSTRAINT TIMING AND LOCS (9ns)

Pad to pad delays:

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
d	f0	10.442	SLOW
b	f0	10.194	SLOW
b	f1	10.119	SLOW
b	f2	10.050	SLOW
d	f1	9.944	SLOW
a	f1	9.903	SLOW
c	f1	9.880	SLOW
d	f2	9.871	SLOW
c	f2	9.809	SLOW
a	f2	9.800	SLOW
a	f3	8.474	SLOW
d	f3	8.315	SLOW
c	f3	8.269	SLOW
b	f3	8.162	SLOW

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

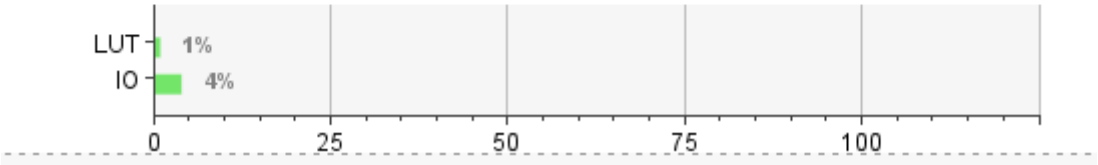


DECODER WITH TIMING CONSTRAINTS (NO LOC)

From Port	To Port	Max Delay	Max Process Corner
b	f2	8.803	SLOW
a	f2	8.786	SLOW
d	f2	8.733	SLOW
a	f1	8.589	SLOW
d	f0	8.490	SLOW
b	f0	8.429	SLOW
d	f1	8.324	SLOW
c	f2	8.322	SLOW
a	f3	8.241	SLOW
c	f1	8.224	SLOW
c	f3	8.168	SLOW
b	f1	8.103	SLOW
d	f3	8.041	SLOW
b	f3	7.702	SLOW

Summary

Resource	Utilization	Available	Utilization %
LUT	4	32600	0.01
IO	8	210	3.81

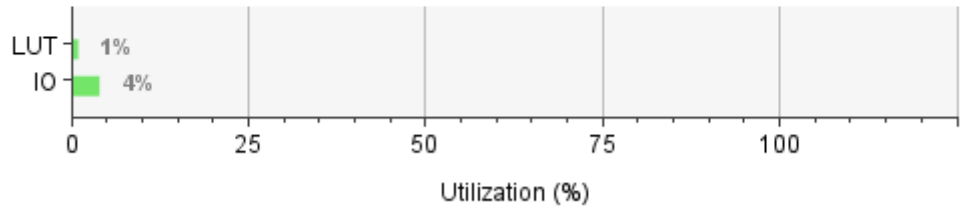


REPORTS WITH TIMING CONSTRAINTS (6 ns, NO LOC)

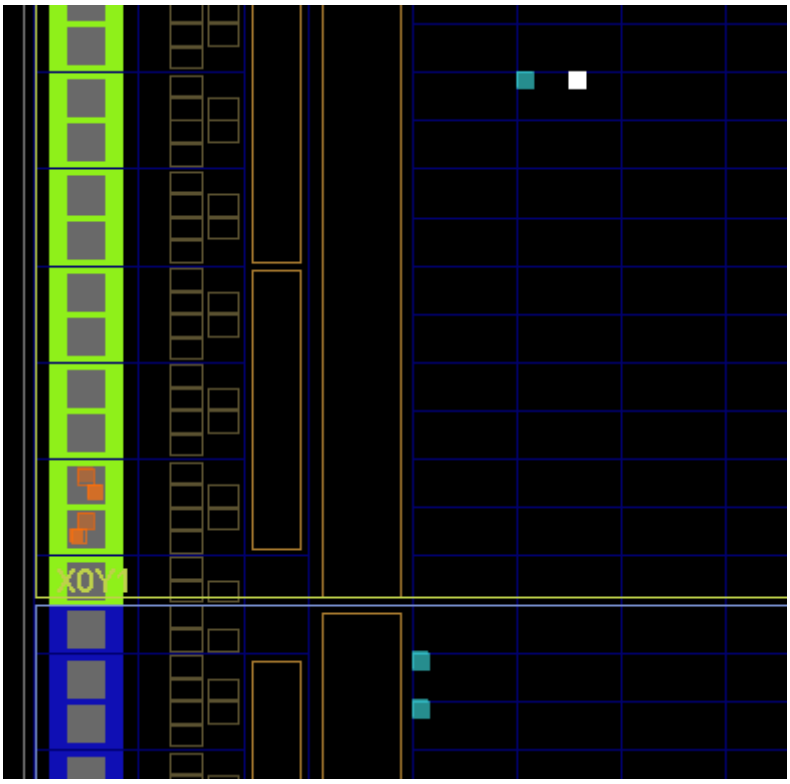
Combinational Delays

From Port	To Port	Max Delay [1]	Max Process Corner
a	f2	9.042	SLOW
d	f2	8.730	SLOW
b	f2	8.489	SLOW
b	f1	8.463	SLOW
c	f2	8.437	SLOW
a	f1	8.410	SLOW
c	f1	8.331	SLOW
d	f1	8.291	SLOW
b	f0	8.262	SLOW
a	f3	8.144	SLOW
d	f3	8.126	SLOW
c	f3	8.068	SLOW
d	f0	8.067	SLOW
b	f3	7.953	SLOW

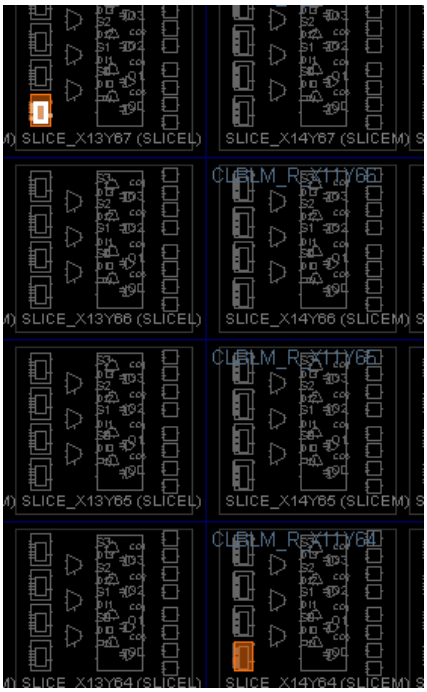
Resource	Utilization	Available	Utilization %
LUT	4	32600	0.01
IO	8	210	3.81



Locations of the LUT cells with 6ns and no LOC constraint.



REPORTS WITH TIMING&LOC CONSTRAINTS (6 ns)

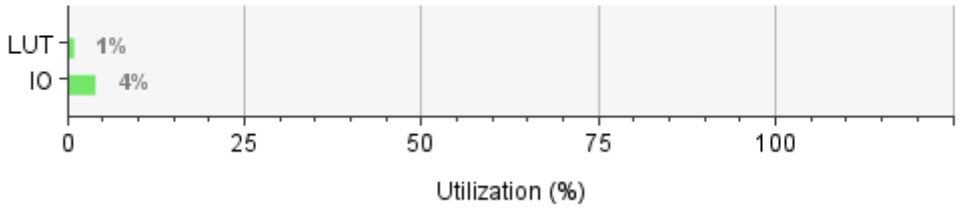


Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
b	f1	9.878	SLOW
a	f1	9.866	SLOW
c	f1	9.827	SLOW
b	f2	9.793	SLOW
a	f2	9.749	SLOW
c	f2	9.744	SLOW
d	f1	9.556	SLOW
d	f2	9.471	SLOW
b	f0	9.239	SLOW
d	f0	8.979	SLOW
d	f3	8.155	SLOW
c	f3	8.068	SLOW
b	f3	7.936	SLOW
a	f3	7.897	SLOW

Utizilation report of 6 ns and loc constraints structure:

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



3)MULTIPLEXER

F0 = bd
F1 = bcd’ + a’bc + ac’d + ab’d = ad(c’+b’)+bc(a’+d’)
F2= ab’c+acd’ = ac(b’+d’)
F3 = (ab)(cd)

If we choose a and c select, so in f0, everycase is goes to bd.

F1= a’bcd + a’bcd’ + abcd’ + abc’d + ab’c’d+ ab’cd

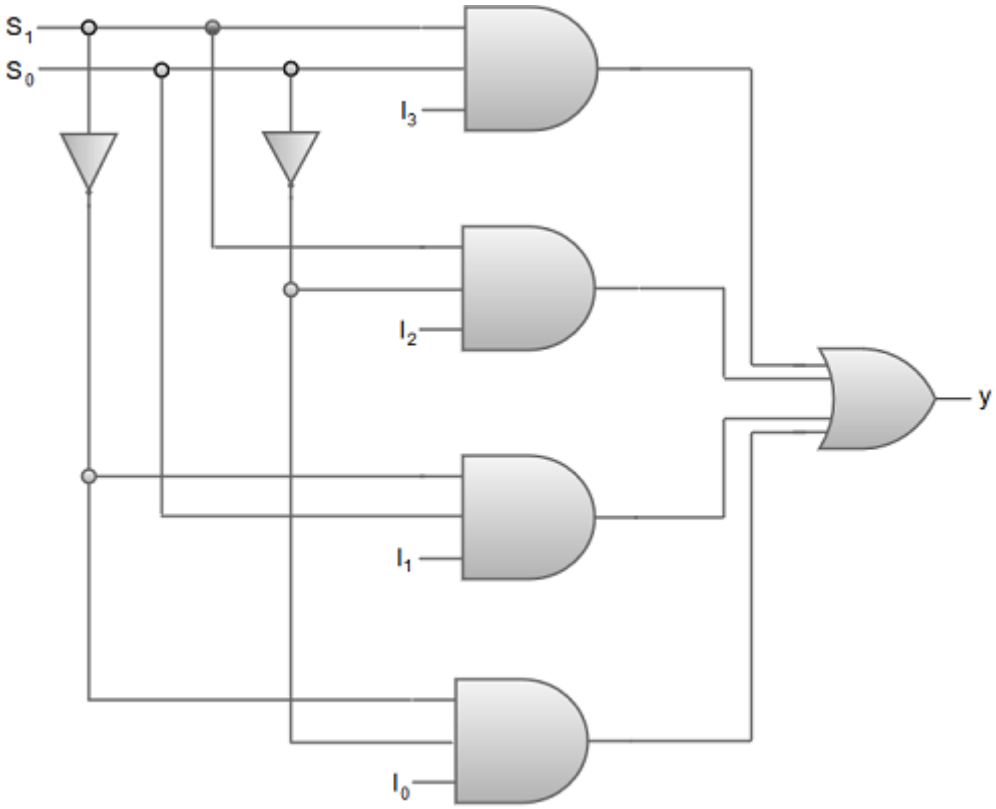
So f1 => a’c(b) + ac(bd’+b’d) + ac’(d) + a’c’(0) when ac is selection.

F2 = abcd’+ab’cd+ab’cd’

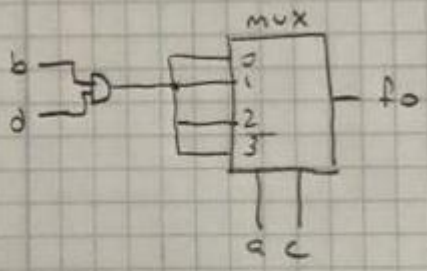
So f2=> ac(bd’+b’d+b’d’) = ac(b’(d+d’)+bd’) = ac(b’+bd’)

F3 = ac(bd) else 0

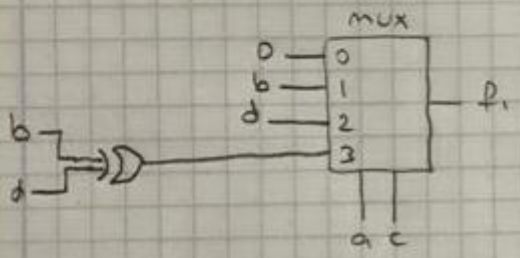
4 to 1 (2 select) MUX :



$f_0 = a'bc'd + a'bcd + abc'd + abcd$

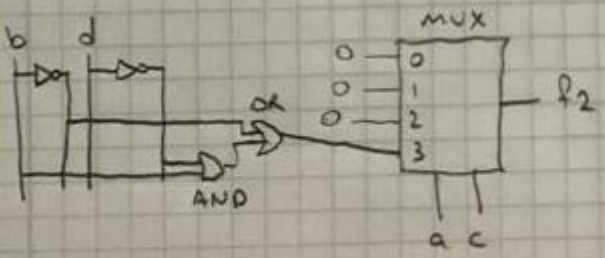


$f_1 = a'c(b) + ac'(d) + ac(b \oplus d)$

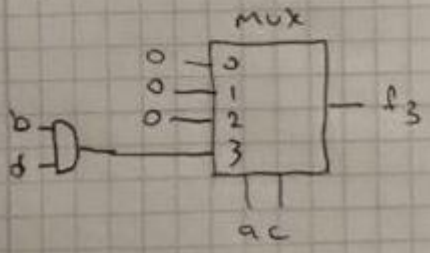


$f_2 = ab'cd' + ab'cd + abcd'$

$= ac(b' + bd')$



$f_3 = abcd$



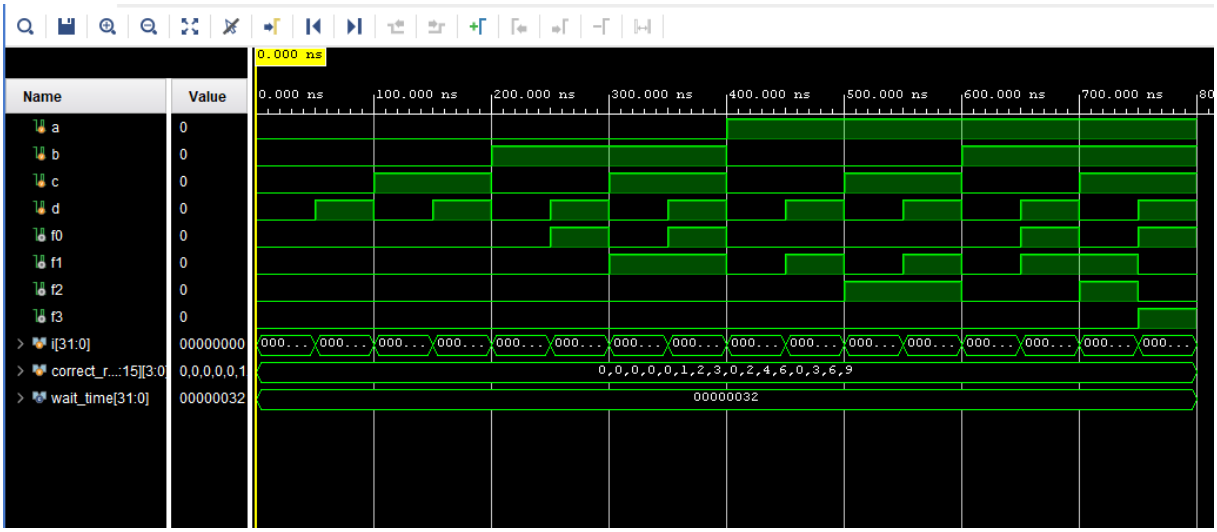
Source Code:

```
`timescale 1ns / 1ps

module with_mux(
input a,b,c,d,
output f3,f2,f1,f0
    );


wire bd;
wire b_not;
wire d_not;

//f0
and_gate AND1(.I1(b),.I2(d),.O(bd));
MUX MUX_f0(.D({bd,bd,bd,bd}),.S({a,c}),.O(f0));

//f1
wire xorf1;
xor_gate XOR1 (.I1(b),.I2(d),.O(xorf1));
MUX MUX_f1(.D({xorf1,d,b,1'b0}),.S({a,c}),.O(f1));
//f2
wire bdnot;
wire bnotorbdnot;
not_gate NOT1(.I(b),.O(b_not));
not_gate NOT2(.I(d),.O(d_not));
and_gate AND2 (.I1(b),.I2(d_not),.O(bdnot));
or_gate OR1 (.I1(bdnot), .I2(b_not),.O(bnotorbdnot));
MUX MUX_f2(.D({bnotorbdnot,1'b0,1'b0,1'b0}),.S({a,c}),.O(f2));
//f3
MUX MUX_f3 (.D({bd,1'b0,1'b0,1'b0}),.S({a,c}),.O(f3));

endmodule
```

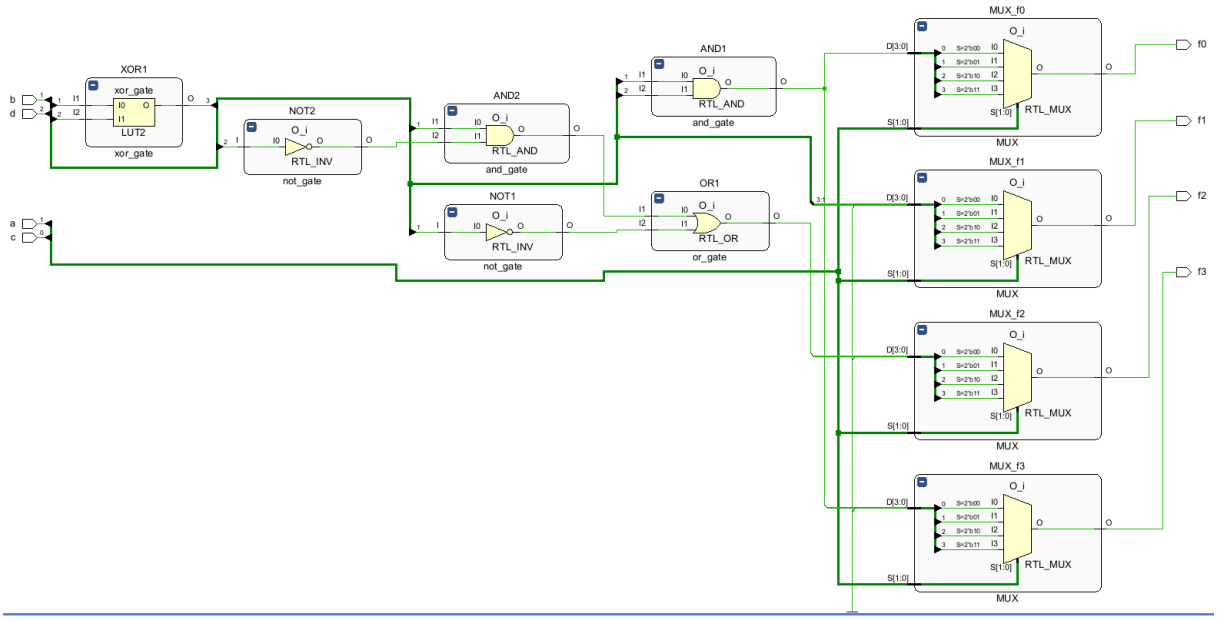
Testbench results:

```
{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/Progg/
```

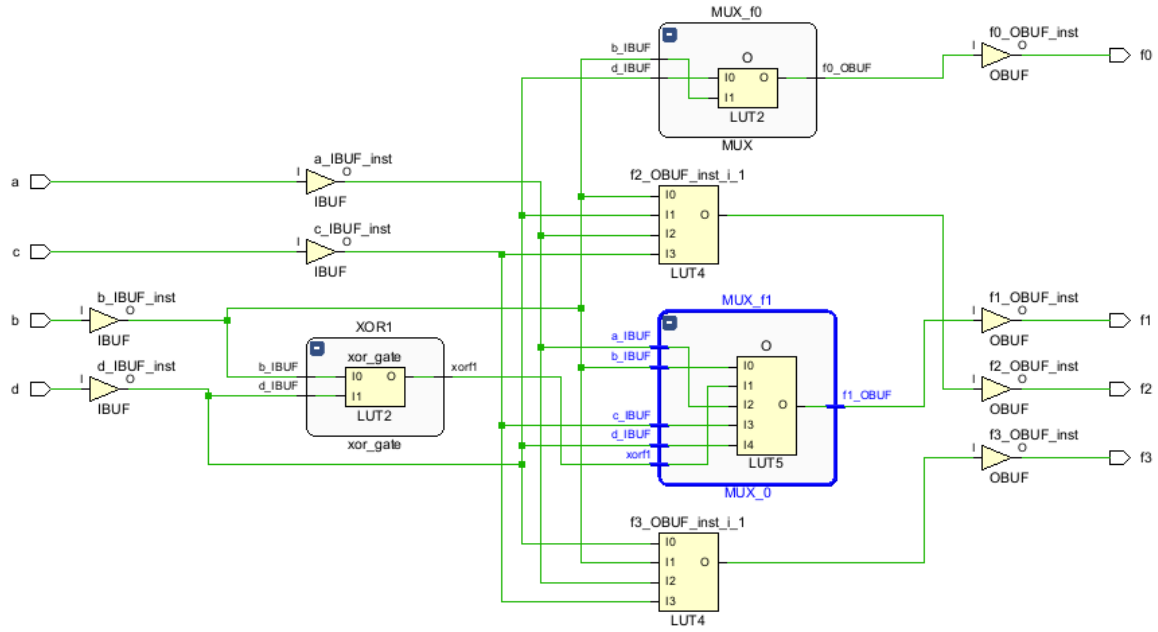
Behavioral simulation results:



RTL Schematic:



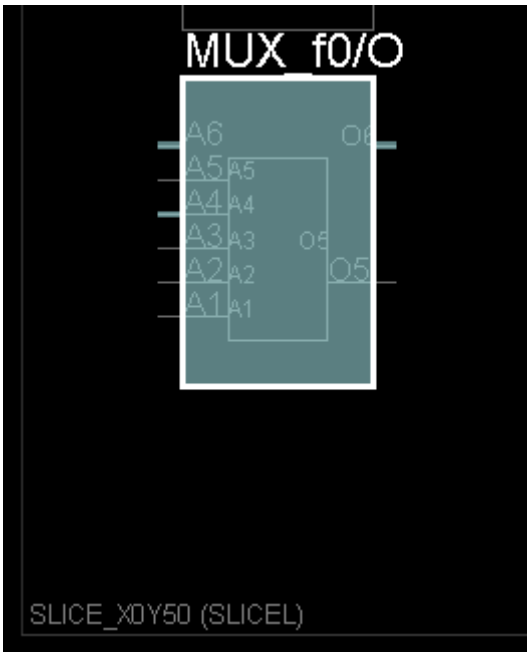
Technology Schematic:

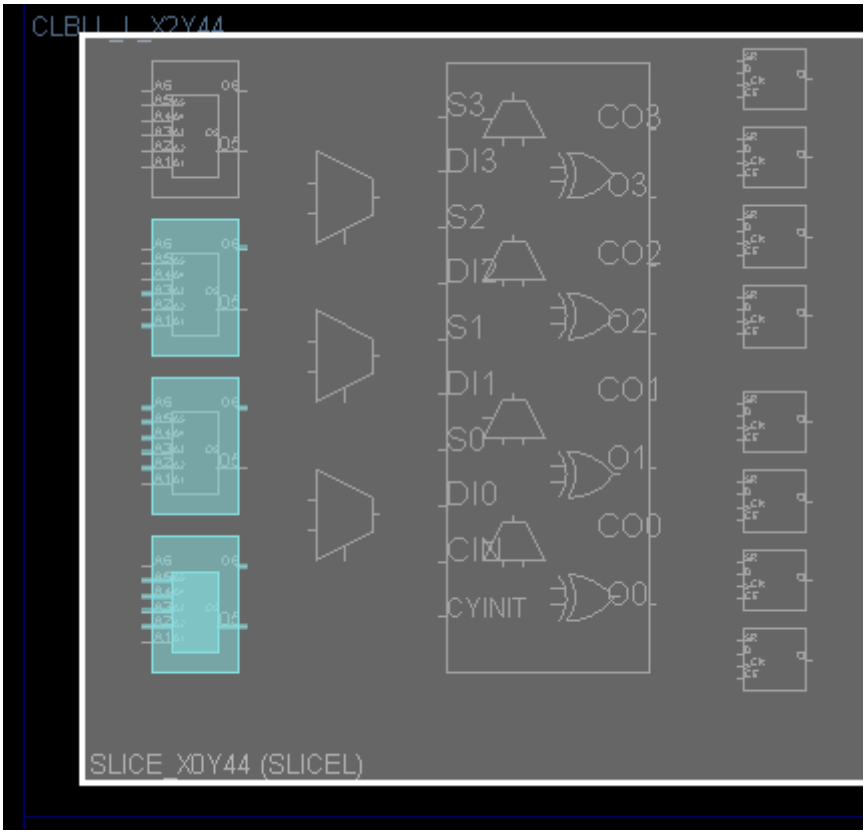


Pad to pad delays without constraints:

From Port	To Port	M a 1	Max Process Corner	Min Delay	Min Process Corner
a	f2	9.726	SLOW	3.016	FAST
d	f2	9.494	SLOW	2.916	FAST
a	f3	9.229	SLOW	2.827	FAST
b	f2	9.127	SLOW	2.769	FAST
a	f1	9.046	SLOW	2.782	FAST
d	f3	8.999	SLOW	2.724	FAST
d	f1	8.954	SLOW	2.666	FAST
b	f0	8.911	SLOW	2.677	FAST
b	f1	8.894	SLOW	2.522	FAST
c	f2	8.662	SLOW	2.613	FAST
b	f3	8.633	SLOW	2.582	FAST
c	f3	8.199	SLOW	2.419	FAST
d	f0	8.109	SLOW	2.399	FAST
c	f1	8.018	SLOW	2.371	FAST

Zooming FPGA:

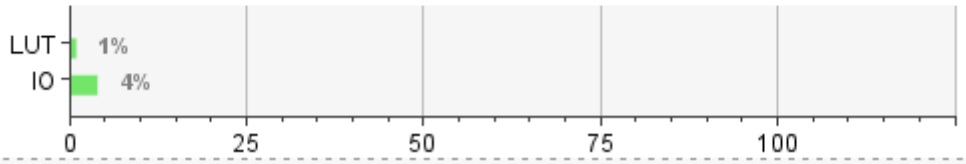




These are the LUTs used in my device.

Utilization report:

Resource	Utilization	Available	Utilization %
LUT	4	32600	0.01
IO	8	210	3.81



There are 4 LUTs are used.

Comparison:

MUX : 4 LUT

Decoder : 3 LUT

SSI : 2 LUT

So here is the results, MUX uses most place and the gate level type uses least.

With 6ns timing constraint pad to pad delays

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
b	f2	8.902	SLOW
a	f2	8.835	SLOW
d	f1	8.786	SLOW
b	f1	8.692	SLOW
d	f2	8.655	SLOW
c	f2	8.321	SLOW
d	f3	8.306	SLOW
a	f3	8.299	SLOW
a	f1	8.298	SLOW
b	f0	8.246	SLOW
c	f1	8.149	SLOW
b	f3	8.133	SLOW
d	f0	8.092	SLOW
c	f3	7.881	SLOW

4)

-Behavioral simulation is wave representation of RTL schematic.
Post synthesis simulation is the simulation of programs synthesized circuit and netlist. If there is any difference created by program, it will bee seen in this step.
Post implementation simulation is the end step that we can understand what we are going to exactly get and what were we targetting.

-According the truth table, the operation is multiplication of 2 2 bit number.

-Evaluating the designs:

Design difficulty: Decoder > MUX > Gate Level

Coding difficulty: Gate level > Decoder > MUX

LUT usage: MUX > Decoder > Gate level

Path Delays: Depends on application type.