#100 DAYS OF RTL

DAY 4/100 Insights

MUX TREES

WITH SHANNON EXPANSION THEOREM

Understanding Shanon expansion Theorem

- Any Boolean function f(w_1,w_2,…,w_n) can be written in the form

$$f(w_1, w_2, \ldots, w_n) = \overline{w}_1 \cdot f(0, w_2, \ldots, w_n) + w_1 \cdot f(1, w_2, \ldots, w_n)$$

- or

$$f(w_1, w_2, \ldots, w_n) = \overline{w}_1 \cdot f_{\overline{w}_1} + w_1 \cdot f_{w_1}$$

- where $f_{\overline{w}_1}$ is called the cofactor of f with respect to $\overline{w}_1$ and f_{w1} is called the cofactor of f with respect to w_1

- Shannon's Expansion expresses a given logic function in terms of select inputs and data inputs required for logic synthesis using a multiplexer

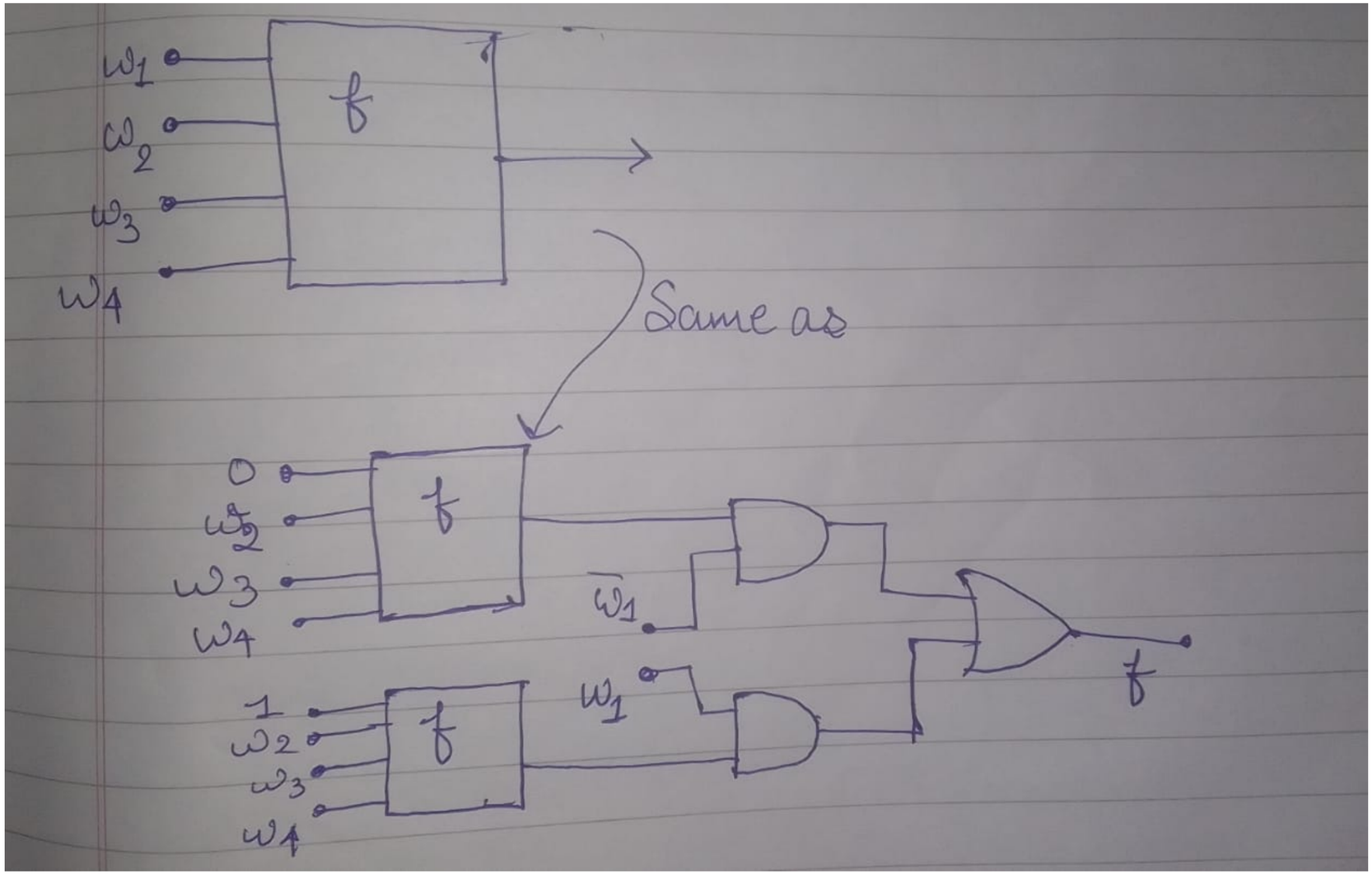
Applications of Shannon Expansion

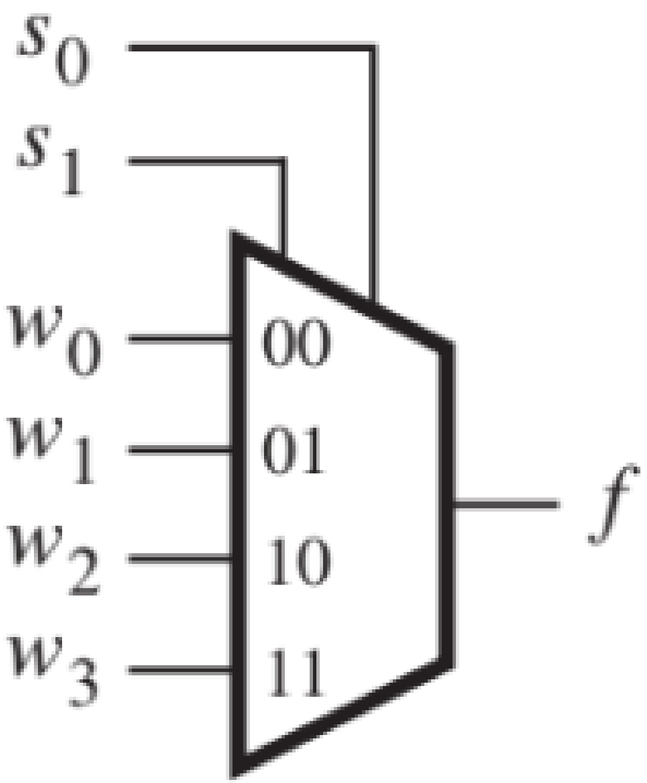
Reduce the power consumption in digital system

After synthesis, technology mapping and placement are complete, we apply Shannons expansion to the most critical sections of the circuit. This approach allows us to precompute the values of functions that depend on late-arriving critical signals and use a multiplexer to quickly select the appropriate value when the signal arrives. Any new logic elements created by this technique are incrementally placed in a minimally disruptive fashion to ensure convergence between the circuit optimization and the netlist placement.

It is very useful in Actel FPGAs

Visual Representation of Shannon expansion theorem





(a) Graphical symbol

s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

(b) Truth table

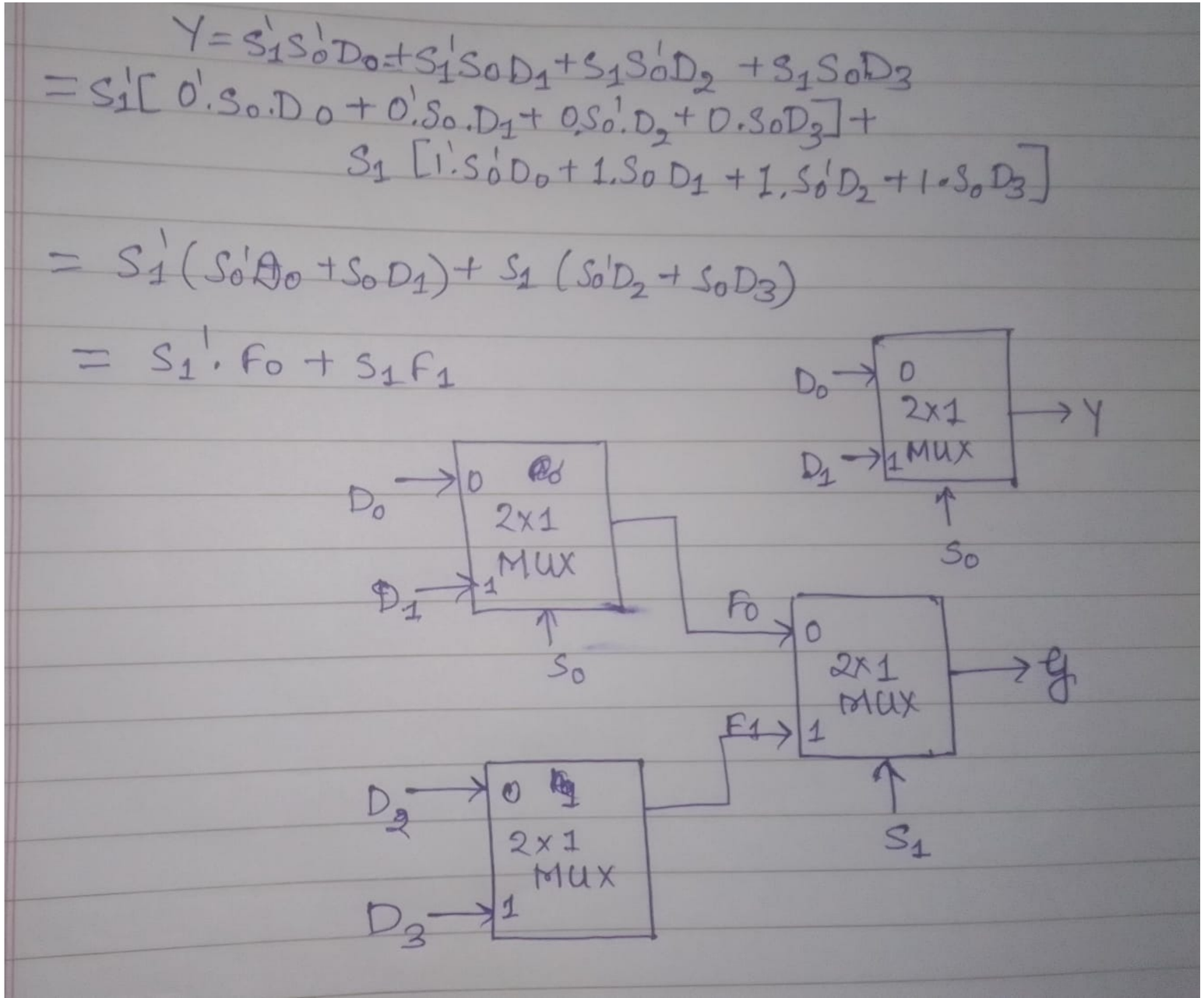
How we can use it to implement MUX TREE

$$Y = S_1' S_0' D_0 + S_1' S_0 D_1 + S_1 S_0' D_2 + S_1 S_0 D_3$$

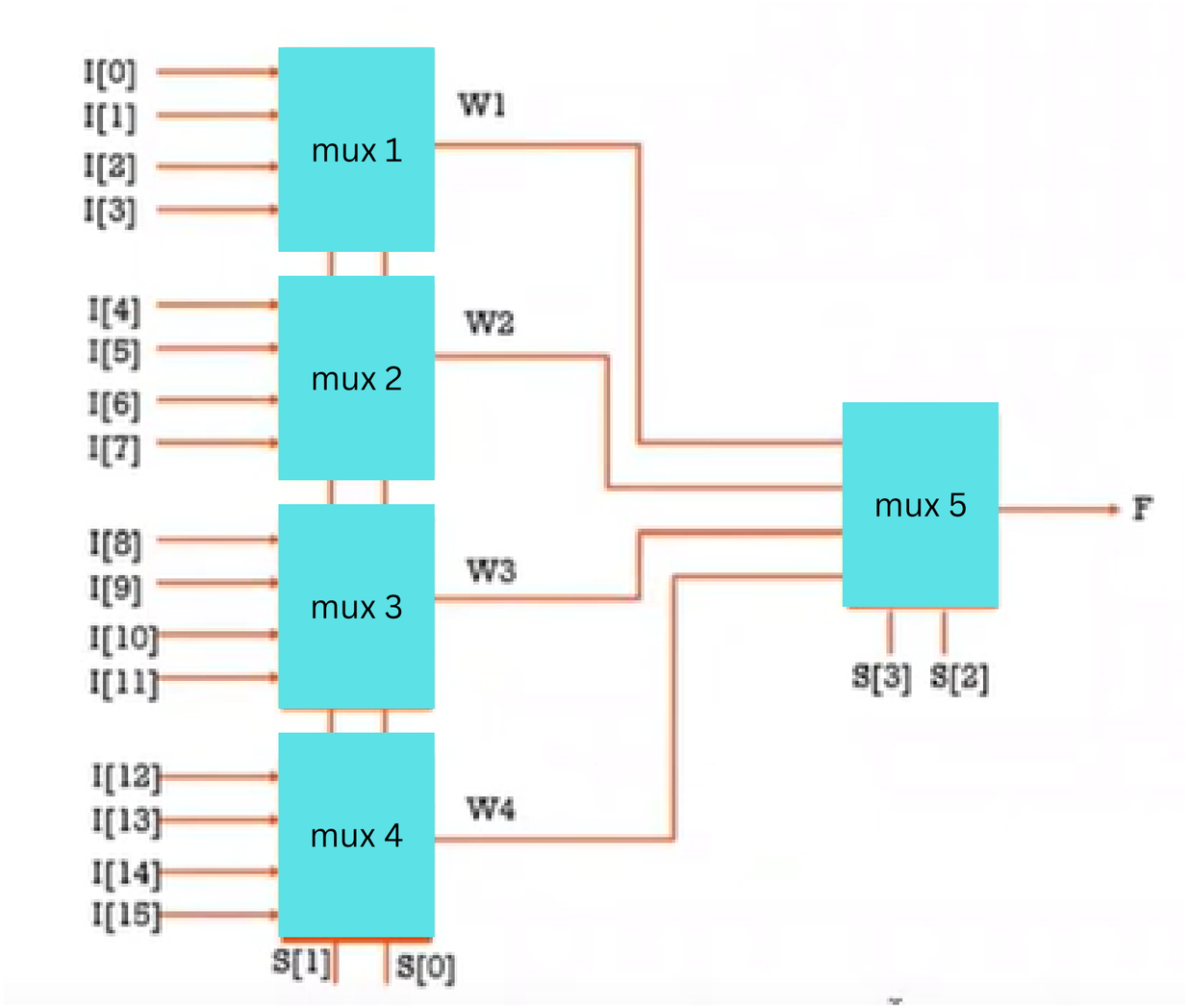
$$= S_1' [0' \cdot S_0 \cdot D_0 + 0' \cdot S_0 \cdot D_1 + 0 \cdot S_0' \cdot D_2 + 0 \cdot S_0 D_3] +$$
$$S_1 [1' \cdot S_0' D_0 + 1 \cdot S_0 D_1 + 1 \cdot S_0' D_2 + 1 \cdot S_0 D_3]$$

$$= S_1' (S_0' D_0 + S_0 D_1) + S_1 (S_0' D_2 + S_0 D_3)$$

$$= S_1' \cdot F_0 + S_1 F_1$$



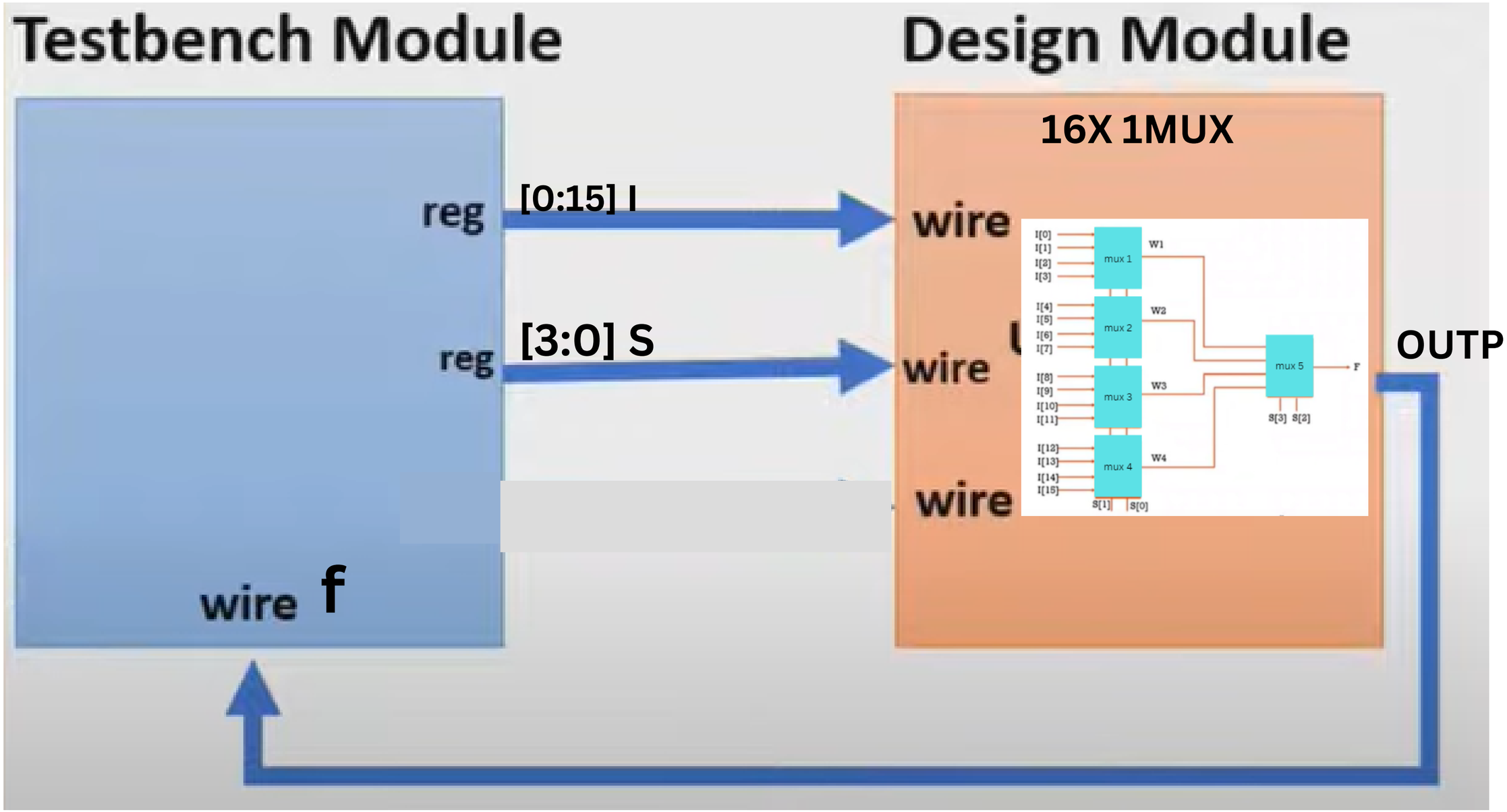
16X1 MUX using 4x1 MUX



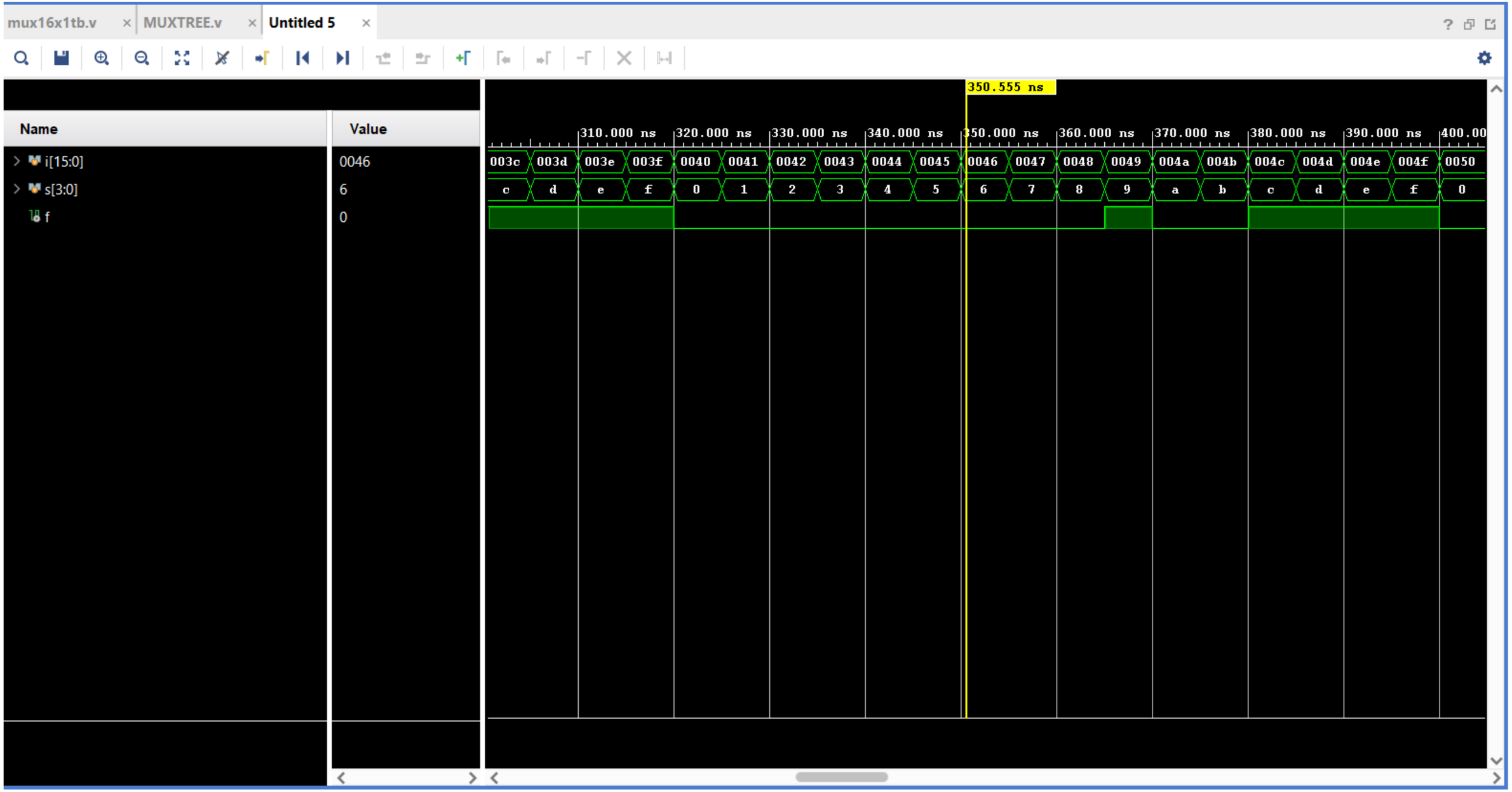

```
module mux16to1 (I, S, f);
input [0:15] I;
input [3:0] S;
output f;
wire [0:3] W;
mux4to1 Mux1 (I[0:3], S[1:0], W[0]);
mux4to1 Mux2 (I[4:7], S[1:0], W[1]);
mux4to1 Mux3 (I[8:11], S[1:0], W[2]);
mux4to1 Mux4 (I[12:15], S[1:0], W[3]);
mux4to1 Mux5 (W[0:3], S[3:2], f);
endmodule

module mux4to1 (W, S, f);
input [0:3] W;
input [1:0] S;
output reg f;
always @(*)
if (S == 2'b00)
f = W[0];
else if (S == 2'b01)
f = W[1];
else if (S == 2'b10)
f = W[2];
else
f = W[3];
endmodule
```

TEST BENCH


Testbench Module
Design Module
16X 1MUX
reg
[0:15] I
reg
[3:0] S
wire
wire
wire
wire f
OUTPUT f
I[0]
I[1]
I[2]
I[3]
mux 1
W1
I[4]
I[5]
I[6]
I[7]
mux 2
W2
I[8]
I[9]
I[10]
I[11]
mux 3
W3
I[12]
I[13]
I[14]
I[15]
mux 4
W4
mux 5
F
S[3]
S[2]
S[1]
S[0]

```
module mux_16_1_using_4_1_TB;
//inputs are reg
reg [15:0]i;
 reg [3:0]s;
//outputs are wires
wire f;
//instantiate UUT
mux16to1 mux16x1(.I(i),.S(s),.f(f));
 initial
 begin
 $monitor($time, "f=%b,i=%b,s=%b",f,i,s); //initialize inputs
  i=0;
s=0;
end
always #5 i=i+1;
always #5 s = s +1;
endmodule
```


mux16x1tb.v
MUXTREE.v
Untitled 5
Name
Value
i[15:0]
0046
s[3:0]
6
f
0
350.555 ns
310.000 ns
320.000 ns
330.000 ns
340.000 ns
350.000 ns
360.000 ns
370.000 ns
380.000 ns
390.000 ns
400.00
003c 003d 003e 003f 0040 0041 0042 0043 0044 0045 0046 0047 0048 0049 004a 004b 004c 004d 004e 004f 0050
c d e f 0 1 2 3 4 5 6 7 8 9 a b c d e f 0

16x1MUXwith4x1MUX - [C:/Users/HP/Desktop/100daysofrt/DAY_4_MUX_TREE/16x1MUXwith4x1MUX/16x1MUXwith4x1MUX.xpr] - Vivado 2023.2

File Edit Flow Tools Reports Window Layout View Help Quick Access Ready

Default Layout

Flow Navigator

- **PROJECT MANAGER**
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- **IP INTEGRATOR**
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- **SIMULATION**
  - Run Simulation
- **RTL ANALYSIS**
  - Run Linter
  - Open Elaborated Design
    - Report Methodology
    - Report DRC
    - Report Noise
    - Schematic
- **SYNTHESIS**
  - Run Synthesis
  - Open Synthesized Design

ELABORATED DESIGN - xc7a100tcsg324-2

Sources | Netlist

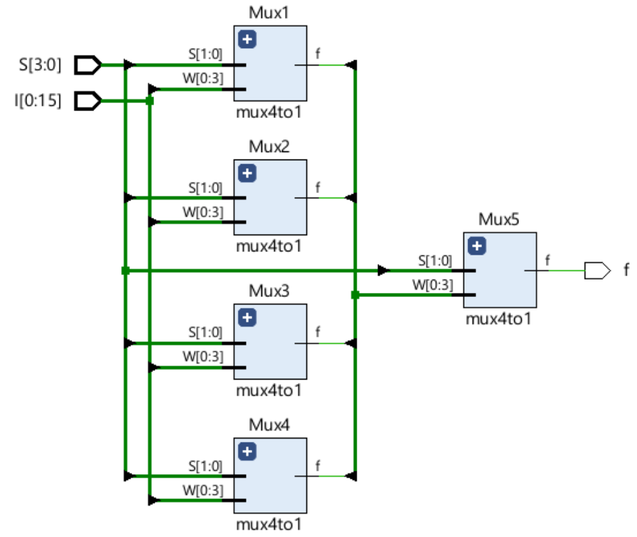
- mux16to1
  - Nets (25)
  - Mux1 (mux4to1)
  - Mux2 (mux4to1)
  - Mux3 (mux4to1)
  - Mux4 (mux4to1)
  - Mux5 (mux4to1)

Source File Properties

Select an object to see properties

Project Summary | Schematic | mux16x1tb.v | MUXTREE.v

5 Cells 21 I/O Ports 25 Nets



Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start	Elapsed
synth_1	constrs_1	Not started																		
impl_1	constrs_1	Not started																		

Type here to search ENG 11:21 30-11-2023