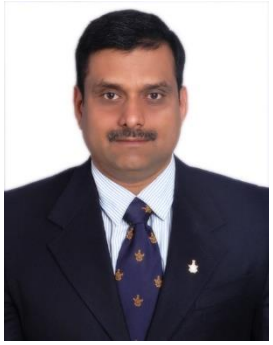
BITS Pilani
Hyderabad Campus
Department of Electrical Engineering

Digital Design : 2020-21
Lab 9
Combinational Circuit Design using Decoders and Multiplexers in Xilinx

By Dr. Sanjay Vidhyadharan

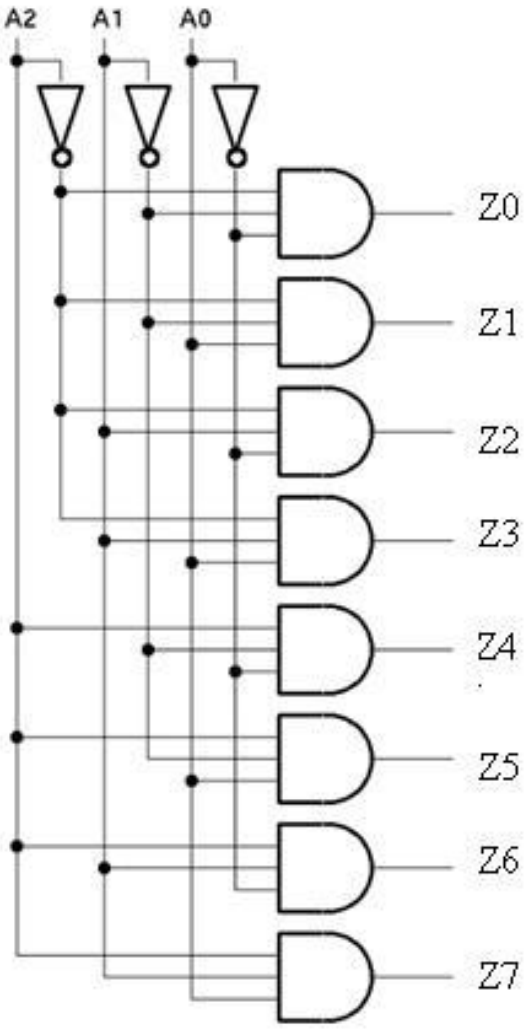


New Concepts

Conditional Statements

The Decoder

```
module
Decoder(a,b,c,d0,d1,d2,d3,d4,d5,d6,d7);
input a,b,c;
output d0,d1,d2,d3,d4,d5,d6,d7;
assign d0=(~a&~b&~c),
d1=(~a&~b&c),
d2=(~a&b&~c),
d3=(~a&b&c),
d4=(a&~b&~c),
d5=(a&~b&c),
d6=(a&b&~c),
d7=(a&b&c);
endmodule
```



The Conditional Statemnet

If statement in Verilog

```
if (sel == 1) f = a;
else f = b;
```

Example : X is output of a Decoder, Y is output Function

```
begin
        if (x[3] == 1'b1)  y = 2'b11;
        else if (x[2] == 1'b1)  y = 2'b10;
        else if (x[1] == 1'b1)  y = 2'b01;
        else  y = 2'b00;
end
```

The Decoder

```
module decoder (in,out);
input [2:0] in;
output [7:0] out;
wire [7:0] out;
assign out =
(in == 3'b000 ) ? 8'b0000_0001 :
(in == 3'b001 ) ? 8'b0000_0010 :
(in == 3'b010 ) ? 8'b0000_0100 :
(in == 3'b011 ) ? 8'b0000_1000 :
(in == 3'b100 ) ? 8'b0001_0000 :
(in == 3'b101 ) ? 8'b0010_0000 :
(in == 3'b110 ) ? 8'b0100_0000 :
(in == 3'b111 ) ? 8'b1000_0000 :
8'h00;
endmodule
```

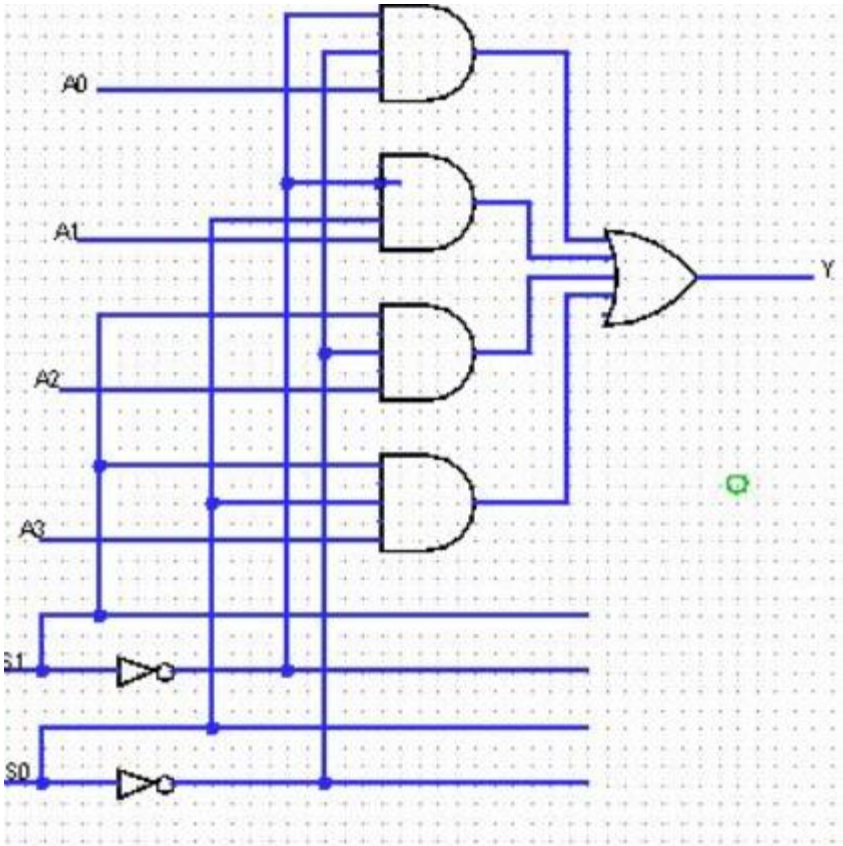
The Decoder

Problem 1 : Design an Odd Parity Generator using a Decoder. Use Decoder as a verilog module.

```
module Parity(
    input [2:0] A,
    output F
    );
```

The Multiplexer

```
module m41 (out, a, b, c, d, s0, s1);
output out;
input a, b, c, d, s0, s1;
wire sobar, s1bar, T1, T2, T3, T4;
not (s0bar, s0), (s1bar, s1);
and (T1, a, s0bar, s1bar),
(T2, b, s0bar, s1),
(T3, c, s0, s1bar),
(T4, d, s0, s1);
or(out, T1, T2, T3, T4);
endmodule
```



The Multiplexer

```
module Mux(
    input [7:0] I,
    input [2:0] S,
    output Y
    );
assign Y = I[S];
endmodule
```

The Multiplexer based FA

```
module FullAdder(
    input A,
    input B,
    input C,
    output Sum
    );
wire [7:0] s;
assign s = 8;
Mux m1 (s,{A,B,C},Sum);
endmodule
```

```
module Mux(
    input [7:0] I,
    input [2:0] S,
    output Y
    );
assign Y = I[S];
endmodule
```

The Multiplexer based FA

Home Assignments

1. Parity Generator with Mux
2. Full Adder with Decoder