Hacettepe University

Computer Engineering Department

BM233 Logic Design Lab - 2021 Fall

# Combinational Circuits in Verilog

December 3, 2021

*Student name:*
Osman Faruk Derdiyok

*Student Number:*
b21946036

# 1 Problem Definition

Combinational circuits are circuits whose outputs, at any instant of time, depend only on the present inputs (the combinational circuits do not use any memory elements). That is, the previous inputs or state of the circuit do not have any effect on its present state.

A combinational circuit performs a specific information-processing operation fully specified logically by a set of Boolean functions. The 'n' input variables come from an external source whereas the 'm' output variables go to an external destination. In many applications, the source or destination are storage registers.

Decoder + Mux = Circuit

# 2 Solution Implementation (decoder_2x4.v)

```
module decoder_2x4(
    input[1:0] A,
    output[3:0] D
);
    assign D[0] = (~A[1] && ~A[0]);
    assign D[1] = (~A[1] && A[0]);
    assign D[2] = (A[1] && ~A[0]);
    assign D[3] = (A[1] && A[0]);
endmodule
```

# 3 Testbench Implementation (decoder_2x4_tb.v)

Testbench code where we tried all cases for decoder.

```
`timescale 1ns / 1ps
module decoder_2x4_tb;
    reg[1:0] A_tb;
    wire[3:0] D_tb;

    decoder_2x4 P1(.A(A_tb),.D(D_tb));

    initial begin
    #0 A_tb[1]=0; A_tb[0]=0;
    #50 A_tb[1]=0; A_tb[0]=1;
    #50 A_tb[1]=1; A_tb[0]=0;
    #50 A_tb[1]=1; A_tb[0]=1;
    #50 $finish();
    end
endmodule
```

# 4 Solution Implementation (mux_4x1.v)

```
module mux_4x1(
    input[3:0] i,
    input[1:0] s,
    output F
);
    reg F;
    always @(i, s) begin
        case (s)
            0: F = i[0];
            1: F = i[1];
            2: F = i[2];
            3: F = i[3];
        endcase
     end
endmodule
```

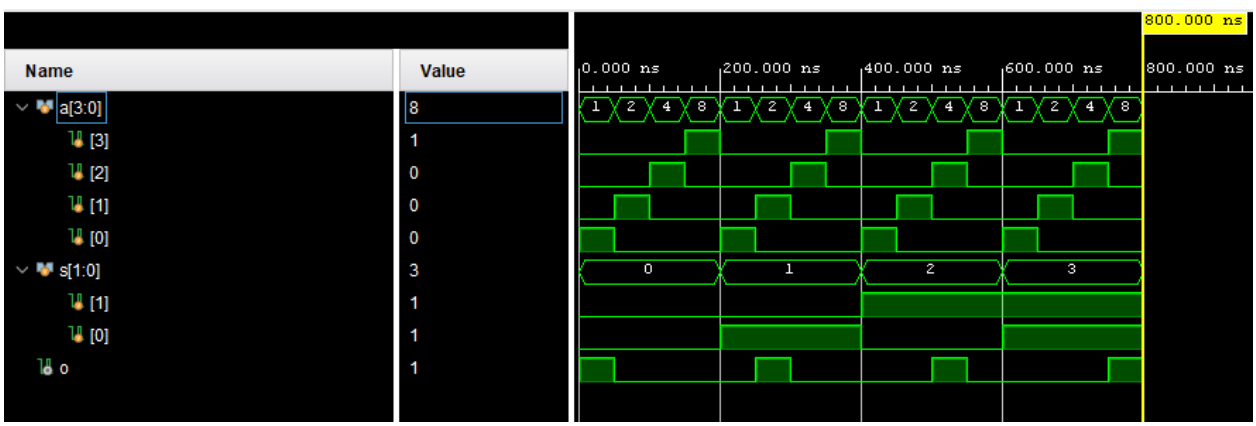


Figure 1: Resulting MUX's Waveform

*figure 1 will we explained in the result part.

## 5 Testbench Implementation (mux_4x1_tb.v)

Testbench code where we tried all cases for 4x1 MUX.

```
`timescale 1ns / 1ps


module mux_4x1_tb();
    reg [3:0] a;
    reg [1:0] s;
    wire o;

    mux_4x1 DUT(.i(a), .s(s), .F(o));

    initial begin

        s=0;
        a[0]=1 ; a[1]=0 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=1 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=1 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=0 ; a[3]=1 ; #50

        s=1;
        a[0]=1 ; a[1]=0 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=1 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=1 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=0 ; a[3]=1 ; #50

        s=2;
        a[0]=1 ; a[1]=0 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=1 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=1 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=0 ; a[3]=1 ; #50

        s=3;
        a[0]=1 ; a[1]=0 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=1 ; a[2]=0 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=1 ; a[3]=0 ; #50
        a[0]=0 ; a[1]=0 ; a[2]=0 ; a[3]=1 ; #50
        $finish();
    end

endmodule
```

## 6 Solution Implementation (circuit.v)

```
module circuit(
    input a,
    input b,
    input c,
    input d,
    output F
);
    wire[3:0] outDecoder;
    decoder_2x4 D1({a,b},outDecoder);
    mux_4x1 D2(outDecoder,{c,d},F);
endmodule
```

## 7 Testbench Implementation (circuit_tb.v)

Testbench code where we tried all cases for decoder.

```
module circuit_tb;

reg a,b,c,d;
wire F;
circuit C(.a(a), .b(b), .c(c), .d(d), .F(F));

initial begin
    a = 0; b = 0; c=0; d=0; #50
    a = 0; b = 0; c=0; d=1; #50
    a = 0; b = 0; c=1; d=0; #50
    a = 0; b = 0; c=1; d=1; #50

    a = 0; b = 1; c=0; d=0; #50
    a = 0; b = 1; c=0; d=1; #50
    a = 0; b = 1; c=1; d=0; #50
    a = 0; b = 1; c=1; d=1; #50

    a = 1; b = 0; c=0; d=0; #50
    a = 1; b = 0; c=0; d=1; #50
    a = 1; b = 0; c=1; d=0; #50
    a = 1; b = 0; c=1; d=1; #50

    a = 1; b = 1; c=0; d=0; #50
    a = 1; b = 1; c=0; d=1; #50
    a = 1; b = 1; c=1; d=0; #50
    a = 1; b = 1; c=1; d=1; #50
    $finish();
end
endmodule
```

# 8 Results

There is a 3 result. Respectively, decoder, multiplexer, main circuit. Actually, we don't need the first two one but we can check the accuracy the steps thanks to them.

The funtion: $F(a, b, c, d) = \sum(0, 5, 10, 15)$

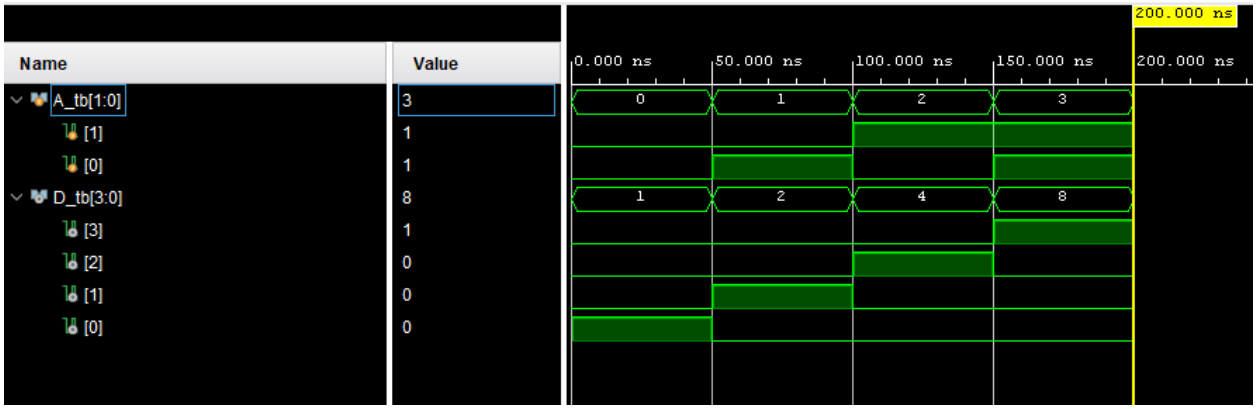


Figure 2: Resulting Decoder's Waveform

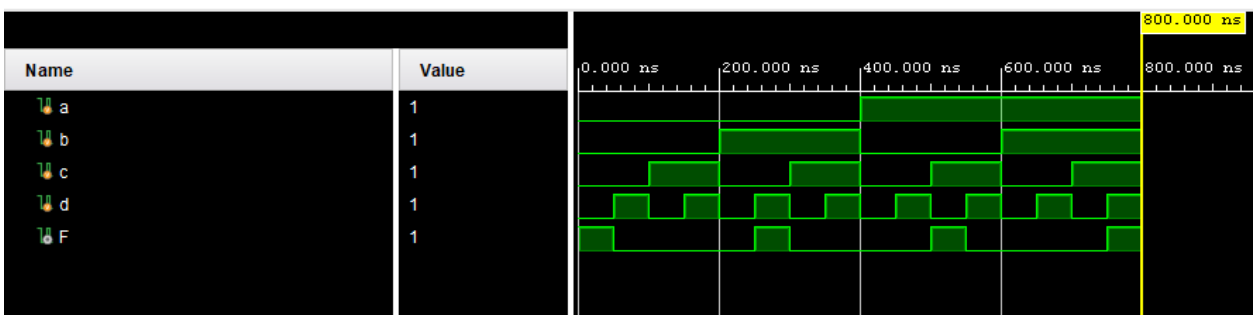


Figure 3: Resulting Circuit's Waveform

## 9 Notes

Problem: Due to the most significant ranking of a and b in the inputs we gave, my result was the opposite.
Solution: The parameter we get the decoder is reversed (i[0] – i[1])


Figure 4: Wrong Order

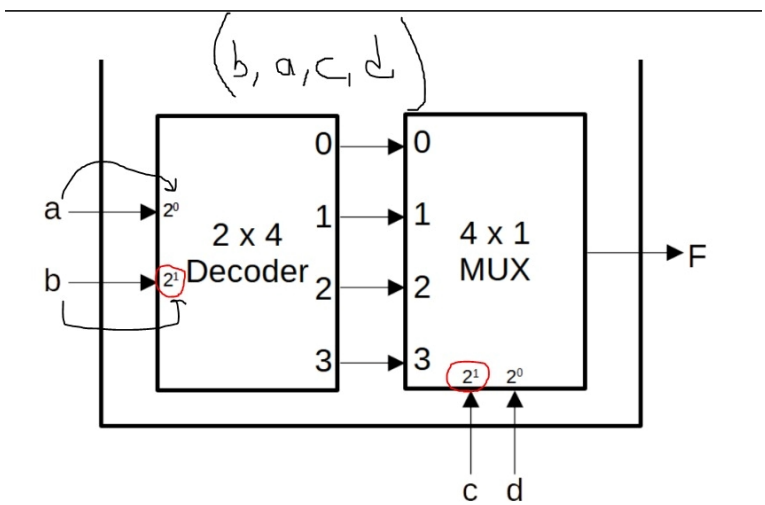


Figure 5: Order of parameters i need before i fix the error

Problem: When using another module, we should pay attention to the parameter order that we need to give. Otherwise it may cost you 5 6 hours. :)
Solution: Change order :D

```
wire[3:0] outDecoder;
decoder_2x4 D1({a,b},outDecoder);
mux_4x1 D2(outDecoder,{c,d},F);
```

Figure 6: Parameter order

## References


- 231 Lecture notes
- 233 Lecture notes
- https://tex.stackexchange.com/questions/48632/underscores-in-words-text/48633