



# CS 223 Digital Design

Bilkent University  
Spring 2021/2022

## Laboratory Assignment 3 Preliminary Report

### Section 2

Deniz Tuna Onguner  
22001788

Mon. March 14<sup>th</sup>, 2022

## 1 1-to-2 Decoder System Verilog Module

```
module decoder1to2(input logic i, s, output logic y0, y1);
    assign y0 = ~i & s;
    assign y1 = i & s;
endmodule
```

### 1-to-2 Decoder Testbench

```
module decoder1to2_TestBench();
    logic i, s, y0, y1;
    decoder1to2 dut(i, s, y0, y1);
    initial begin
        i = 0; s = 0; #10;
        i = 1;      #10;
        i = 0; s = 1; #10;
        i = 1;      #10;
    end
endmodule
```

## 2 2-to-4 Decoder System Verilog Module

```
module decoder2to4(input logic i0, i1, s,
                    output logic y0, y1, y2, y3);
    logic temp0, temp1;
    decoder1to2 dec1(i0, s, temp0, temp1);
    decoder1to2 dec2(i1, temp1, y1, y0);
    decoder1to2 dec3(i1, temp0, y3, y2);
endmodule
```

### 2-to-4 Decoder Testbench

```
module decoder2to4_TestBench();
    logic i0, i1, s, y0, y1, y2, y3;
    decoder2to4 dut(i0, i1, s, y0, y1, y2, y3);
    initial begin
        s = 0; i0 = 0; i1 = 0; #10;
        s = 1;      #10;
        i1 = 1; #10;
        i0 = 1; i1 = 0; #10;
        i1 = 1; #10;
    end
endmodule
```

### 3 2-to-1 Multiplexer System Verilog Module

```
module mux2to1(input logic i0, i1, s, e, output logic y);
    logic temp0, temp1, temp2;
    assign temp0 = i0 & ~s;
    assign temp1 = i1 & s;
    assign temp2 = temp0 | temp1;
    assign y = temp2 & e;
endmodule
```

#### 2-to-1 Multiplexer Testbench

```
module mux2to1_TestBench();
    logic i0, i1, s, y;
    mux2to1 dut(i0, i1, s, y);
    initial begin
        i0 = 0; i1 = 0; s = 0; #10; s = 1; #10;
        i1 = 1; s = 0; #10; s = 1; #10;
        i0 = 1; i1 = 0; s = 0; #10; s = 1; #10;
        i1 = 1; s = 0; #10; s = 1; #10;
    end
endmodule
```

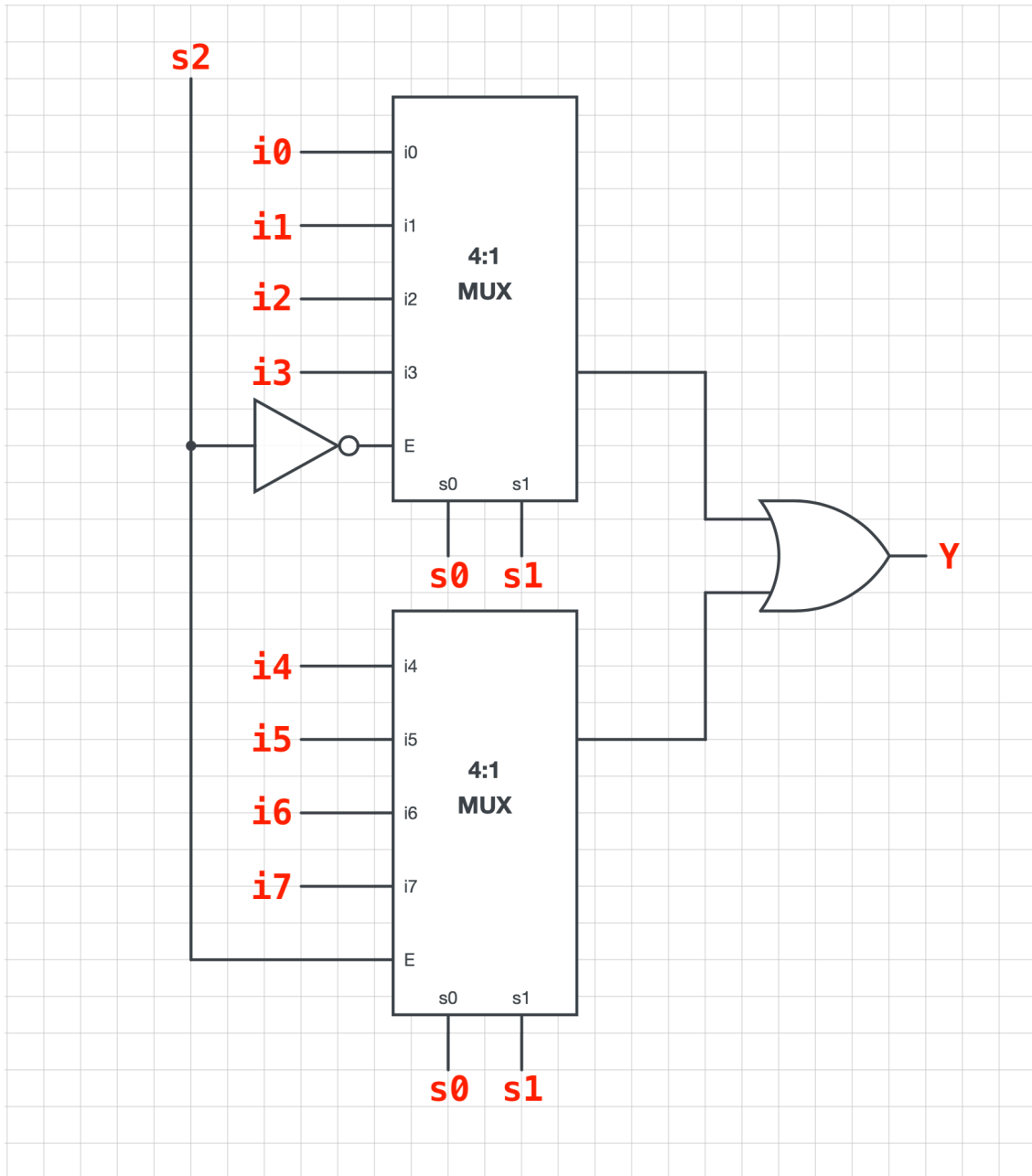
### 4 4-to-1 Multiplexer System Verilog Module

```
module mux4to1(input logic i0, i1, i2, i3, s0, s1, e,
               output logic y);
    logic temp0, temp1;
    mux2to1 mux1(i0, i1, s0, e, temp0);
    mux2to1 mux2(i2, i3, s0, e, temp1);
    mux2to1 mux3(temp0, temp1, s1, e, y);
endmodule
```

#### 4-to-1 Multiplexer Testbench

```
module mux4to1_TestBench();
    logic i0, i1, i2, i3, s0, s1, y;
    mux4to1 dut(i0, i1, i2, i3, s0, s1, y);
    initial begin
        s0 = 0; s1 = 0; i0 = 1; i1 = 0; i2 = 0; i3 = 0; #10;
        s0 = 0; s1 = 1; i0 = 0; i1 = 1; i2 = 0; i3 = 0; #10;
        s0 = 1; s1 = 0; i0 = 0; i1 = 0; i2 = 1; i3 = 0; #10;
        s0 = 1; s1 = 1; i0 = 0; i1 = 0; i2 = 0; i3 = 1; #10;
    end
endmodule
```

## 5 8-to-1 Multiplexer Block Design



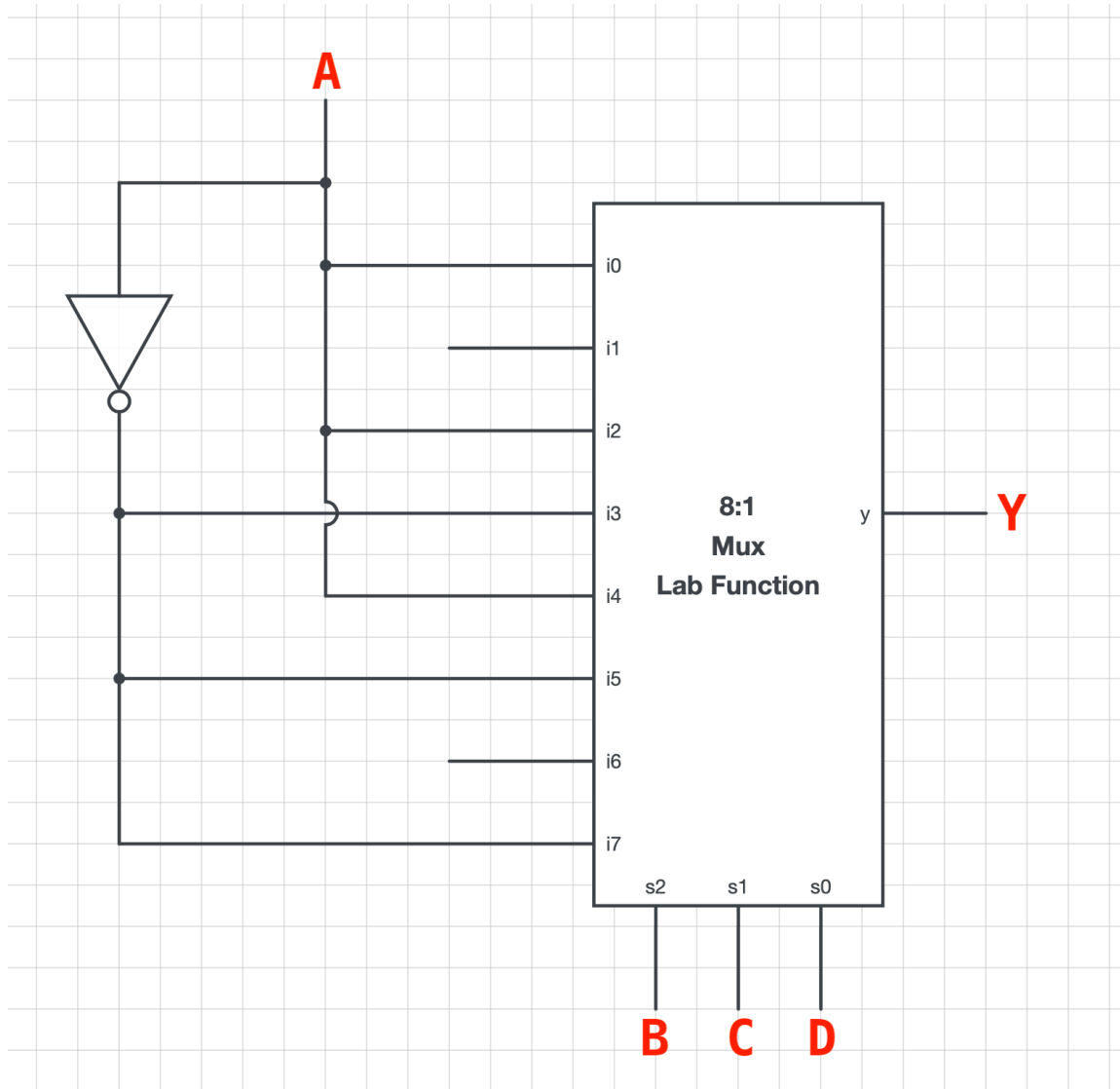
## 8-to-1 Multiplexer System Verilog Module

```
module mux8to1(input logic i0, i1, i2, i3,
               i4, i5, i6, i7, s0, s1, s2, output logic y);
    logic temp0, temp1;
    mux4to1 mux1(i0, i1, i2, i3, s0, s1, ~s2, temp0);
    mux4to1 mux2(i4, i5, i6, i7, s0, s1, s2, temp1);
    assign y = temp0 | temp1;
endmodule
```

## 8-to-1 Multiplexer Testbench

```
module mux8to1_TestBench();
    logic i0, i1, i2, i3, i4, i5, i6, i7,
           s0, s1, s2, y;
    mux8to1 dut(i0, i1, i2, i3, i4, i5, i6, i7,
                s0, s1, s2, y);
    initial begin
        i0 = 0; i1 = 0; i2 = 0; i3 = 0; i4 = 0; i5 = 0; i6 = 0;
        i7 = 0; s0 = 0; s1 = 0; s2 = 0; #10;
        i0 = 1; i1 = 0; i2 = 0; i3 = 0; i4 = 0; i5 = 0; i6 = 0;
        i7 = 0; s0 = 0; s1 = 0; s2 = 0; #10;
        i0 = 0; i1 = 1; i2 = 0; i3 = 0; i4 = 0; i5 = 0; i6 = 0;
        i7 = 0; s0 = 1; s1 = 0; s2 = 0; #10;
        i0 = 0; i1 = 0; i2 = 1; i3 = 0; i4 = 0; i5 = 0; i6 = 0;
        i7 = 0; s0 = 0; s1 = 1; s2 = 0; #10;
        i0 = 0; i1 = 0; i2 = 0; i3 = 1; i4 = 0; i5 = 0; i6 = 0;
        i7 = 0; s0 = 1; s1 = 1; s2 = 0; #10;
        i0 = 0; i1 = 0; i2 = 0; i3 = 0; i4 = 1; i5 = 0; i6 = 0;
        i7 = 0; s0 = 0; s1 = 0; s2 = 1; #10;
        i0 = 0; i1 = 0; i2 = 0; i3 = 0; i4 = 0; i5 = 1; i6 = 0;
        i7 = 0; s0 = 1; s1 = 0; s2 = 1; #10;
        i0 = 0; i1 = 0; i2 = 0; i3 = 0; i4 = 0; i5 = 0; i6 = 1;
        i7 = 0; s0 = 0; s1 = 1; s2 = 1; #10;
        i0 = 0; i1 = 0; i2 = 0; i3 = 0; i4 = 0; i5 = 0; i6 = 0;
        i7 = 1; s0 = 1; s1 = 1; s2 = 1; #10;
    end
endmodule
```

## 6 Lab Function Block Design



## Lab Function System Verilog Module

```
module lab_func(input logic a, b, c, d, output logic y);  
    mux8to1 mux(a, 0, a, ~a, a, ~a, 0, ~a, d, c, b, y);  
endmodule
```

## Lab Function Testbench

```
module lab_func_TestBench();  
    logic a, b, c, d, y;  
    lab_func dut(a, b, c, d, y);  
    initial begin  
        a = 0; b = 0; c = 0; d = 0; #10;  
        d = 1; #10;  
        c = 1; d = 0; #10;  
        d = 1; #10;  
        b = 1; c = 0; d = 0; #10;  
        d = 1; #10;  
        c = 1; d = 0; #10;  
        d = 1; #10;  
        a = 1; b = 0; c = 0; d = 0; #10;  
        d = 1; #10;  
        c = 1; d = 0; #10;  
        d = 1; #10;  
        b = 1; c = 0; d = 0; #10;  
        d = 1; #10;  
        c = 1; d = 0; #10;  
        d = 1; #10;  
    end  
endmodule
```

\*\*\* “CircuitLab” online schematic editor is used to design and implement block designs in this paper:  
*Online circuit simulator & schematic editor*. CircuitLab. (n.d.). Retrieved February 24, 2022, from  
<https://www.circuitlab.com/>