Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

Digital System

ENCS234

Verilog project

****

Student's Name: Hazar Michael

ID: #1201838

Section: #4

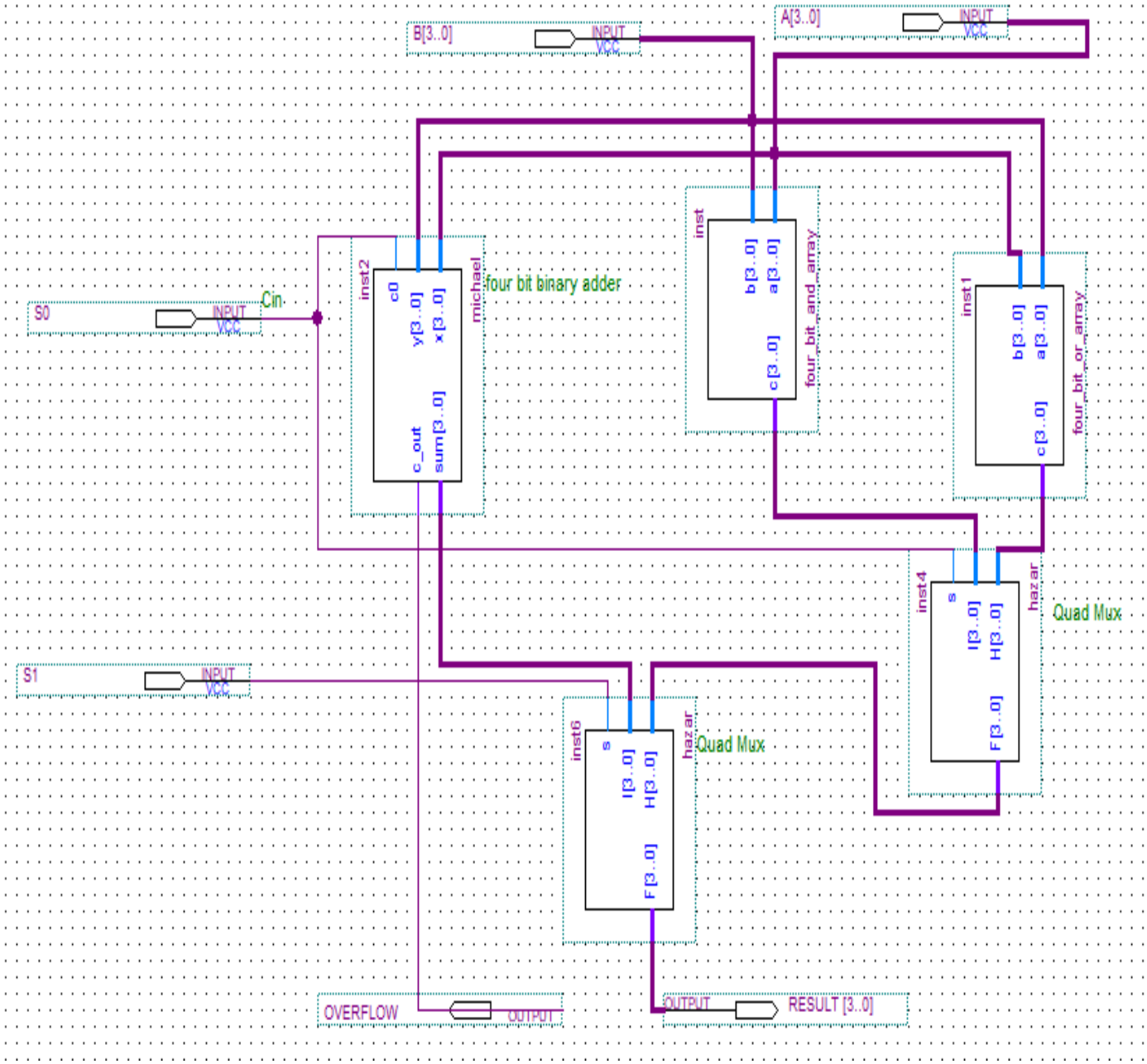
Instructor: Dr. Ismail Khater

*****

June 2022

Contents

Whole system design:



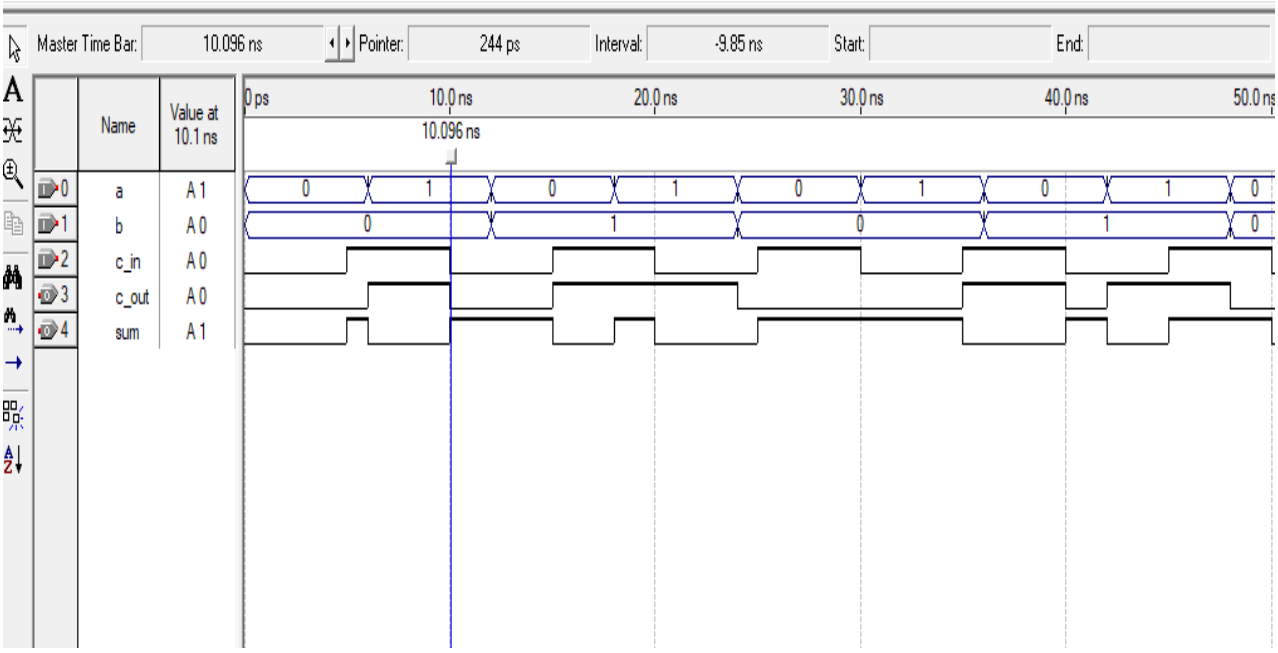
One-Bit binary adder:

- **Verilog code:**

one_bit_adder.v

```
1   //hazar michael 1201838
2
3   module one_bit_adder(a,b,c_in,sum,c_out);
4
5   input a,b,c_in;
6   output sum, c_out;
7   assign sum = a^b^c_in;
8   assign c_out = (a&&b)||(a&&c_in)||(b&&c_in);
9   endmodule
10
```

- **Stimulation**:



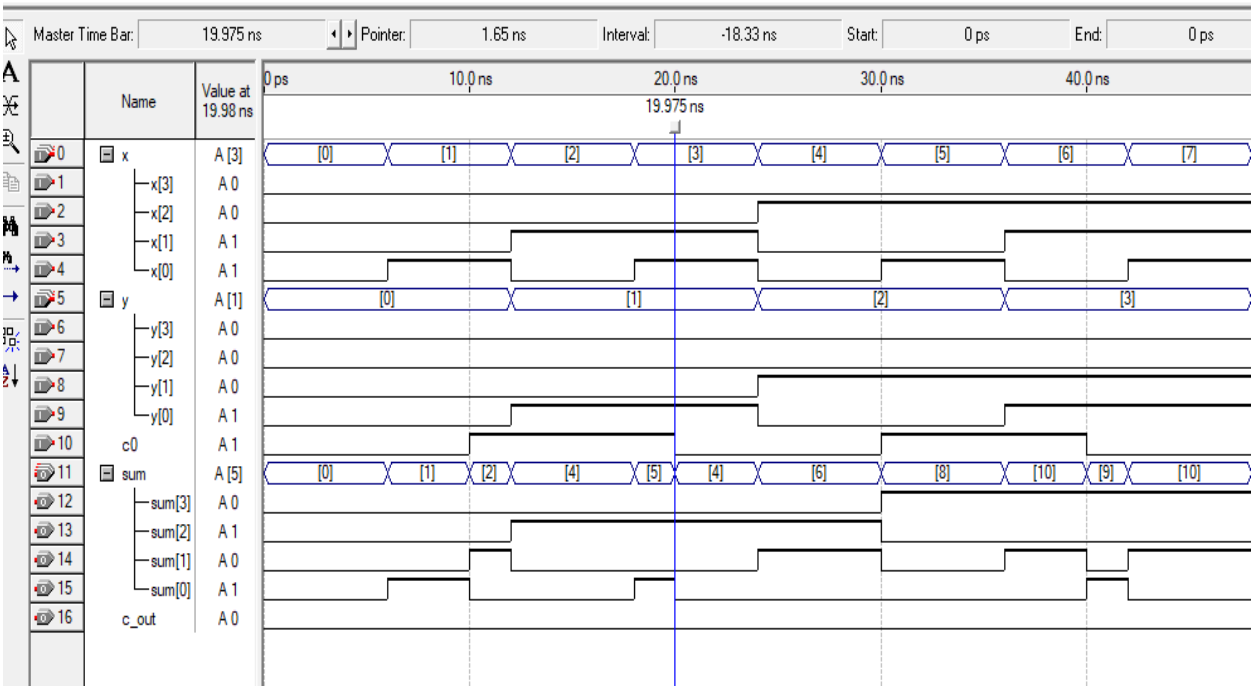
Four-Bit binary adder:

- **Verilog code:**

michael.v

```
1
2   //hazar michael 1201838
3   //four bit binary adder
4
5   module michael (x,y,c0,sum,c_out);
6   input c0;
7   input [3:0] x,y;
8   output [3:0] sum;
9   output c_out;
10  wire c1,c2,c3;
11
12  one_bit_adder G1 (x[0],y[0], c0, sum[0], c1);
13  one_bit_adder G2 (x[1],y[1], c1, sum[1], c2);
14  one_bit_adder G3(x[2],y[2], c2, sum[2], c3);
15  one_bit_adder G4(x[3],y[3], c3, sum[3], c4);
16
17
18  endmodule
```

- **Stimulation**:



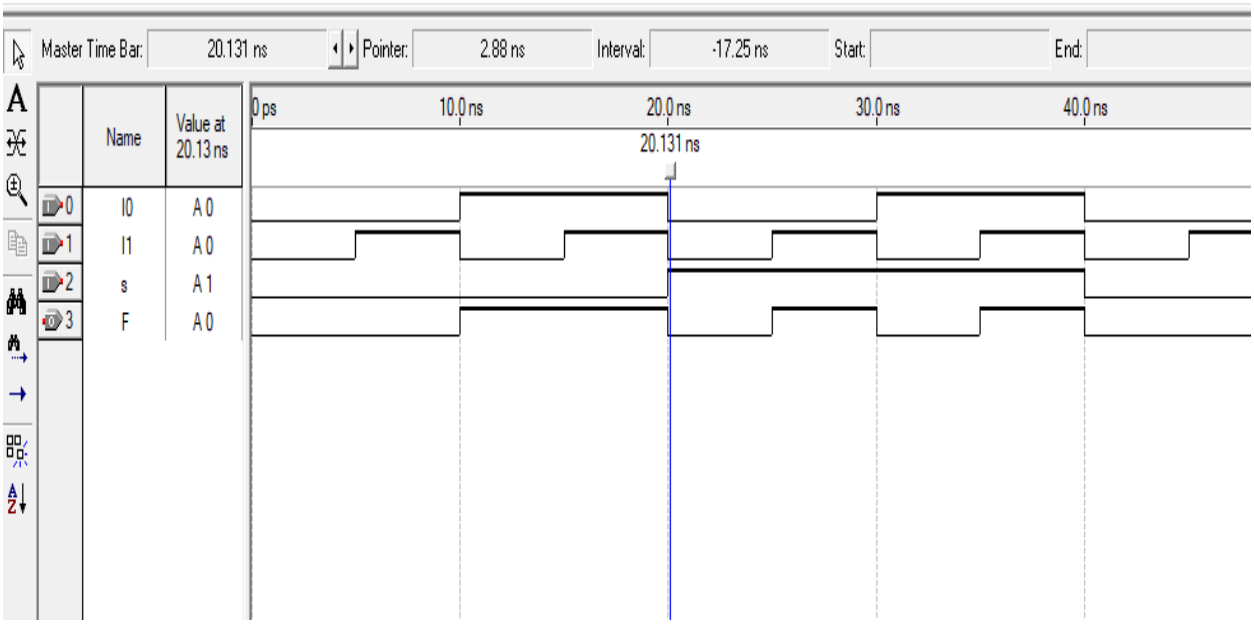
Mux 2x1:

- **Verilog code:**

mux_2x1.v

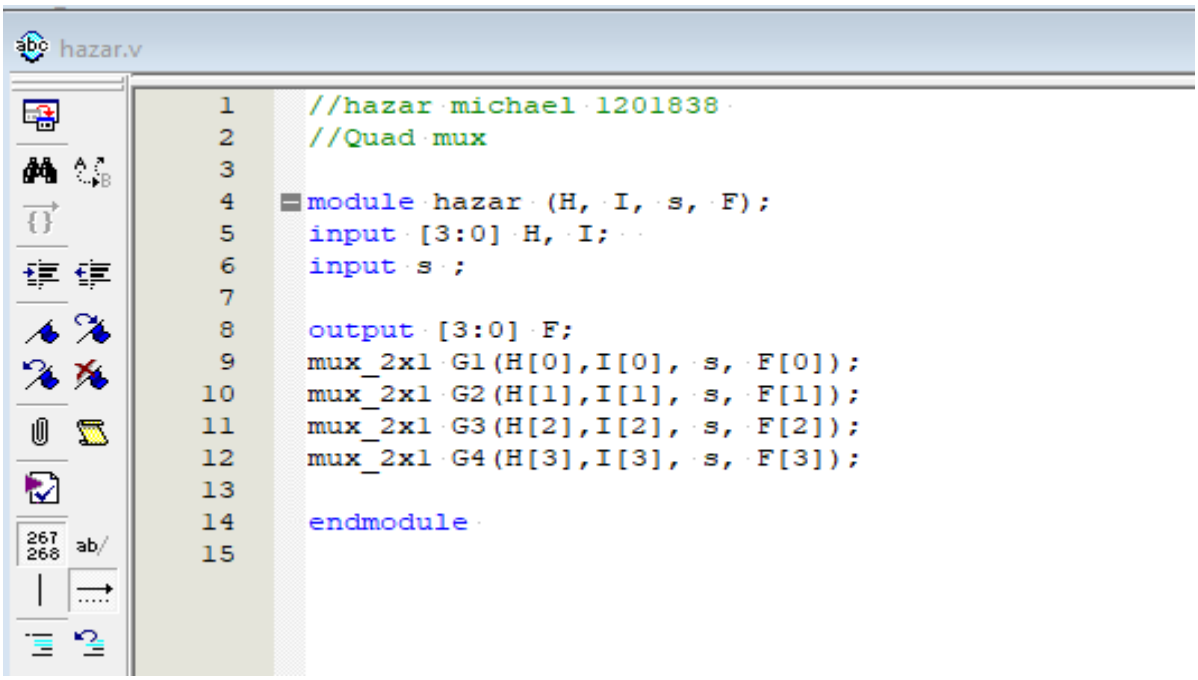
```
1  //hazar michael 1201838
2  module mux_2x1 (I0,I1,s,F);
3
4  input I0,I1,s;
5  output F;
6
7  assign F = (I0&&!s)||(I1&&s);
8  endmodule
```

- **Stimulation**:

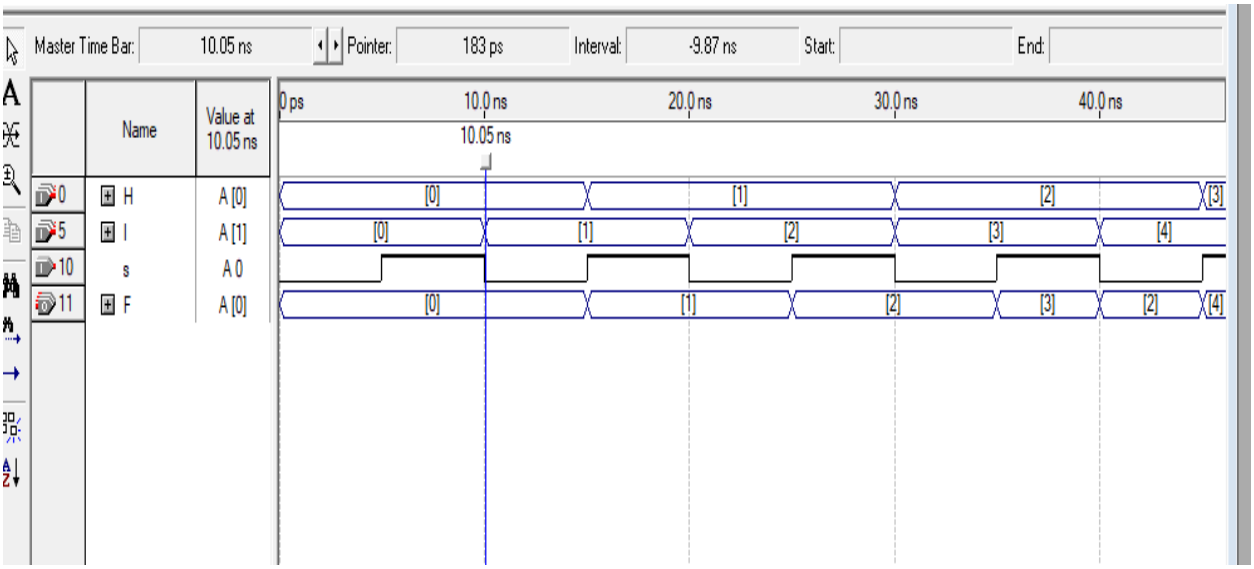


Quad mux 2x1:

- **Verilog code:**



- **Stimulation**:



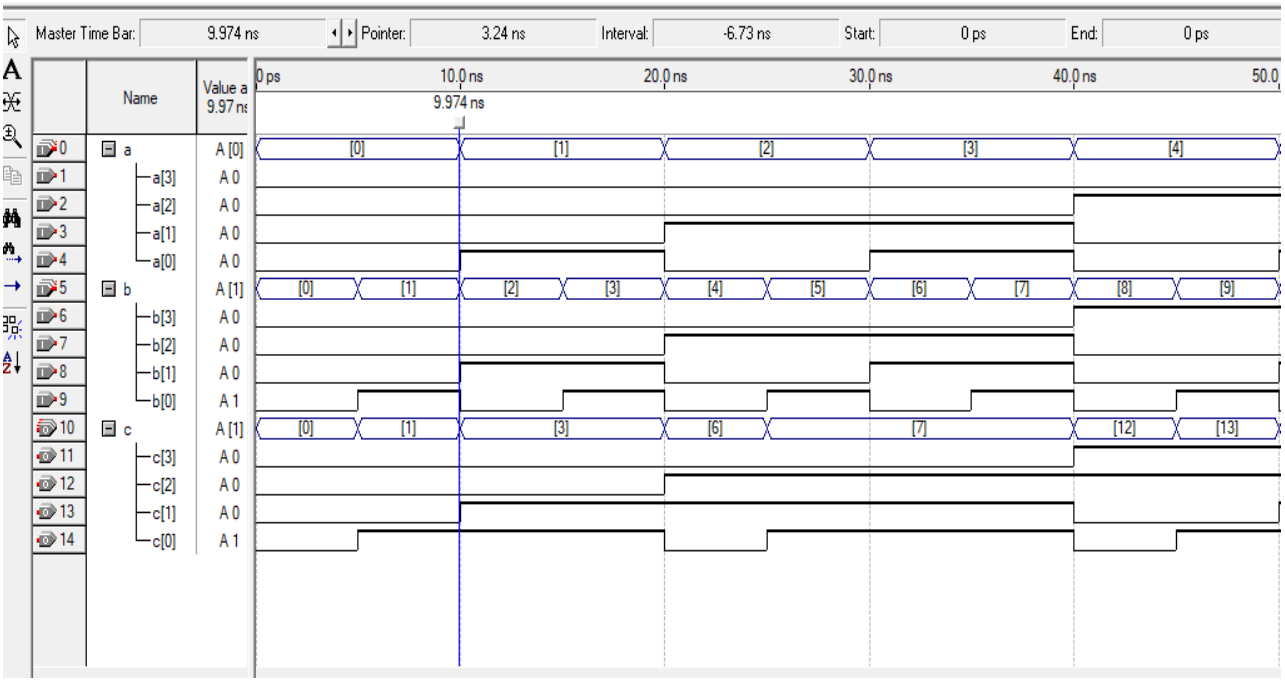
Four bit OR array:

- **Verilog code:**

four_bit_or_array.v

```
1   //hazar michael 1201838
2   module four_bit_or_array (a,b,c);
3   input [3:0]a,b;
4   output [3:0] c;
5
6   assign c[0] = a[0] || b[0];
7   assign c[1] = a[1] || b[1];
8   assign c[2] = a[2] || b[2];
9   assign c[3] = a[3] || b[3];
10
11  endmodule
12
```

- **Stimulation**:



Four bit AND array:

- **Verilog code:**

four_bit_and_array.v

```
1   //hazar michael 1201838
2   module four_bit_and_array (a,b,c);
3   input [3:0]a,b;
4   output [3:0] c;
5
6   assign c[0] = a[0] && b[0];
7   assign c[1] = a[1] && b[1];
8   assign c[2] = a[2] && b[2];
9   assign c[3] = a[3] && b[3];
10
11  endmodule
12
```

- **Stimulation**: