

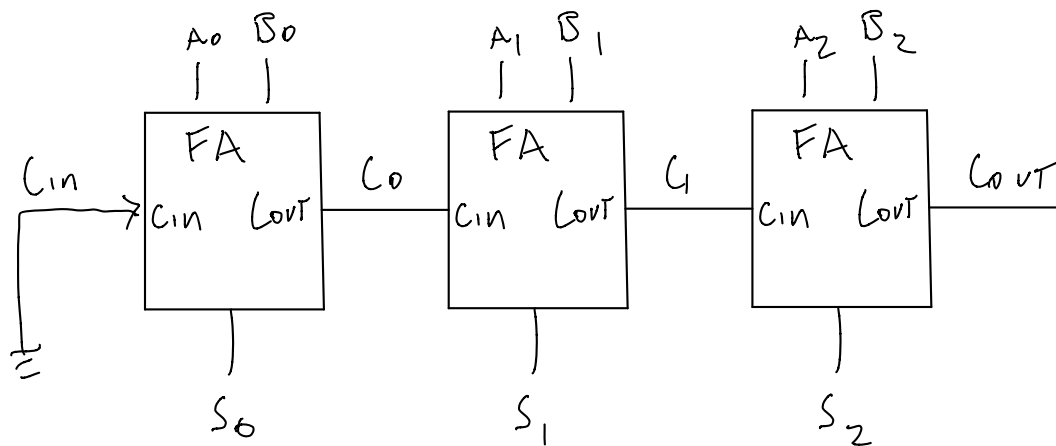
Class:	CPE-300L	Semester:	Spring 2020
Points		Document author:	Brysen Kokubun
		Author's email:	Brysenkokubun96@gmail.com
		Document topic:	Prelab 3
Instructor's comments:			

Introduction / Theory of operation

- For lab experiment #3 we will be getting familiar with ModelSim and its software for testbenches. We will also be reviewing and designing combinational circuits. Some of the circuits being designed are Ripple adder, ALU, implementing 7-seg on DE2 board, and Comparator.

Prelab main content

1. 3-bit ripple carry adder



2. Decoder of

a	b	y ₁	y ₂	y ₃	y ₄
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

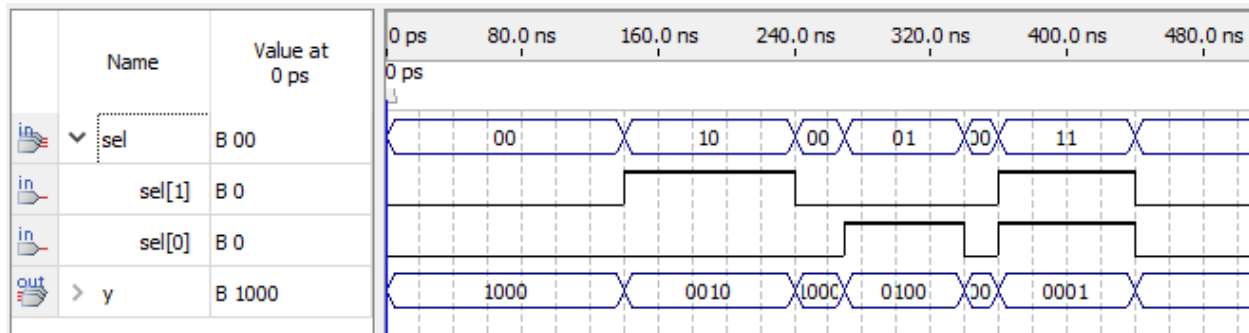
- Code

```

1  module decoder(sel, y);
2      input[1:0] sel;
3      reg[3:0] y;
4      output [3:0] y;
5      always@(*)
6      case(sel)
7          2'b00: y = 4'b1000;
8          2'b01: y = 4'b0100;
9          2'b10: y = 4'b0010;
10         2'b11: y = 4'b0001;
11     endcase
12 endmodule
13

```

- Waveform

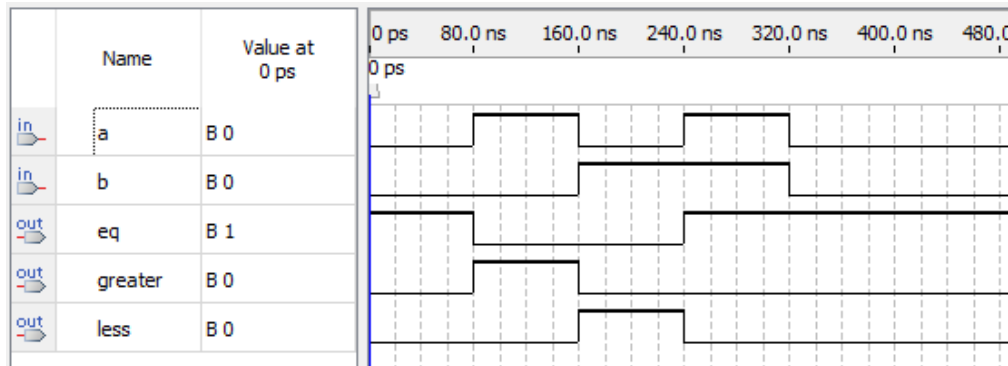


3. Magnitude Comparator

- Code

```
1 module Compare(a,b,less,eq,greater);
2     input a, b;
3     output less, eq, greater;
4     wire x1, x2;
5     not N1(x1,a);
6     not N2(x2,b);
7     and A1(less, x1, b);
8     and A2(greater, x2, a);
9     xnor F1(eq, a, b);
10 endmodule
```

- Waveform



4. Completed Altera ModelSim Tutorial

The screenshot displays the Altera ModelSim environment. On the left, a library pane lists various components like '220model', 'altera', and 'arriagx'. The main window shows a Verilog code editor with the following code:

```
1 module serial(A, B, start, resetn, clock, sum);
2     input [7:0] A, B;
3     input resetn, start, clock;
4     output [8:0] LEDR;
5
6     // Registers
7     wire [7:0] A_reg, B_reg;
8     wire [8:0] sum;
9     reg cin;
10
11     // Wires
12     wire reset, enable, load;
13     wire bit_sum, bit_carry;
14
15     // Control FSM
16     FSM my_control(start, clock, resetn, reset, enable, load);
17
18     // Datapath
19     shift_reg reg_A( clock, 1'b0, A, 1'b0, enable, load, A_reg);
20     shift_reg reg_B( clock, 1'b0, B, 1'b0, enable, load, B_reg);
21
22     // a full adder
23     assign bit_carry, bit_sum = A_reg[0] + B_reg[0] + cin;
24
25     always @(posedge clock)
26     begin
27         if (enable)
28             if (reset)
29                 cin <= 1'b0;
30             else
31                 cin <= bit_carry;
32         end
33     end
34
35     shift_reg reg_sum( clock, reset, 9'd0, bit_sum, enable, 1'b0, sum);
36     defparam reg_sum.n = 9;
37 endmodule
```

On the right, a 'Create Project' dialog box is open, showing the project name 'serial' and the project location 'C:/altera/13.1'. The 'Copy Library Mappings' checkbox is checked.