

EE 3755

Homework 2

Due: TBA

Estimated time to solve:

Prob.0	5 mins.
Prob.0-1	2 mins.
Prob.1	10 mins.
Prob.2	50 mins.
Prob.3	10 mins.
Total:	77 mins.

How to submit? // Hard copy at the class room.
Please write down actual time you spend to do the each
problem.

Prob. 0 What kind of hardware is this?(module WhatIsThis1)
(you may find it on lecture note) (5 mins.)

```
module WhatIsThis1(x,control,i0,i1,i2,i3);
  input [1:0] control;
  input [7:0] i0, i1, i2, i3;
  output      x;
  reg [7:0]   x;

  always @( control or i0 or i1 or i2 or i3 )
    begin

      case ( control )
        0: x = i0;
        1: x = i1;
        2: x = i2;
        3: x = i3;
      endcase

    end
end
```

```
endmodule
```

Prob.0-1 What kind of hardware is this?(module WhatIsThis2)

```
module WhatIsThis2(x,a,b,c,d,control);
  input [31:0] a, b, c, d;
  input [1:0] control;
  output [31:0] x;

  assign      x =
              control == 0 ? a :
              control == 1 ? b :
              control == 2 ? c : d;
endmodule
```

```
endmodule
```

Prob. 1 Without running the simulator, answer this question.(you may check later to verify your solution).

What will be the output of this program? (10 mins.)

```
module behavioral(x);
  output x;

  reg [7:0] x;

  initial

  begin

    x = 1;
    $display("Hello, x=%d, t=%t",x,$time);
    #15;

    x = 2;
    $display("Hello, x=%d, t=%t",x,$time);
    #20;

    x = 3;
    $display("Hello, x=%d, t=%t",x,$time);
  end
endmodule
```

```

        #10;

    end

// Initial block B
initial
    begin
        #5;

        x = 10;
        $display("Hello, x=%d, t=%t",x,$time);
        #20;

        x = 20;
        $display("Hello, x=%d, t=%t",x,$time);
        #5;

        x = 30;
        $display("Hello, x=%d, t=%t",x,$time);
        #10;

    end

endmodule

```

Prob. 2

Modify or rewrite the population counter to do the job faster.

The original population counter produces results after 34 clock cycles.

Think about something to do that with less or a lot less clock cycles. (50mins)

(Just writing down idea will get some partial credit)

Hint : Think about parallelism.

(More always blocks.)

```

## Original population counter.
module pop(p,a,clk);
    input [31:0] a;
    input        clk;
    output       p;

    reg [5:0]    p;
    reg [31:0]   acopy;
    reg [5:0]    pcopy;

```

```
initial acopy = 0;
initial pcopy = 0;
always @( posedge clk )
    begin

        if( acopy == 0 )
            begin
                p = pcopy;
                pcopy = 0;
                acopy = a;
            end
        else
            begin
                pcopy = pcopy + acopy[0];
                acopy = acopy >> 1;
            end

    end

endmodule
```

Prob. 3 Write a verilog program for the Prob.5 of verilog Hw1.(magnitude comparator slice). // the figure will be posted after collecting the Hw. // (10 mins.)