

**EE 4301 – Digital Design with Programmable Logic – Summer Session, 2008**  
**Midterm Exam (8:00 – 9:55 AM)**

Closed book, closed notes, no calculators, no electronic devices.

One 8.5"x11" information sheet is allowed.

NOTE: Be sure to clearly show how you obtained your answer to each question!

1. The following two operands are input to an 8-bit adder:

$$A = 11001010$$

$$B = 01110111$$

(a) (6 points) Assume that the carry in to the least-significant bit position is:  $C_0 = 0$ . Calculate the values of the following logic functions for  $i = 0, 1, \dots, 7$ :

$$P_i = A_i + B_i, \quad G_i = A_i B_i, \quad C_{i+1} = G_i + P_i C_i, \quad P_{sum_i} = A_i \oplus B_i \quad \text{and} \quad S_i = P_{sum_i} \oplus C_i.$$

Check your results by considering A and B to be unsigned numbers and performing a decimal calculation of the addition operation.

(b) (6 points) Assume that the carry in to the least-significant bit position is:  $C_0 = 1$ . Calculate the values of the following logic functions for  $i = 0, 1, \dots, 7$ :

$$P_i = A_i + B_i, \quad G_i = A_i B_i, \quad C_{i+1} = G_i + P_i C_i, \quad P_{sum_i} = A_i \oplus B_i \quad \text{and} \quad S_i = P_{sum_i} \oplus C_i.$$

Check your results by considering A and B to be unsigned numbers and performing a decimal calculation of the addition operation.

(c) (8 points) Using the carry look-ahead technique, derive the general equation for  $C_6$  as a function of  $P_i$ ,  $G_i$  and  $C_0$ . (Note: Do not use group propagate or group generate functions.) Then, evaluate this expression for the specific A and B operand values given above and for both possible values of  $C_0$ . Do you obtain the same values you obtained for  $C_6$  in parts (a) and (b)?

2. A combinational logic 10:4 encoder has a 10-bit input  $raw[9:0]$  and a 4-bit output coded[3:0]. If bit position  $i$  of  $raw$  is 1 and all the other bit positions are 0, then the coded output will be the unsigned binary code for  $i+3$ . (Any other input combinations are treated as don't-cares.) For example, if the input at  $raw$  is 0010000000, then the output at  $coded$  will be 1010.

(a) (6 points) Draw a minimized logic diagram using only OR gates to implement this function. (The OR gates may have as many inputs as needed.)

(b) (7 points) Write a complete behavioral Verilog description for this module using `always` and `if-else` statements.

(c) (7 points) Write a complete behavioral Verilog description for this module using `always` and `case` statements.

3. (20 points) A file called question.v contains the following:

```
module quad(a, b, c, d, s, t);

input  a, b, c, d;
output s, t;

assign s = a^b^c^d;
assign t = (a&b) | (a&c) | (a&d) | (b&c) | (b&d) | (c&d);

endmodule

module ripple(p, q, r, f);

input  [2:0] p, q, r;
output [3:0] f;

wire   w1, w2, w3, w4, w5, w6;

nand(w1, p[0], p[1], p[2]);
nand(w2, q[0], q[1], q[2]);
nand(w3, r[0], r[1], r[2]);
nand(w4, w1, w2, w3);

quad q0(p[0], q[0], r[0], w4, f[0], w5);
quad q1(p[1], q[1], r[1], w5, f[1], w6);
quad q2(p[2], q[2], r[2], w6, f[2], f[3]);

endmodule
```

Another file called tbquestion.v contains the following:

```
module tbquestion;

reg  [2:0] x, y, z;
wire [3:0] g;
reg  [8:0] h;
integer i;

ripple instance1(x, y, z, g);

initial begin
    x = 3'b110;
    y = 3'b101;
    for (i = 1; i < 8; i = i + 3) begin
        z = i;
        #10 h = x * y * z;
        $display($time, " %d %d %d %d %d", x, y, z, g, h);
    end
end

endmodule
```

If the files `question.v` and `tbquestion.v` are simulated using Verilog, give the output that will be produced by the `$display` statement. Be sure to clearly show how you obtained your answer!

4. The following Verilog code describes a finite state machine:

```
module fsm (z_out, x_in, clock, reset_b);
    output      z_out;
    input       x_in;
    input       clock, reset_b;
    reg [1:0]   state, next_state;
    reg         z_out;
    parameter   S0 = 0,
                S1 = 1,
                S2 = 2,
                S3 = 3;

    always @ (posedge clock or negedge reset_b)
        if (reset_b == 0) state <= S0;
        else state <= next_state;

    always @ (state or x_in) begin
        if ((state == S2) && (x_in == 1'b0))
            z_out = 1'b0;
        else
            z_out = 1'b1;
        end

    always @ (state or x_in) begin
        case (state)
            S0: begin if (x_in == 0) next_state = S2;
                    else next_state = S1; end
            S1: begin next_state = S2; end
            S2: begin if (x_in == 0) next_state = S1;
                    else next_state = S3; end
            S3: begin next_state = S0; end
        endcase
    end
endmodule
```

(a) (4 points) Is this a Moore-type or a Mealy-type FSM? Why?

(b) (8 points) Draw the corresponding state transition diagram. (Use the abstract state names in your diagram.)

(c) (8 points) Construct the corresponding state transition table. (Use the specified state encoding to correspond to the Q outputs of a pair of flip-flops.)

5. (a) (4 points) Draw the diagram for a 4-bit by 2-bit unsigned multiplier using the same style as for the 3-bit by 3-bit design discussed in class, i.e. as an interconnection of the minimum number of full adders and half adders.

(b) (4 points) If  $X = 13$  and  $Y = 3$  are applied at the inputs of the multiplier, give the values at the carry output of each full adder and half adder and the value at each output bit position of the multiplier. (Indicate these values on your diagram of part (a)).

(c) (4 points) Consider the design of a functional block to compute  $(X*Y) + (2*Z)$ , where  $X$  is a 4-bit unsigned number,  $Y$  is a 2-bit unsigned number and  $Z$  is a 3-bit unsigned number. What is the decimal value of the largest result that could be produced? What is the least number of bits required to represent this value?

(d) (4 points) Use the 4-bit by 2-bit unsigned multiplier design of part (a) together with an optimized ripple carry adder to create the functional block of part (c). (The ripple carry adder should contain a minimum number of half adders and full adders, using full adders only where they are necessary.) Draw a diagram which clearly shows how the individual components are connected together to form the complete design.

(e) (4 points) If  $X = 13$ ,  $Y = 3$  and  $Z = 6$  are applied at the inputs of the functional block, give the values at the carry output of each full adder and half adder in the ripple carry adder and the value at each output bit position of the functional block. (Indicate these values on your diagram of part (d)).