Sample Solutions to Exercise 2

Chapter 8

4. alu_version_2. (See Example 8.33 on pages 336-337 for a similar case.)

5. circuit_1 is more efficient in simulation since it has fewer inputs in the event control expression. Both are synthesis friendly since each has all the inputs in the event control expression.

7. the condition k >= 0 in the for statement is always true since k is a 1-bit register. Therefore if a is equal to b, the for loop will run forever.

9. It will be ignored.

25. The description should be synthesizable to Synopsys.

Chapter 9

3. Should add reset for state, and default for the case statement. If the width of each s0, s1, s2, and state (whose data type is changed to register) is specified, the size of hardware register might be reduced.

10. module p10 (data, count, clk, reset);
    input data, clk, reset;
    output [7:0] count;
    reg [7:0] count, temp_count, max_count, valid_bits; i;
    reg [63:0] bit_string;
    always @(posedge clk or posedge reset)
        if (reset)
            begin
                count = 8'b0;
                bit_string = 64'bx;
                valid_bits = 0;
            end
        else
            begin
                bit_string = {bit_string [62:0], data};
                temp_count = 1;
                max_count = 1;
                if valid_bits < 64
                    valid_bits = valid_bits + 1;
                if valid_bits > 1
                    begin
                        for (i = 0; i < valid_bits - 1; i = i + 1)
begin
    if (bit_string[i] == bit_string[i+1])
        begin
            temp_count = temp_count + 1;
            if (temp_count > max_count)
                max_count = temp_count;
        end
    else
        temp_count = 1;
    end
end

endmodule

12. module p12 (data, result, clk);
    input [15:0] data;
    input clk;
    output [15:0] result;
    reg [15:0] result;
    reg [4:0] i;
    always @ (posedge clk)
        begin : shift_loop
            result = data;
            for (i = 0; i < 15; i = i + 1)
                if (result[15] == 0)
                    result = result << 1;
                else disable shift_loop;
        end
endmodule

22. Should use non-blocking assignments.

24. The two modules have the same function.
Chapter 10

6. Static loop without internal timing controls.

8. Both have the same dataflow graph shown below.

![Dataflow Graph]

9. Latches will be created because of the incompletely specified case statement. alu_reg will not respond to any change to data_a or data_b when opcode remains unchanged. Note that there is a syntax error in which clk declared as input but does not appear in the port list.

10. The if condition is always false because the rightmost two bits of \{data_reg, serial_input\} are either 00 or 11 (which makes \{data_reg, serial_input\} not equal to pattern whose rightmost two bits are 10). Therefore pattern_match is always 0.