UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Digital Logic Design, ECE 367 / Digital Systems I, ECE 894

Spring 1399-1400

Homework 6

RTL Combinational Parts and Iterative Structures

Name: **Date:**

Username:

1. **A)** Write Verilog code for a 1-bit magnitude comparator with *a, b, l, g* inputs and *lt* and *gt* outputs. **B)** In a Verilog module wire eight of comparators of Part A to build an 8-bit comparator. Assign constants to the left-most *l* and *g* inputs as appropriate.

2. A priority-encoder circuit has four source inputs (S3, S2, S1, and S0), and an active-low enable input (EI), a two-bit prioritized N output, an active-low enable output (EO) and an active-low source output (AS). The source inputs are active-high. The EI input makes all outputs inactive when it is 1. The priority of the sources is S3 to S0, with S3 having the highest, and S0 having the lowest priority. When enabled, the number of the active source with the highest priority is encoded and appears on the N output. This means that if S2 and S0 are 1, N becomes 2. On the other hand, if only S0 is 1, N becomes 0. The AS output becomes 0 when at least one input source is active. This distinguishes the case that only S0 is active from the case that all sources are inactive. The EO output becomes 0 (enabled) when no input source is active. This output is used for cascading priority encoders. The priority-encoder discussed here can be built with a circuit for prioritizing the input sources followed by a 4-to-2 encoder, and some discrete logic gates. Show complete gate-level structure of a 4-bit priority encoder circuit.

3. Show SystemVerilog description (using **assign** statements, one for every output) for the priority-encoder of Problem 2.

4. Use the priority-encoder circuit of Problem 2 to build a 16-bit priority-encoder circuit. Show schematic diagram of this circuit using blocks representing the 4-bit circuit of Problem 2. Use a **generate** statement for instantiating four 4-input priority-encoders. You will need extra gates for this cascading.

5. Show the design of a 2-bit combinational multiplier with A[1:0], B[1:0] inputs and M[3:0] outputs. Using Karnaugh maps. Show the complete circuit diagram. Show how this 2-bit multiplier and adders can be used to build a 4-bit multiplier.

6. Describe the multipliers of Problem 5 in SystemVerilog.

7. A circuit for counting overlap of complementing double-bits OC2B of a 64-bit input is to be built. Use an iterative structure for implementation of this circuit. Use basic gates, full-adders and other structures discussed in class.

8. Use **generate** statements to implement the circuit of Part 7.