

Student ID: \_\_\_\_\_

# CS 151 Quiz 7

Name : \_\_\_\_\_ , \_\_\_\_\_  
(Last Name) (First Name)

Student ID : \_\_\_\_\_

Signature : \_\_\_\_\_

## **Instructions:**

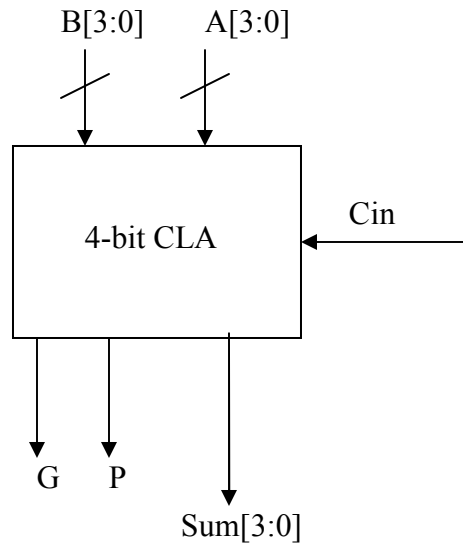
1. Please verify that your paper contains **7 pages** including this cover.
2. Write down your Student-Id on the top of each page of this quiz.
3. This exam is **closed book**. No notes or other materials are permitted.
4. Total credits of this quiz are **45 points**.
5. To receive credit you must show your work clearly.
6. **No re-grades will be entertained if you use a pencil.**
7. Calculators are **NOT** allowed.

Student ID: \_\_\_\_\_

**Q1 [Carry Look-Ahead adder]**

**[30 points]**

1. Design a 8-bit hierarchical carry look-ahead following steps (1.a) and (1.b), using 4-bit carry look-ahead adders shown below. **[15 points]**  
**In this Question use one level of hierarchy.**



- a) Draw the interface between the 4-bit CLAs and the CLA Logic that should be added to implement the 8-bit adder. (At this stage you do not need to implement the circuit inside the blocks.) **[5 points]**

Student ID: \_\_\_\_\_

**b) Write the equations for the outputs of the CLA Logic block. [10 points]**

Student ID: \_\_\_\_\_

2. In this question using the 8-bit carry look-ahead adder you designed in the previous problem, you should design a 16-bit carry look-ahead adder. **[15 points]**
  - a) Draw the block structure of the 16-bit adder and show the interface signals between the blocks. (At this stage you do not need to implement inside the blocks.) **[5 points]**

Student ID: \_\_\_\_\_

- b)** Write the equations for the outputs of the CLA Logic blocks in each level.  
**[10 points]**

Student ID: \_\_\_\_\_

**Q2 [Add and Shift Multiplier]**

**[15 points]**

We want to multiply two 4-bit unsigned binary numbers using add and shift method with a data path shown below. The **multiplcand** is equal to 1010 and the **multiplier** is equal to 0110. The table on the next page shows the algorithmic steps in the “Action” column and contents of registers Multiplicand, Multiplier and Running Sum. We have filled on the first 5 steps. Complete the table for executing the multiplication until it is done and show content of the registers after each step of adds and shifts.

