Nov. 28, 2010

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Major Exam I

 (Open Book Exam)

First Semester (101)

Time: 1:00-3:30 PM

Student Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

|  |  |  |
| --- | --- | --- |
| **Question** | **Max Points** | **Score** |
| **Q1** | **15** |  |
| **Q2** | **15** |  |
| **Q3** | **10** |  |
| **Q4** | **20** |  |
| **Q5** | **20** |  |
| **Q6** | **20** |  |
| **Total** | **100** |  |

#  **[15 Points]**

# **(Q1)** Draw the ROBDD for the function F=a⊕b⊕c⊕d, with the variable ordering {a, b, c, d}. How can we easily obtain the ROBDD for $\overbar{F}$ from the ROBDD for F? Don’t draw the ROBDD for $\overbar{F}$, just explain.

#

#

#  **[15 Points]**

# **(Q2)** Write an algorithm, called **ROBDD**, that receives a function F and a variable ordering and constructs and ROBDD for the input function. Explain clearly the terminal cases and the structure of the tables you will use in your algorithm.

#  **[10 Points]**

# **(Q3)** Consider the function. Using recursive paradigm, determine if the function F is **tautology** or not. You need to choose the right variable for expansion to minimize computations.

#

#  **[20 Points]**

# **(Q4)** Consider the two Boolean functions  and  given below:

#

#

# Draw the **ITE DAG** for the function using the variable order {A, B, C, D}. Show all the details of your solution using ITE procedure including the resulting unique table and computed table.

##

#

#  **[20 Points]**

# **(Q5)** Consider the function

## Compute the **complement** of the function using the recursive complementation procedure outlined in section 7.3.4.

## **C**ompute all the **prime implicants** of the function using the method outlined in section 7.3.4.

#

#  **[20 Points]**

# **(Q6)** Consider the following given matrix representing a covering problem:



## Find a **minimum cover** using **EXACT\_COVER** procedure. Show all the details of the algorithm. Assume the following order in branching selection when needed: C1, C2, C3, C4, C5, C6, C7, C8. Propose two ideas that can be employed to make the **EXACT\_COVER** procedure execute efficiently in general.

#