Oct. 20, 2011

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Major Exam I

(Open Book Exam)

First Semester (111)

Time: 2:00-4:30 PM

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

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

|  |  |  |
| --- | --- | --- |
| **Question** | **Max Points** | **Score** |
| **Q1** | **10** |  |
| **Q2** | **10** |  |
| **Q3** | **10** |  |
| **Q4** | **20** |  |
| **Q5** | **30** |  |
| **Q6** | **20** |  |
| **Total** | **100** |  |

# **[10 Points]**

# **(Q1)**

## Represent the cover , using positional cubical notation.

## Compute the **complement** of the cover *F* using **sharp** operation.

# **[10 Points]**

# **(Q2)** Consider the function and the set of implementations given below.

# Assume that the area and delay of a gate are directly related to the number of its inputs. Compute the **area** and **delay** cost for each implementation and determine the **Pareto** **optimal** **points**.

|  |  |  |
| --- | --- | --- |
| Implementation | Area | Delay |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 

# **[10 Points]**

# **(Q3)** Consider the following 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.

## 

# 

# **[30 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.

# 