Jan. 8, 2012

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Final Exam

(Open Book Exam)

First Semester (111)

Time: 7:00-10:00 PM

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

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

|  |  |  |
| --- | --- | --- |
| **Question** | **Max Points** | **Score** |
| **Q1** | **20** |  |
| **Q2** | **22** |  |
| **Q3** | **21** |  |
| **Q4** | **12** |  |
| **Q5** | **25** |  |
| **Total** | **100** |  |

#

**[20 Points]**

#  **(Q1)** Consider a technology library containing the following cells:

#

|  |  |  |
| --- | --- | --- |
| **Cell** | **Area Cost** | **Gate** |
| INV(x1) = x1’ | 1 |  |
| NAND2(x1, x2) = (x1 x2)’ | 2 |  |
| NAND3(x1, x2, x3) = (x1 x2 x3)’ | 3 |  |
| NOR2(x1, x2) = (x1 + x2)’ | 2 |  |
| NOR3(x1, x2, x3) = (x1 + x2 + x3)’ | 3 |  |
| AOI21(x1, x2, x3) = ((x1 x2) + x3)’ | 3 |  |
| AOI22(x1, x2, x3, x4) = ((x1 x2) + (x3 x4))’ | 4 |  |
| OAI21(x1, x2, x3) = ((x1+x2) x3)’ | 3 |  |
| OAI22(x1, x2, x3, x4) = ((x1+x2) (x3+x4))’ | 4 |  |

#

## Consider the circuit given below with inputs *{a, b, c, d, e, f, g, h}* and output {*Z}*. Using the dynamic programming approach and **Structural Matching**, **map** the circuit using the given library into the **minimum area** cost solution.

##



## Can you obtain a better mapping than the one obtained in (i). If the answer is yes, show the better solution and explain how it is obtained.

##  Assuming **Boolean Matching**, determine the number of ROBDD’s that need to be stored in the cell library for the following cell. Justify your answer.

Y = a b c d + a’ b’ c d + e f + e’ g

#  **[22 Points]**

#

# **(Q2)** Consider the incompletely-specified FSM that has 5 states, one input (X) and one outputs (Z), represented by the following state table:

|  |  |
| --- | --- |
| **Present State** | **Next State, Z** |
| **X=0** | **X=1** |
| S0 | S3, – | S0, – |
| S1 | S4, 0 | S0, – |
| S2 | S3, 0 | S1, – |
| S3 | S2, – | S2, – |
| S4 | S2, 1 | S1, – |

## Determine the incompatible states and the compatible states along with their implied pairs.

## Compute the maximal compatible classes along with their implied state pairs.

## Compute the prime compatibility classes along with their implied state pairs.

## Reduce the state table into the minimum number of states and show the reduced state table.

#  **[21 Points]**

# **(Q3)** Consider the given FSM which has 4 states, one input and one output, represented by the following state table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **Input** | **Present State** | **Next State** | **Output** |
| P1 | 0 | S1 | S2 | 0 |
| P2 | 1 | S1 | S2 | 0 |
| P3 | 0 | S2 | S2 | 0 |
| P4 | 1 | S2 | S3 | 0 |
| P5 | 0 | S3 | S4 | 0 |
| P6 | 1 | S3 | S3 | 0 |
| P7 | 0 | S4 | S4 | 0 |
| P8 | 1 | S4 | S1 | 1 |

## Assuming the following constraints: S3 covers S2, and that the code of S4 is covered by all other state codes, the state table can be reduced into the table given below. Using implicant merging and covering relations show step by step how you can obtain the reduced state stable given below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Present State** | **Next State** | **Output** |
| *–* | S1, S2 | S2 | 0 |
| 1 | S2, S3 | S3 | 0 |
| 1 | S4 | S1 | 1 |

## Compute all the seed dichotomies and construct their compatibility graph. Find a minimum cover for the seed dichotomies. Based on the found cover, derive an encoding satisfying the given constraints with minimal bit length.

**[12 Points]**

# **(Q4)** Consider the sequential circuit given below having 3 inputs {A, B, C} and one output {Z}. Assume that the delay of all given gates is 2 unit delays.

#

## Determine the critical path of this circuit and the maximum propagation delay.

## Using only the **Retiming** transformation, reduce the critical path of this circuit with the minimum number of flip-flops possible. Determine the maximum propagation delay after retiming.

**[25 Points]**

#  **(Q5)** Consider the network given below with inputs *{i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11}* and outputs *{o1, o2, o3}.* Assume that the delay of both the Adder and the Multiplier fit within one clock cycle and that the input values will be available to the circuit for only one clock cycle. Also assume that both addition and subtraction operations will be performed by the Adder.

#  a = i1 + i2; b = a – i3; c = i4 + i5; d = i7 \* i8; e = i9 + i10;

# f = d + e; g = i11 \* 7; o1= b \* 3; o2 = c + i6; o3= g \* f;

## Using **List Scheduling** algorithm LIST\_L, schedule the sequencing graph into the **minimum number of cycles** under the resource constraints of one Adder and one Multiplier. Show the details of the algorithm step by step and the resulting scheduled sequencing graph.

## Using **List Scheduling** for minimum resource usage algorithm LIST\_R, schedule the sequencing graph under the latency constraint of **5 clock** **cycles** minimizing the number of resources required. Show the details of the algorithm step by step and the resulting scheduled sequencing graph.

## Consider the scheduled sequencing graph below:



* 1. Show the life-time of all variables.
	2. Determine the minimum number of registers that are required to store all the variables. Show the mapping of variables to registers. Select a mapping that **minimizes the number of multiplexers and interconnect area** as much as possible.
	3. Draw the **data-path** implementing the scheduled sequencing graph based on the variable-register mapping that you obtained in (b).

##