Feb. 2, 2009

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Final Exam

(Open Book Exam)

First Semester (081)

Time: 7:00-10:00 PM

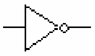
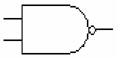
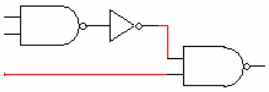
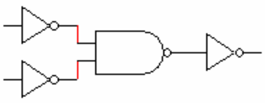
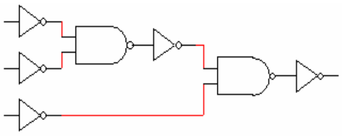
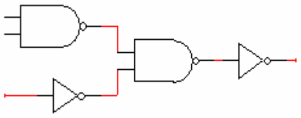
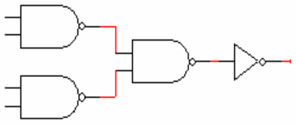
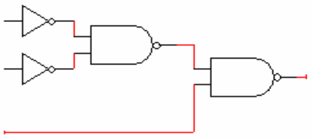
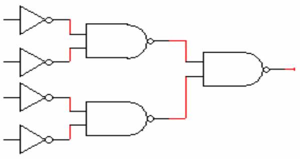
Student Name : ____

Student ID. : ____

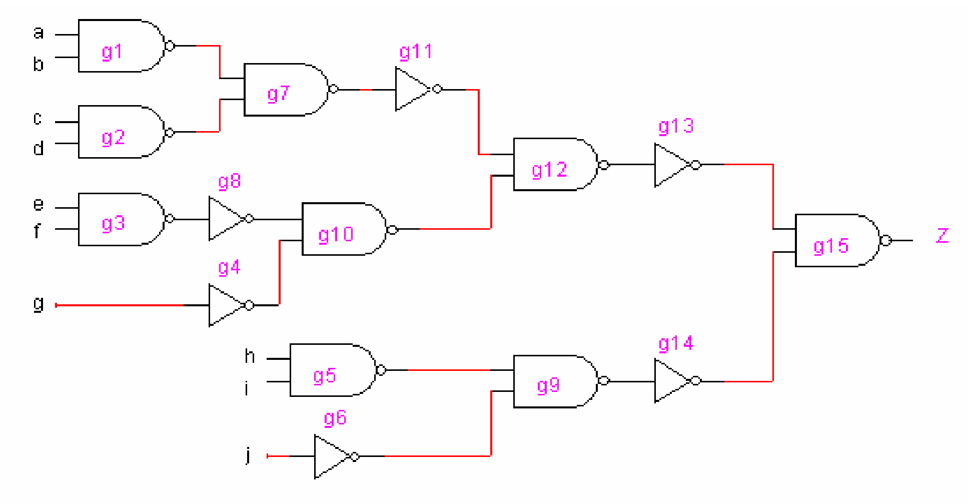
Question	Max Points	Score
Q1	20	
Q2	20	
Q3	25	
Q4	10	
Q5	25	
Total	100	

[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	

(i) Given the circuit below with inputs *{a, b, c, d, e, f, g, h, i, j}* and output {*Z*}. Using the dynamic programming approach, **map** the circuit using the given library into the **minimum area** cost solution.



(ii) 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 + a' b' c' + d e f

[20 Points]

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

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

(i) Determine the incompatible states and the compatible states along with their implied pairs.

(ii) Compute the maximal compatible classes along with their implied state pairs.

(iii) Reduce the state table into the minimum number of states and show the reduced state table.

[25 Points]

(Q3) Consider the incompletely-specified FSM that has 4 states, two inputs and one output, represented by the following state table:

Product	Input	Present State	Next State	Output
P1	00	S0	S0	0
P2	00	S1	S0	0
P3	00	S2	S0	0
P4	00	S3	S1	1
P5	01	S0	S0	0
P6	01	S1	S1	0
P7	01	S2	S0	0
P8	01	S3	S0	0
P9	10	S0	S2	0
P10	10	S1	S2	0
P11	10	S2	S2	0
P12	10	S3	S1	1
P13	11	S0	S3	1
P14	11	S1	S1	0
P15	11	S2	S3	1
P16	11	S3	S1	0

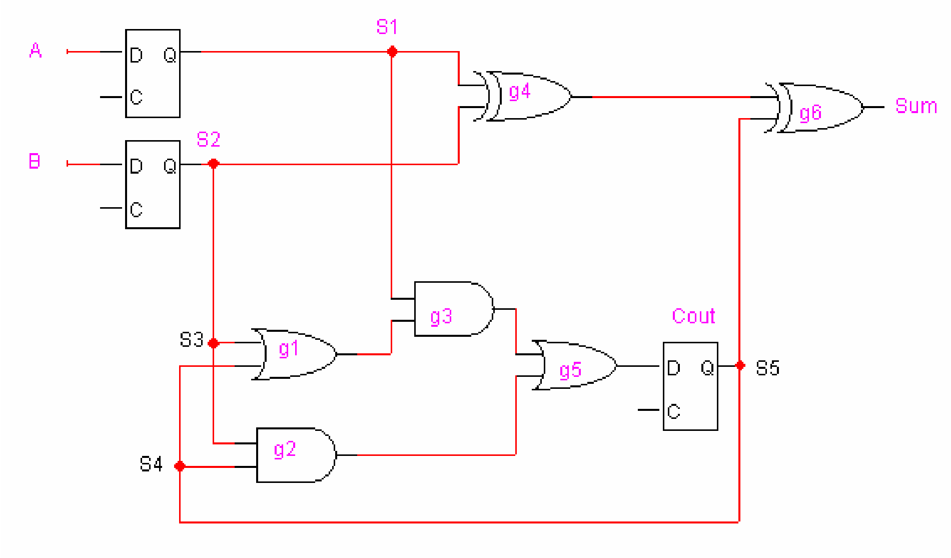
(i) Assuming the following constraints, S1 covers S0 and that the code of S2 is covered by all other state codes, the state table can be reduced into the table given below. Using implicant merging and covering relations show step by step how you can obtain the reduced state stable given below.

Input	Present State	Next State	Output
0–	S0, S1, S2, S3	S0	0
01	S1	S1	0
–0	S3	S1	1
11	S0, S2	S3	1
11	S1, S3	S1	0

(ii) Compute all the prime dichotomies. Find a state encoding satisfying the given constraints.

[10 Points]

(Q4) Consider the circuit given below representing a serial adder. Assume that the delay of a 2-input AND gate is 2 unit delays, the delay of a 2-input OR gate is 2 unit delays and the delay of a 2-input XOR gate is 4 unit delays.



(i) Determine the critical path of this circuit and the maximum propagation delay.

(ii) Using only the **Retiming** transformation, reduce the critical path of this circuit with the minimum number of flip-flops possible.

[25 Points]

(Q5) Consider the network given below with inputs *{a, b, c, d}* and output *{y}*:

[1]=b+c; [2]=a+b; [3]=b+d; [4]=a+[3];
[5]=[1]+[3]; [6]=[2]+[4]; y=[5]+[6];

Assume that the delay of an Adder fits within one clock cycle.

(i) Draw the **sequencing graph** for the above network.

(ii) Show the **ASAP** and **ALAP** scheduling of the sequencing graph. Compute the **mobility** of each operation.

(iii) Using **List Scheduling** algorithm LIST_L, schedule the sequencing graph into the **minimum number of cycles** under the resource constraints of two Adders. Show the details of the algorithm step by step and the resulting scheduled sequencing graph. Note that if two operations can be equally scheduled in a step, schedule the one with the least operation number.

(iv) Consider the scheduled sequencing graph based on **your solution in (iii)**. Assume that the input values will be available to the circuit for only one clock cycle.

a. Show the life-time of all intermediate variables.
b. Determine the minimum number of registers that are required to store all the temporary variables. Show the mapping of variables to registers. Select a mapping that **minimizes the number of multiplexers and interconnect area** as much as possible. Assume that operations [3], [4] and [5] are bound to the same Adder and the remaining operations are mapped to the other Adder.
c. Draw the **data-path** implementing the scheduled sequencing graph based on the variable-register mapping that you obtained in (b).