CSE 370, Spring 2008

(40pts) Problem 1: Let's warm up for the rest of the exam!

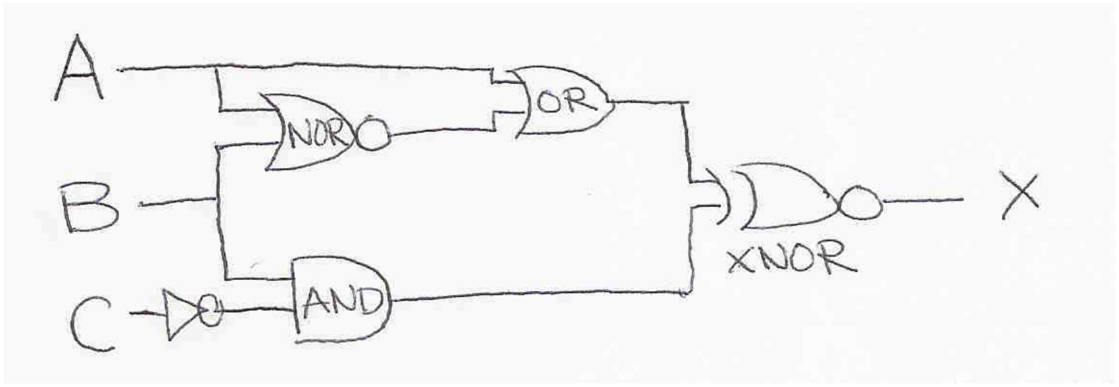
(a) [6pts] Convert $2B_{16}$ to a binary number.

(b) [6pts] Convert $2B_{16}$ to a decimal number.

(c) [6pts] Add 'negative 12' and 'positive 5' in 1's complement (leave your answer in 1's complement form).

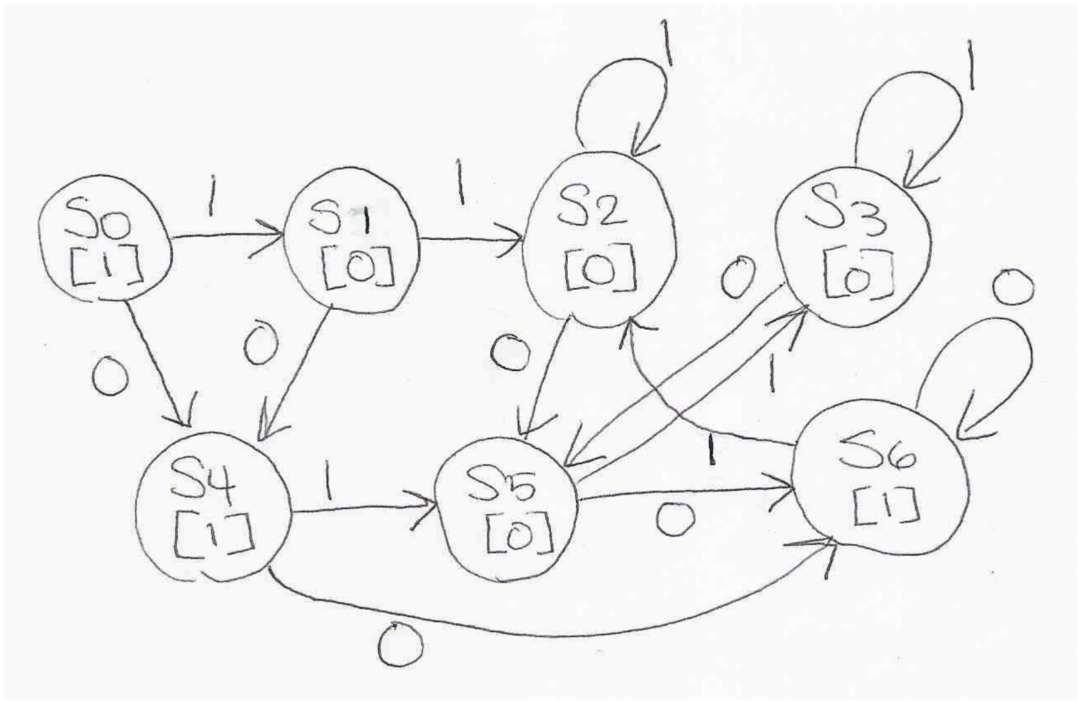
(d) [6pts] Add 'negative 12' and 'positive 5' in 2's complement (leave your answer in 2's complement form).

(e) [16pts] For the following multi-level logic circuit, convert it to a two-level logic circuit. Use SOP in canonical form (and no need to simplify beyond that). You can use a bar or a dash to note inversion. You can also have multi-input AND/OR gates. Hint: when you make the truth table, it helps to also put in some intermediate values.



(40pts) Problem 2: Implication chart

For the following state diagram, we are interested in minimizing this diagram.



(a) [5pts] Explain why you may <u>not</u> want to use row matching technique to find the most minimized state diagram.

(b) [10pts] Fill in the following state transition table (for states, simply use S_0, S_1, etc.)

Present State	Next State		Output
	input = 0	input = 1	

(c) [10pts] Fill in the implication chart below. Stop before starting to cross off the grid with states written in. At the end of this process, each grid should have either a cross or several states listed (but do not cross off any grids with states inside --- that's in part (d))

S_1						

(d) [10pts] List all grids that need to be crossed off based on the states written inside. Do this systematically (top to bottom, left to right) and explain your steps. For example:

1. Crossed off grid S2/S0 because S3/S0 already crossed off.
2. …

(e) [5pts] List all states that could be combined together for the most minimized state diagram (but no need to draw the diagram itself).

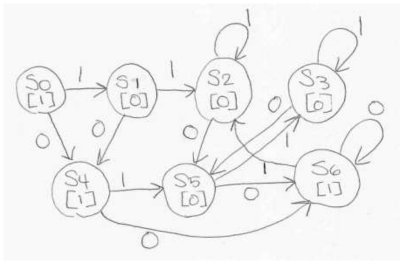
(40pts) Problem 3: Partitioning

For the same state diagram used for Problem 2 above, let's think of clever ways to partition the state diagram. Each sub-question makes different assumptions (for example, the assumptions made in (a) are not carried over to (b), etc).

(a) [10pts] Let's assume that we will be using chips that contain only 2 flip-flops each (but we have as many chips as you need). Assuming you use gray-code encoding of the states, how many chips do you need to represent this state diagram? Explain.

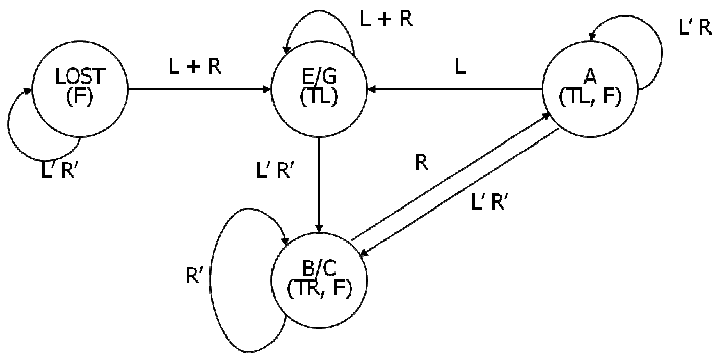
(b) [10pts] Let's assume the first partition has states S_0, S_1, and S_4. Let's assume that we want to partition the rest one more time. Choose the best partition strategy (or strategies) and explain why this is a good solution (no need to draw a diagram).

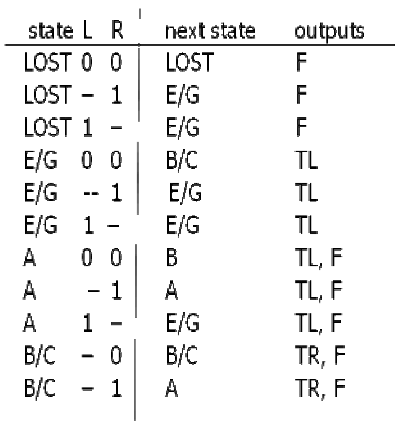
(c) [20pts] Let's assume you need to partition this state diagram into two groups: $S_0 – S_2$ and S_3—S_6. Draw the partitioned state diagram. Make sure all arcs and states are labeled properly. To prevent page flipping, the duplicate state diagram copied here.



(40pts) Problem 4: State Encoding

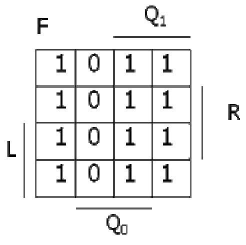
In this problem, we will try three different state-encoding strategies for the robotic ant maze and compare which one works well for this FSM. I copied you the state diagram and state-transition table we had in Lecture 19.



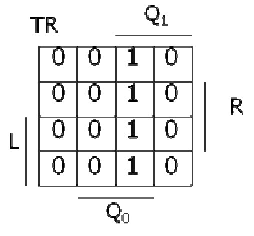


state	L	R	next state	outputs
LOST	0	0	LOST	F
LOST	–	1	E/G	F
LOST	1	–	E/G	F
E/G	0	0	B/C	TL
E/G	--	1	E/G	TL
E/G	1	–	E/G	TL
A	0	0	B	TL, F
A	–	1	A	TL, F
A	1	–	E/G	TL, F
B/C	–	0	B/C	TR, F
B/C	–	1	A	TR, F

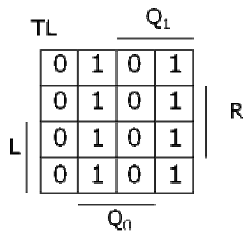
(a) [10pts] If you encode the states using sequential encoding (as we did in class), here are the k-maps you end up with. Write down the logic equations for the output functions (F, TR, and TL) using Q_1, Q_0, L and R. If we used only AND and OR gates (and you can have as many inverters as you want for free implicitly), how many AND and OR gates required to express these output functions (total of all three outputs)? No need to draw the circuit schematics.



1	0	1	1
1	0	1	1
1	0	1	1
1	0	1	1



0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0



0	1	0	1
0	1	0	1
0	1	0	1
0	1	0	1

(b) [15pts] If you encode the states using one-hot encoding, write down the logic equations for the output functions (F, TR, and TL) using Q's (as state bits), L and R. If we used only AND and OR gates, how many AND and OR gates required to express these output functions (no need to draw a circuit)?

(c) [15pts] Output-oriented encoding question removed.

(40pts) Problem 5: Sequence Detector

You will be designing a FSM that detects sequence of 11 or 0010 in a sequence of inputs. Reset starts the sequence of inputs. Once one of the sequences are detected, then the FSM outputs a 1 until it is reset.

(a) [10pts] Draw a state diagram of this sequence detector as a Moore machine. Use S_0, S_1, etc for states. Hint: there are 7 states.
(b) [10pts] Fill in a state transition table assuming a Moore machine. Include reset as one of the inputs. No need to 'encode' the states.
(c) [10pts] Draw a state diagram of this sequence detector as a Mealy machine. Use S_0, S_1, etc for states. Hint: there are 7 states.
(d) [10pts] Fill in a state transition table assuming a Mealy machine. Include reset as one of the inputs. No need to 'encode' the states.