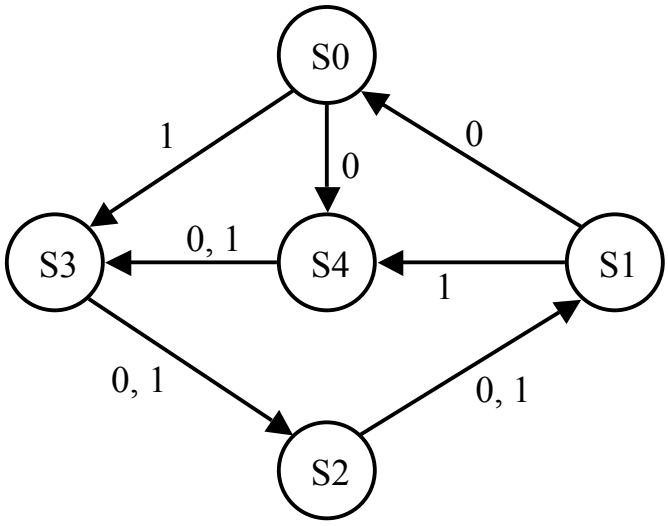
CPRE 281 -- Practice questions for exam #3

1. Construct a 3-bit register with parallel load input using D Flip-Flops and multiplexers.
2. Construct a 4-bit synchronous down-counter with Enable by T flip-flops and any number of 2-input AND gates, 2-input OR gates and NOT gates.
3. Suppose you are given one 3-bit up/down counter, one T flip-flop, two 4-input AND gates, one 2-input OR gate, and several NOT gates. The direction of the counter is controlled by a 1-bit signal K. If K=0, the counter will count up. If K=1, the counter will count down. Using only the components given above, construct a special counter with the following counting sequence:

   000, 001, 010, 011, 100, 101, 110, 111, 110, 101, 100, 011, 010, 001, 000, 001, …..

   counting up counting down counting up

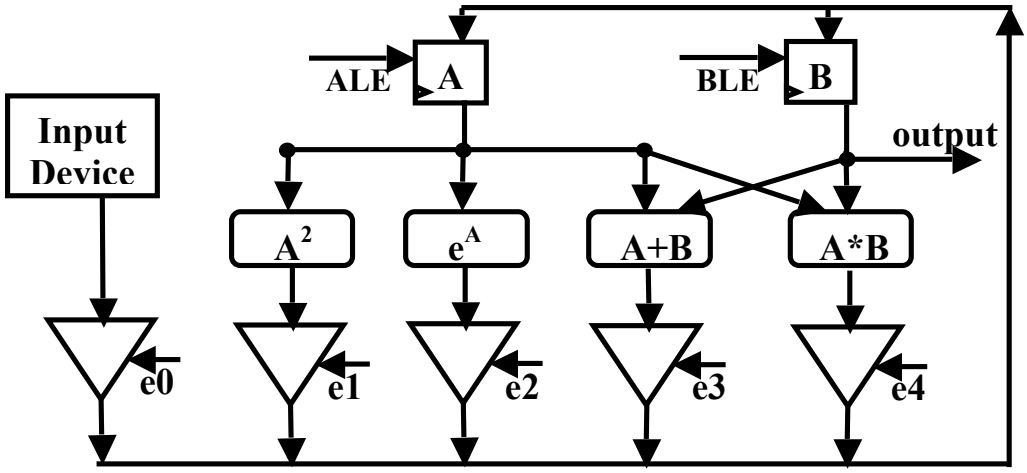
4. Draw the state diagram for a state machine that will repeatedly output the following 4-bit sequence : 0000, 0101, 1001, 0100, 0000, 0001
   At least how many bits are required to encode the states?
5. Below is the state diagram for a state machine with one input variable X.
   (a) Give an state-assigned table for the state machine below. Label your table appropriately. Write down the state assignment you used.



   (b) Derive the simplest next-state expressions in SOP form.
   (Note that we are not working on the outputs in this question.)
6. Assume the initial state for the state machine in Question 5 is S0. Trace the machine for the input sequence 1, 0, 1, 0, 0, 1, 1, 0. What is the final state?
7. You are to design a controller, in the form of a state machine, for a 25-cent gumball machine. Your controller circuit should accept one 2-bit input, C1 C0, to be interpreted as follows: C1 C0 = 01 denotes an input of 5 cents; C1 C0 = 10 denotes an input of 10 cents; C1 C0 = 11 denotes an input of 25 cents. Any amount over 25 cents is merely considered as 25 cent. Once you enter 25 cents or more, subsequent entries will be accepted but ignored (i.e., you are to stay in the "stopping" state, that is, the machine can take more money but not do anything). Two seven-segment display units are used to

display the cumulative amount entered up through 25 cents. You can assume that each seven-segment display unit are already wired to a decoder with 4 bits of input such that the decimal digit *n* will be displayed when *n* as unsigned binary number is supplied to the decoder. For example, '3' will be displayed when 0011 is supplied to the decoder. Your solution should be in the form of a state diagram which generates the output bits to the decoders to display the cumulative amount entered.

8. Design a sequential circuit that will read in a sequence of binary digits, one at a time. The output is 1 and the machine will stop when it finds the pattern 1100. To stop the machine, merely have it orbit in the state it reaches after the successful match.
9. Design a state machine with 1-bit input and 1-bit output such that the output is 1 when (number of "1" in the input sequence) modulo 8 is a prime.
10. Consider the following datapath:



Note that the function unit A^2 and e^A are only connected to register A. The output is only connected to register B. (In other words, the final result must be stored in register B.) Suppose the value x has already been stored inside register A. The datapath can be used to compute different functions of x. For example, to compute $g(x) = x + x^2$, the sequence of steps can be written as follows:

Step 1: B ← A^2
Step 2: B ← A + B

In this question, you are to design a state machine to generate the control signals for the datapath to compute $f(x) = x * e^x + e^x$. You are supposed to compute the function in its current form and are not supposed to simplify it.

(a) Give the sequence of steps to compute the function.
(b) For each step in the sequence of part (a) above, write the corresponding control signals for the datapath.

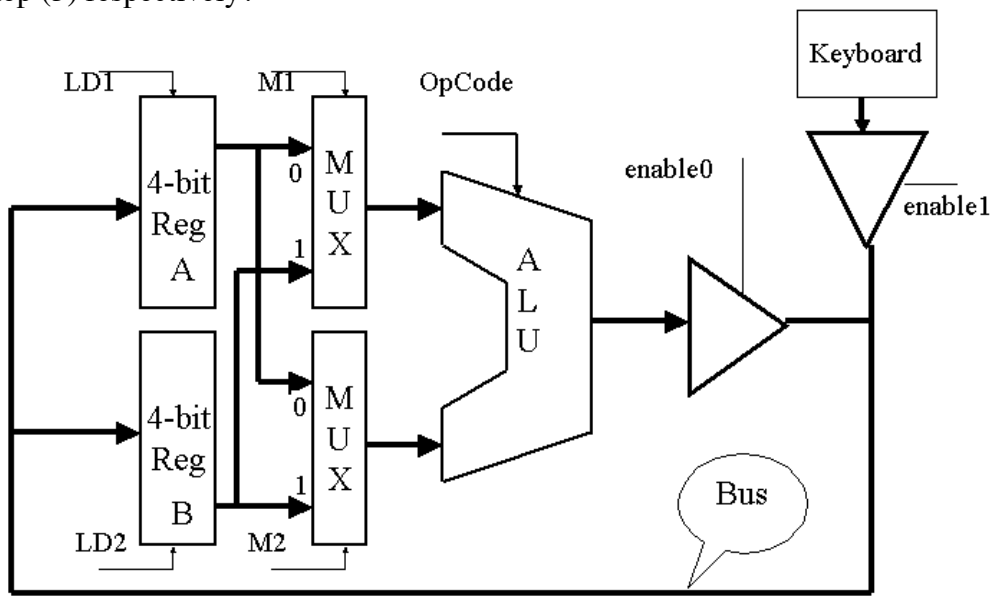
11. For the datapath shown below, assume the Opcode is a 2-bit code defined below:

Opcode	Function
00	ADD
01	SUB
10	AND
11	OR

Write the control signals (LD1, LD2, M1, M2, OpCode, enable0, enable1) for the following **sequence** of operations:

(1) load a value into register A from the keyboard;
(2) load a value into register B from the keyboard;
(3) store the sum of the values in register A and register B into register A;
(4) store the difference of the values in register A and register B into register B; and
(5) store the difference of the values in register A and register B into register A.

If the value loaded into register A in step (1) is 3, and the value loaded into register B in step (2) is 5, what are the values of register A and register B after step (3), step (4), and step (5) respectively?



12. Consider the datapath in Question 11 above. You may notice that the sequence of five operations described above actually swaps the contents in register A and register B. Suppose in this question that the OpCode is 1 bit and the corresponding operations are defined below:

OpCode	Operation
0	Input1 XOR Input2
1	Input1 + Input2

(a) Write the control signals (LD1, LD2, M1, M2, OpCode, enable0, enable1) for the following sequence of operations:
   i) load a value into register A from the keyboard;
   ii) load a value into register B from the keyboard;
   iii) perform XOR of register A and register B and store the result back to register A;
   iv) perform XOR of register A and register B and store the result back to register B;
   v) perform XOR of register A and register B and store the result back to register A;

(b) If the value loaded into register A in step i) is 0101, and the value loaded into register B in step ii) is 1100, what are the values of register A and register B right after each of the five operations?

(c) You may have noticed that this is another way to swap the contents of register A and register B as in the homework. Which way of swapping numbers is better? Why?

13. Consider the datapath in HW#12 Lec. 40 & 41 Question 2. Please design a controller for the datapath such that it will find the remainder of 14 divided by 4, instead of finding GCD(14,4).
14. Suppose a particular type of flip flop is used in a computer. The FF propagation delay is 0.15nsec and the FF set-up time is 0.25nsec.
    (a) If you want to design a 1000MHz computer, what is the maximum time allowed to compute the next state?
    (b) If you want to design a 1333MHz computer, what is the maximum time allowed to compute the next state?
15. In Ch. 7.5 of textbook, a divider designed as a sequential circuit is presented. In this question, we will design a divider circuit as a combinational circuit. The integer division function DIV of two nonnegative numbers A and B is to find the largest integer smaller than A / B. For example, 5 DIV 2 = 2, 8 DIV 3 = 2, and 9 DIV 3 = 3. A 2-bit integer divider circuit takes two 2-bit nonnegative numbers A1 A0 and B1 B0 as inputs. The circuit generates the three bits R1, R0, and E as output. R1 R0 is the result of A1 A0 DIV B1 B0. The bit E is to indicate the "Divide by zero" error (i.e., E=1 iff B1 B0 = 00).
    (a) Write a truth table for the 2-bit integer divider circuit.
    (b) Simplify the three sum-of-product expressions R1, R0, and E as much as possible using K-maps.
    (c) Implement the three expressions using AND gates, OR gates, and NOT gates.