Department of Electrical and Computer Engineering
The University of Texas at Austin

EE 306, Fall, 2006
Yale Patt, Instructor
TAs: Aseem Bathla, Cameron Davison, Lisa de la Fuente, Phillip Duran, Jose Joao,
Jasveen Kaur, Rustam Miftakhutdinov, Veynu Narasiman, Nady Obeid, Poorna Samanta.
Exam 1, October 04, 2006

Name:_____

Problem 1 (20 points):_____

Problem 2 (10 points):_____

Problem 3 (15 points):_____

Problem 4 (15 points):_____

Problem 5 (10 points):_____

Problem 6 (15 points):_____

Problem 7 (15 points):_____

Total (100 points):_____

Note: Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space provided.

Note: Please be sure your name is written legibly on each sheet of the exam.

**I will not cheat on this exam.**

_____
Signature

**GOOD LUCK!**

Name:_____

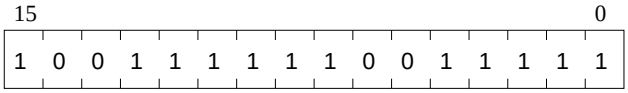
**Problem 1 (20 points)**

**Part a** (3 points): We have learned that we can write one bit of information with a logic circuit called a transparent latch, and that the bit written is available to be read almost immediately after being written.

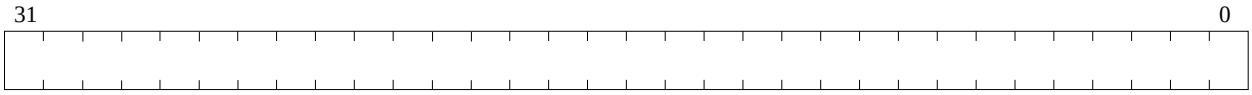
Sometimes it is useful to be able to store a bit, but not be able to read the value of that bit until the next

cycle. An example of a logic circuit that has this property is a _____.

**Part b** (3 points): The value -24,673 can be represented as a 2's complement integer with 16 bits as shown below:




Your job: Represent -24,673 as a 2's complement integer with 32 bits in the box provided.



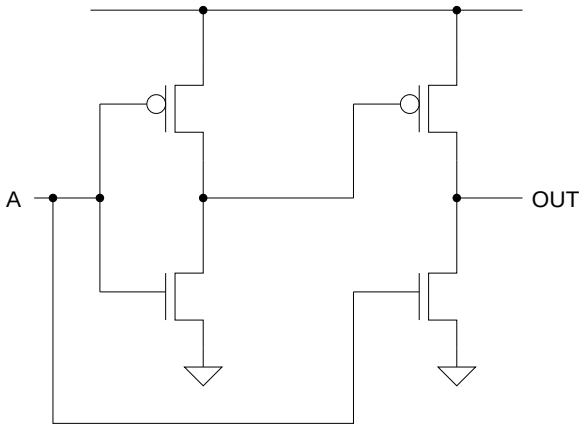

**Part c** (4 points): There are two kinds of logic circuits.

In a _____ logic circuit, the output is a function of the inputs as they are right now.

In a _____ logic circuit, the output is not only a function of the inputs right now, but also a function of previous inputs that have occurred in the past.

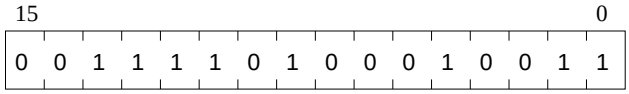
Name:_____

**Part d** (4 points): What does the following transistor circuit do? ... in 20 words or fewer (It took us 9 words). Put your answer in the box below.

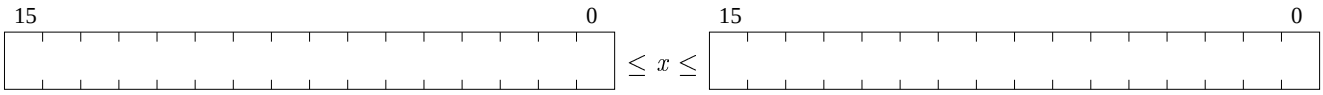



Answer:

**Part e** (6 points): A 16-bit 2's complement number is shown below:




There are values $x$ which, if added to the number above, will cause an overflow condition. What is the range of such values? Fill in the 2's complement binary values in the space provided below that will specify the range of $x$.



Name:_____

**Problem 2 (10 points)**

The representation of A in base $x$ is 13. The representation of B in base $x$ is 23. The representation of their sum (also in base $x$) is 40.

**Part a** (5 points): What is $x$?

Answer:

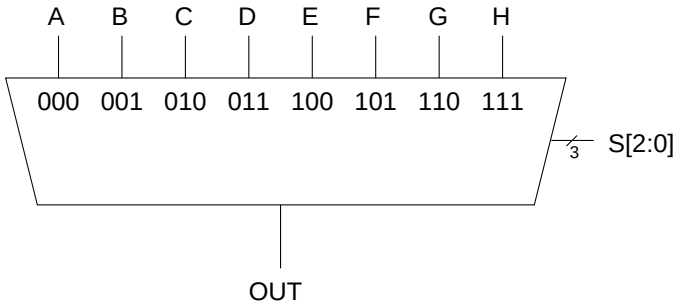
**Part b** (5 points): The representation of C in base $x$ is 241. What is the decimal value of C?

Answer:

Name:_____

**Problem 3 (15 points)**

An 8-to-1 mux (shown below) outputs one of the eight sources, A,B,C,D,E,F,G,H, depending on S[2:0], as shown. Note the value of S[2:0] corresponding to each source is shown just below the input to the mux. For example, when S[2:0] = 001, B is provided to the output.

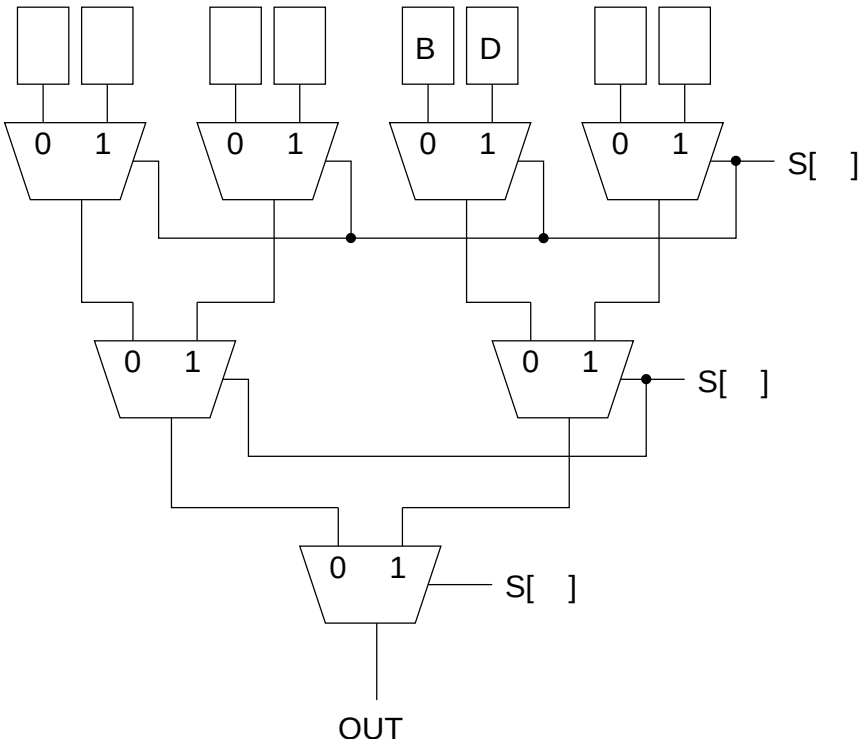



We can implement an 8-to-1 mux with a logic circuit of 2-to-1 muxes, as shown below. In this case, the 0 and 1 below the two inputs to each mux correspond to the value of the select line that will cause that input to be provided to the output of that mux.

Note that only two of the sources are shown. Note also that none of the select bits are labeled. Your task: finish the job.

**Part a** (7 points): Label the select line of each mux, according to whether it is S[2], S[1], or S[0].

**Part b** (8 points): Label the remaining six sources to the 2-to-1 mux circuit, so the circuit behaves exactly like the 8-to-1 mux shown above.



Name:_____

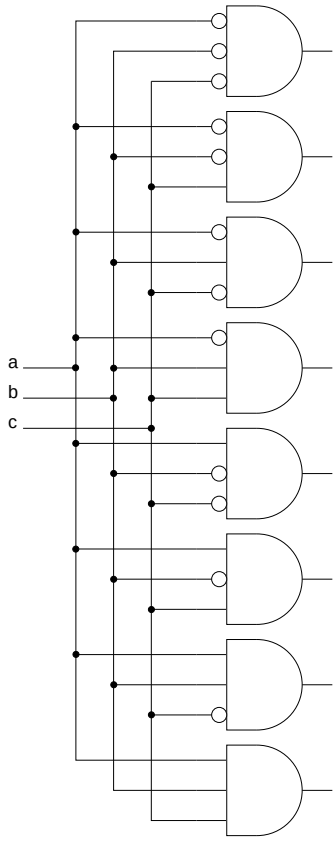
**Problem 4 (15 points)**

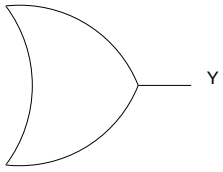
We wish to implement two logic functions Y(a,b,c) and Z(a,b). Y is 1 in exactly those cases where an odd number of a, b, and c equal 1. Z is the exclusive-OR of a and b.

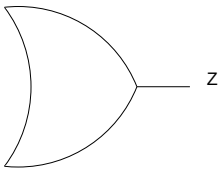
**Part a** (9 points): Construct the truth tables for Y and Z.

| a | b | c | Y | Z |
|---|---|---|---|---|
| 0 | 0 | 0 | | |
| 0 | 0 | 1 | | |
| 0 | 1 | 0 | | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |

**Part b** (6 points): Implement the two logic functions Y and Z described above using ONLY the logic circuits provided below: a 3-to-8 decoder and two OR gates. That is, draw the wires from the outputs of the decoder to the inputs of the OR gates as necessary to do the job. You can assume you have as many inputs to each OR gate as you find necessary.



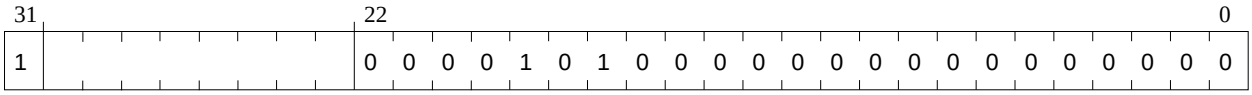



Name:_____

**Problem 5 (10 points)**

Shown below is the IEEE Floating Point representation of a value. Note that the exponent bits have been left out.




**Part a** (5 points): Fill in the exponent bits so that the value being represented is an integer. If you feel there is more than one possible answer, then the "correct" answer is the integer having the smallest absolute value. (Recall that the IEEE Floating Point representation uses a BIAS of 127 for the exponent representation.)

**Part b** (5 points): What is the decimal value of the integer represented in part a?

Answer:

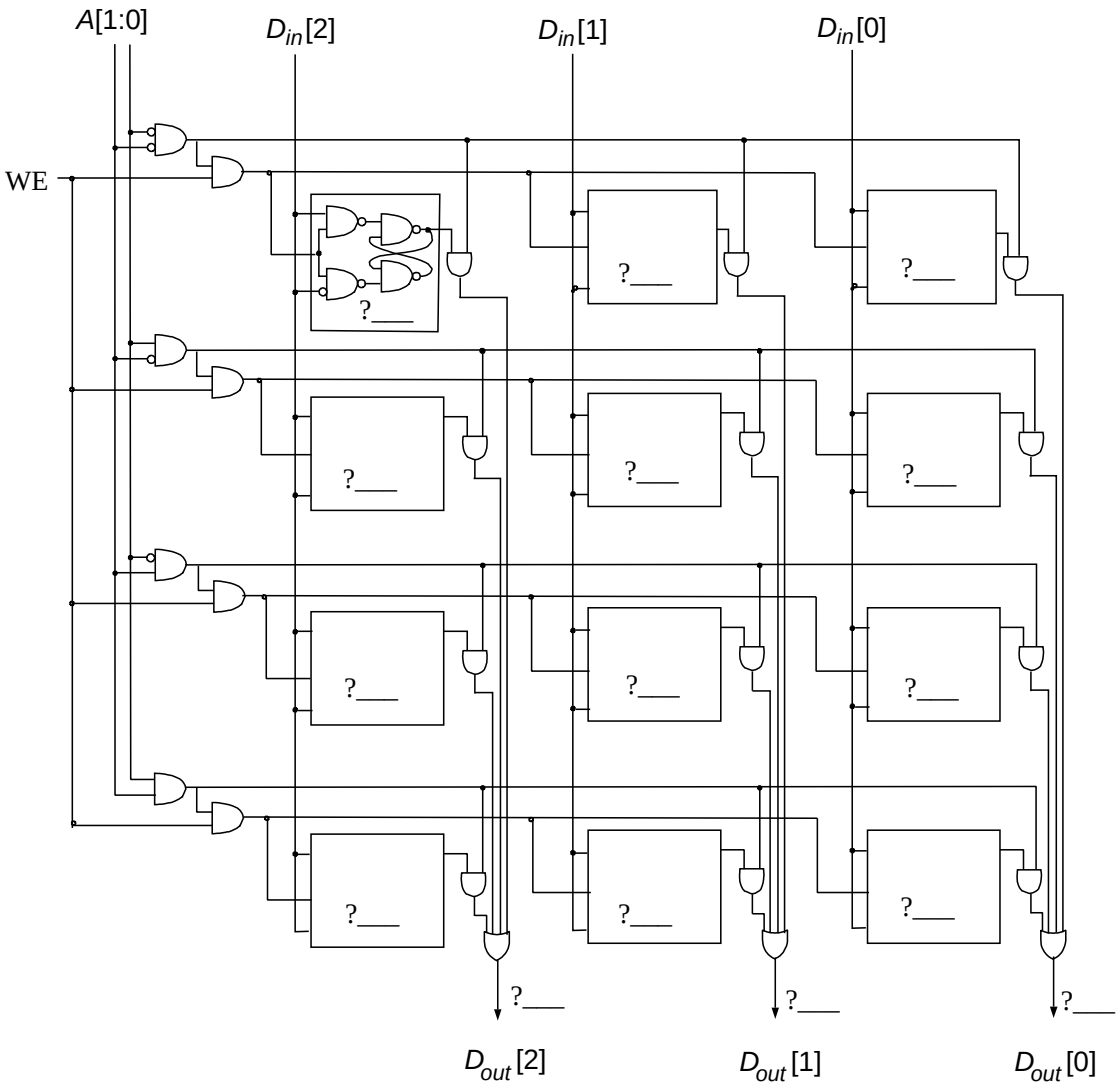
Name:_____

**Problem 6 (15 points)**

The $2^2$ by 3 bit memory discussed in class is accessed during five consecutive clock cycles. The table below shows the values of the 2-bit Address, 1-bit Write Enable, and 3-bit Data-In signals during each access.

| | A[1:0] | WE | $D_{in}$[2:0] |
|---|---|---|---|
| cycle 1 | 0 1 | 1 | 1 0 1 |
| cycle 2 | 1 1 | 0 | 1 1 0 |
| cycle 3 | 1 0 | 1 | 0 1 0 |
| cycle 4 | 0 1 | 1 | 0 1 1 |
| cycle 5 | 0 0 | 0 | 0 0 0 |

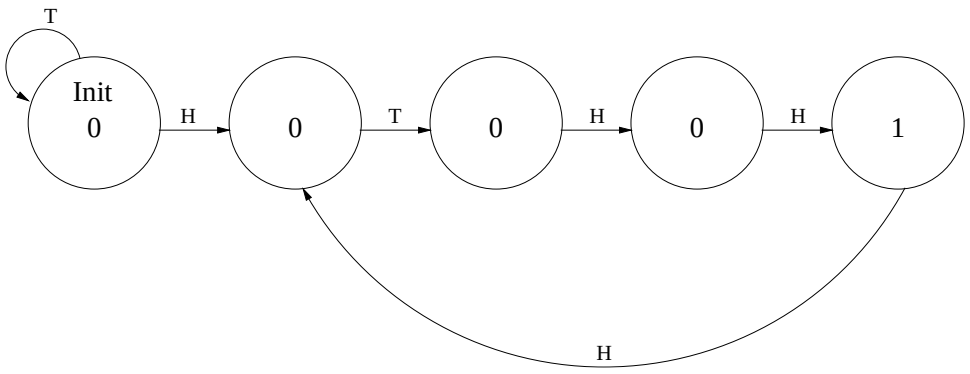
Your job: Fill in the value stored in each memory cell and the three Data-Out lines just before the end of the $5^{th}$ cycle. Assume initially that all 12 memory cells store the value 1. In the figure below, each question mark (?) indicates a value that you need to fill in.



Name:_____

**Problem 7 (15 points)**

Shown below is the partially completed state diagram of a finite state machine that takes an input string of H (heads) and T (tails) and produces an output of 1 every time the string HTHH occurs.




For example,

if the input string is: H H H H H T H H T H H H H H T H H T,
the output would be: 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0.

Note that the $8^{th}$ coin toss (H) is part of two HTHH sequences.

**Part a** (12 points): Complete the state diagram of the finite state machine that will do this for any input sequence of any length.

**Part b** (3 points): If we decide to implement this finite state machine with a sequential logic circuit (similar to the danger sign we designed in class), how many state variables would we need?

Answer: