ICS hw2

howlinggggg

October 22, 2022

T1.

(a.) What is the **smallest** positive **normalized** number that can be represented using the IEEE 32-bit Floating Point standard?

(b.) What is the **largest** positive **subnormal** number that can be represented using the IEEE 32-bit Floating Point standard?

answer.

(a.) 0 0000 0001 000 0000 0000 0000 0000 0000

$$
\begin{aligned}
N &= (-1)^0 \times (1.000\ 0000\ 0000\ 0000\ 0000\ 0000)_b \times 2^{((0000\ 0001)_b - 127)} \\
&= 1 \times (2)^0 \times 2^{1-127} \\
&= 2^{-126}
\end{aligned}
$$

(b.) 0 0000 0000 111 1111 1111 1111 1111 1111

$$
N = (1 - 2^{-23}) \times 2^{-126}
$$

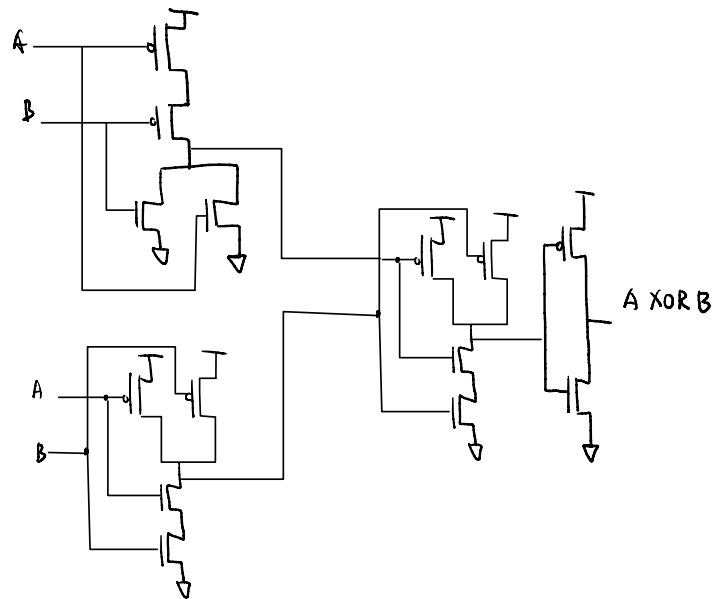**T2.** What is the largest positive number that can be represented in a 32-bit 2's complement scheme?

answer. 0 111 1111 1111 1111 1111 1111 1111 1111

$$
N = 2^{31} - 1
$$

T3. Draw the transistor level circuit of a 2 inputs XOR gate.

answer.

$$
\begin{aligned}
A \oplus B &= AB' + A'B \\
&= (A + B)(A' + B') \\
&= (A + B)(AB)'
\end{aligned}
$$

A XOR B

T4. The transistor circuit shown below produces the accompanying truth table. The inputs to some of the gates of the transistors are not specified. Also, the outputs for some of the input combinations of the truth table are not specified. Complete both specifications, i.e., all transistors will have their gates properly labeled with either A, B, or C, and all the truth table rows will have a 0 or 1 specified as the output.
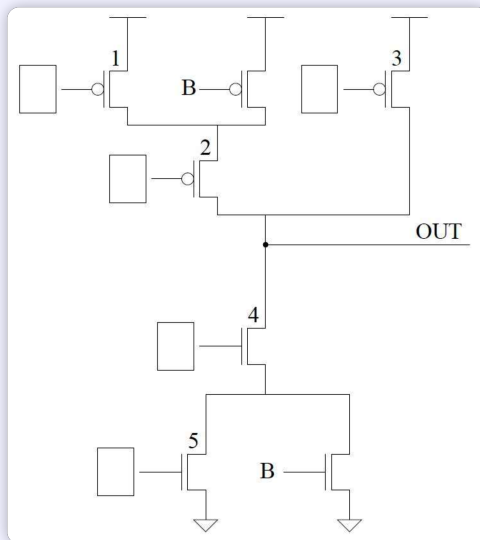
A	B	C	OUT
0	0	0	
0	0	1	
0	1	0	
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	
1	1	1	

answer.

$$((X_1' + B')X_2' + X_3')(X_4(X_5 + B))' = (X_1'X_2' + B'X_2' + X_3')(X_4' + (X_5 + B)')$$
$$= (X_1'X_2' + B'X_2' + X_3')(X_4' + X_5'B')$$
$$= X_1'X_2'X_4' + X_2'X_4'B' + X_3'X_4' +$$
$$X_1'X_2'X_5'B' + X_2'X_5'B' + X_3'X_5'B'$$
$$= B'(X_2'X_4' + X_2'X_5' + X_3'X_5') + X_4'(X_1'X_2' + X_3')$$

(A, B, C) = (0, 1, 1)

$$X_4'(X_1'X_2' + X_3') = 1 \Rightarrow X_4 = A, X_3 = A \text{ or } X_1 = X_2 = A$$

(A, B, C) = (1, 0, 0)

$$X_5'(X_2' + X_3') = 1$$

(A, B, C) = (1, 0, 1)

$$X_5'(X_2' + X_3') = 0$$

$$X_1 = A, X_2 = C, X_3 = A, X_4 = A, X_5 = C$$

A	B	C	OUT
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

T5. Shown below are several logical identities with one item missing in each. X represents the case where it can be replaced by either a 0 or a 1 and the identity will still hold. Your job: Fill in the blanks with either a 0, 1, or X. For example, in part a, the missing item is X. That is 0 OR 0 = 0 and 0 OR 1 = 1

- 0 OR X = __

- 1 OR X = __

- 0 AND X = __

- 1 AND X = __

- __ XOR X = X

answer.

- 0 OR X = X

- 1 OR X = 1

- 0 AND X = 0

- 1 AND X = X

- 0 XOR X = X

T6. Logic circuit 1 in Figure 3.39 (page 102 of the book) has inputs A, B, C. Logic circuit 2 in Figure 3.40 (page 102 of the book) has inputs A and B. Both logic circuits have an output D. There is a fundamental difference between the behavioral characteristics of these two circuits. What is it?

Hint: What happens when the voltage of one input goes from 0 to 1 in both circuits?

Figure 3.39   Logic circuit 1 for Exercise 3.25.

Figure 3.40   Logic circuit 2 for Exercise 3.25.
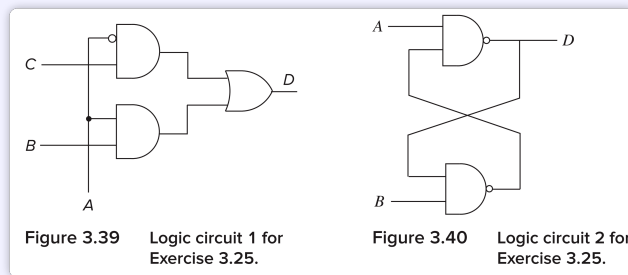
answer.

- figure 1: AB+A'C = D

- figure 2: It's an R-S latch.

T7.

- How many output bits does a five-bit-input decoder have?

- How many output bits does a 16-to-1 MUX have? How many select bits does this MUX have?

answer.

- output: $2^5 = 32$ bits.

- output: 1 bit, select: 4 bits.

T8.   Say the speed of a logic structure depends on the largest number of logic gates through which any of the inputs must propagate to reach an output. Assume that a NOT, an AND, and an OR gate all count as one gate delay. For example, the propagation delay for a two-input decoder shown in Figure 3.11 is 2 because some inputs propagate through two gates.

Figure 3.11   A two-input decoder.
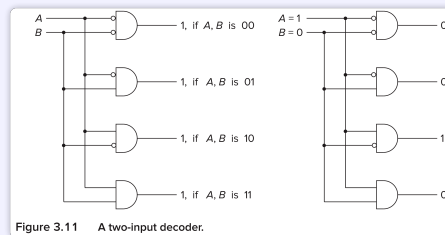
1. What is the propagation delay for the two-input MUX shown in Figure 3.12 (page 68)?

Figure 3.12   A 2-to-1 mux.
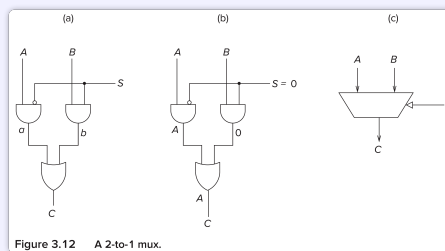
2. Can you reduce the propagation delay for the circuit shown in Figure 2 by implementing the equation in a different way? If so, how?

Figure 2
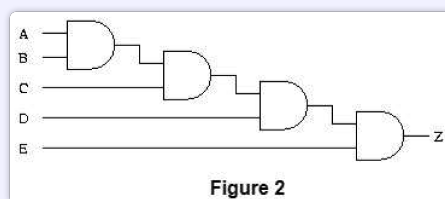
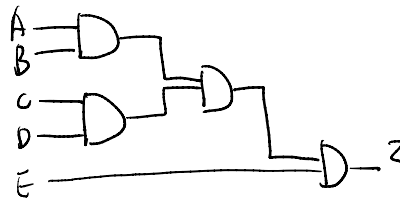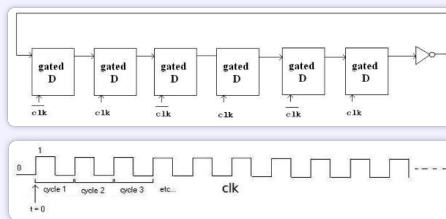**answer.**

1. delay: 3. NOT(S), (A AND NOT(S)), (a+b).

2.
$$Z = ABCDE$$
$$= (AB)(CD)E$$

5

A logic circuit consisting of 6 gated D latches and 1 inverter is shown below:

Let the state of the circuit be defined by the state of the 6 D latches. Assume initially the state is 000000 and CLK starts at the point labeled t0. Question: What is the state after 50 cycles? How many cycles does it take for a specific state to show up again?

answer.

1. 111000

2. cycle 1: 100000

   cycle 2: 111000

   cycle 3: 111110

   cycle 4: 011111

   cycle 5: 000111

   cycle 6: 000001

   cycle 7: 100000

   6 cycles.

T10. Prove that NAND is logically complete.

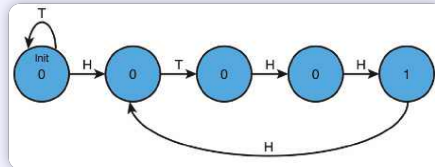*Proof.* Known that the set of gates {AND, OR, NOT} is logically complete.
A AND B = NAND(A NAND B)
A OR B = ((NAND(A)) NAND (NAND(B)))

$NOT(A) = NAND(A)$

NAND is logically complete. □

T11. Shown below is a 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

a.) Complete the state diagram of the finite state machine that will do this for any input sequence of any length.

b.) If this state machine is implemented with a sequential logic circuit how many state variables will be needed? (Recall, the number of state variables is the same as the number of bits needed to represent all of the states.)
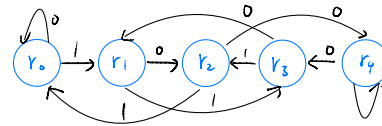
answer.

a.)
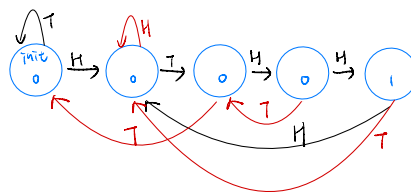
b.) 6

T12. Now consider a set of strings that only consist of 0 and 1. Shown below is a diagram of the state machine that takes a string that contains an even number of zeros and an even number of ones. (eg: 000110, 10100011 will be taken, 01011, 111000 will not be taken). Please design a finite state machine

to take a string when the unsigned binary number represented by the string is divisible by 5. (eg: 001010, 101, and 0000 are divisible by 5, 01011, and 111000 are not divisible by 5)

Hint: The number of states can be derived from the remainder when divided by 5.

answer.

number	binary number	remainder
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	0
6	110	1
7	111	2
8	1000	3
9	1001	4

T13. If a particular computer has 8 **byte** addressability and an 8 **bit** address space, how many bytes of memory does that computer have?

answer. $2^{8+3} = 2^{11}$