HW2

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?

(a) Normalized 的最小表示正数应为 $0\ 00000001\ 00000000000000000000000$, 即 $(-1)^0 * 2^{-126} * 1.00000000000000000000000$, i.e. 2^{-126} .
(b) *此处 Subnormal 是否指的是 Denormalized ?*
Denormalized 最大表示正数应为 $0\ 00000000\ 11111111111111111111111$, 即 $(-1)^0 * 2^{-126} * 0.11111111111111111111111$, 即 $(1-2^{-23}) * 2^{-126}$.

T2

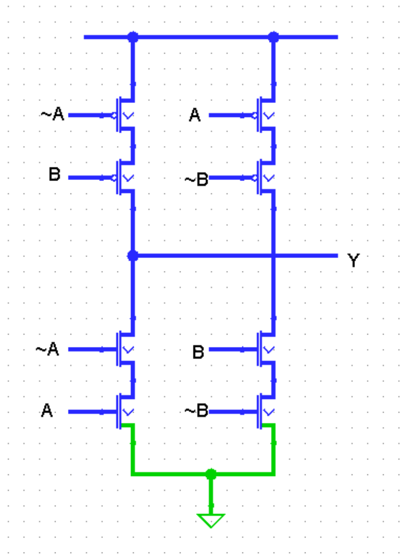
> What is the largest positive number that can be represented in a 32-bit 2's complement scheme?

对于正数而言补码和原码一致，最大的补码为 $01111111111111111111111111111111$, 即 $2^{31}-1$, 所以能表示出的最大整数是 $2^{31}-1$.

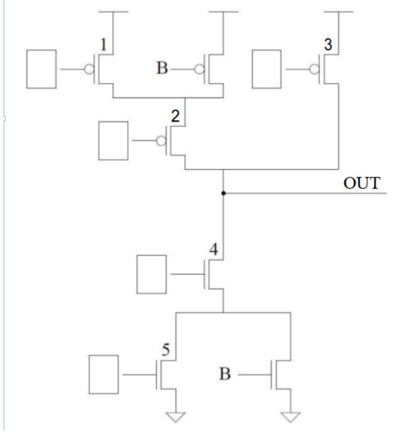
T3

> Draw the transistor level circuit of a 2 input XOR gate.

这个题不给 "you may assume you have both true and complementary versions of the input" 的话也太阴间了...



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 rows of the truth table 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 | |

真值表:

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

1-A 2-C 3-A 4-A 5-C

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

- 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?*

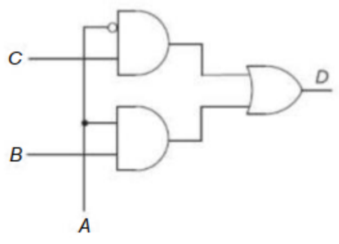


Figure 3.39 **Logic circuit 1 for Exercise 3.25.**

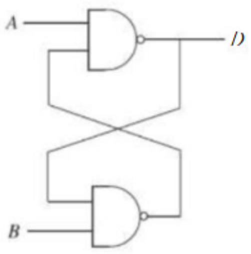


Figure 3.40 **Logic circuit 2 for Exercise 3.25.**

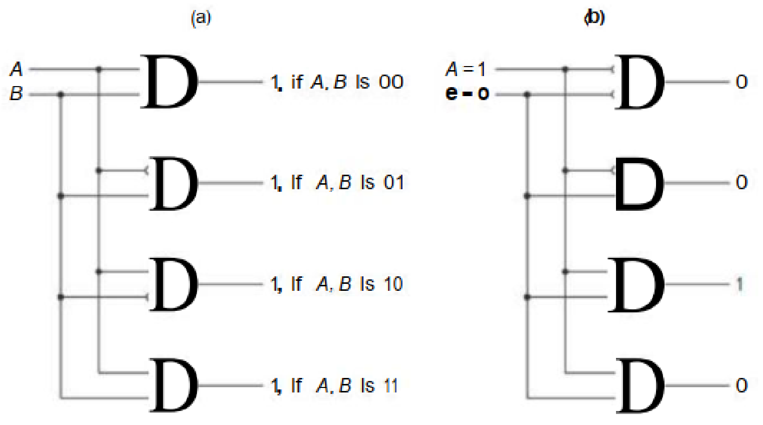
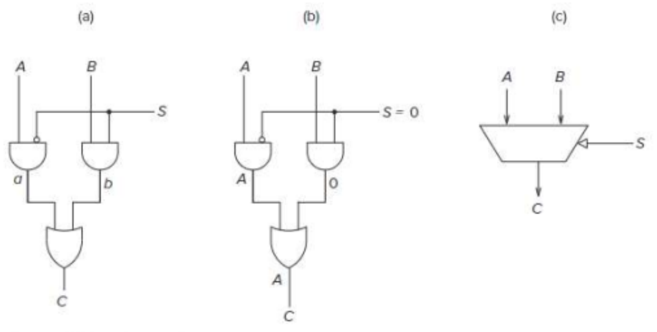
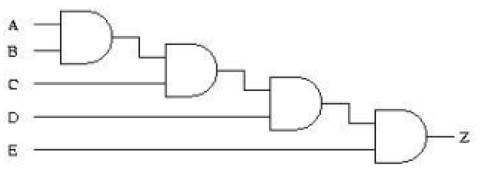
Figure 3.39 是组合逻辑电路，输出结果仅与三个输入是什么有关，而与它们的顺序和变化无关；Figure 3.40 是时序逻辑电路，输出结果与A、B 的现态和变化都有关。

T7

> (a) How many output bits does a five-bit-input decoder have?
> (b) How many output bits does a 16-to-1 mux have? How many select bits does this mux have?

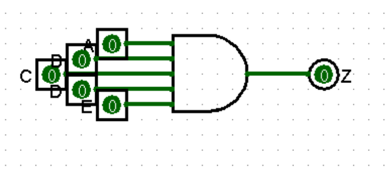
(a) 32.
(b) 1 bit 16-1 MUX 只有一个输出，1个输出位。它有4个选择器，才能满足对16种情况的选择。

T8

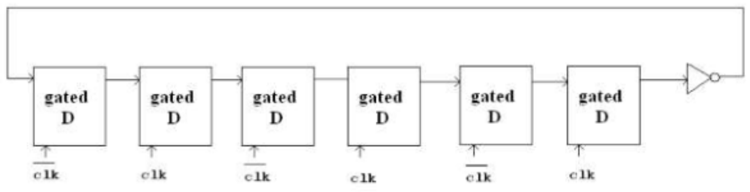
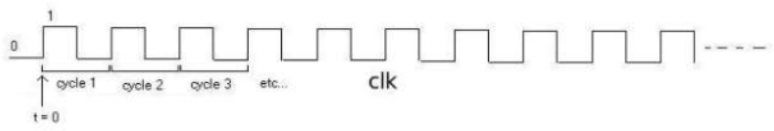
> Say the speed of a logic structure depends on the largest number of logic gates through which any of the inputs must propagate to reach an output. Assume that a NOT, an AND, and an OR gate all count as one gate delay. For example, the propagation delay for a two-input decoder shown in Figure 3.11 is 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)?
> 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 3.12 A 2-to-1 mux.
>
> 
>
>
> **Figure 2**

1. 输入要经过3个门，所以答案为3.

2. $Z = A \cdot B \cdot C \cdot D \cdot E$ ，直接把所有输入输入到同一个与门里就好了�?



T9

> 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 if 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?
>
> 
>
>
> Figure 5
>
> 
>

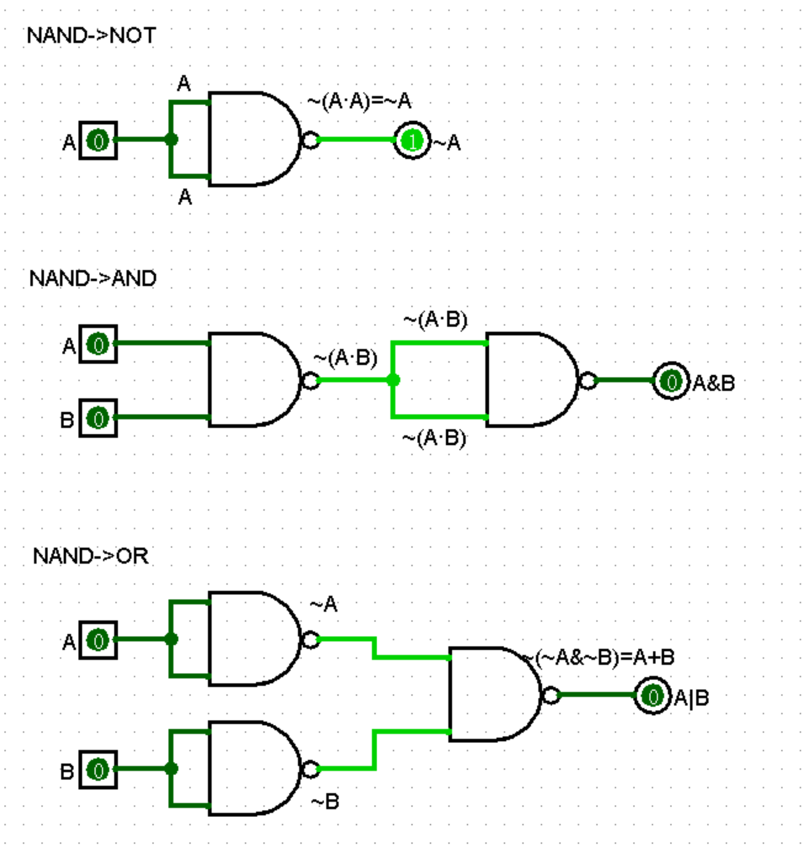
1. $50 = 6 * 8 + 2$, 所以经过50个周期后：`111000` .
2.

```
after cycle 1: 100000
after cycle 2: 111000
after cycle 3: 111110
after cycle 4: 011111
after cycle 5: 000111
after cycle 6: 000001
after cycle 7: 100000
```

每经过6个时钟周期循环一次。

T10

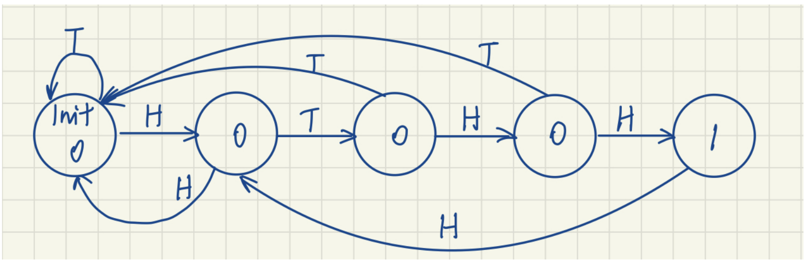
> Prove that NAND is logically complete.



T11

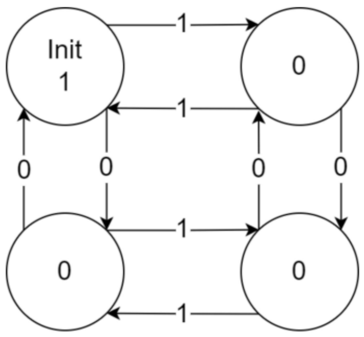
> Shown below is a partially completed state diagram of a finite state machine that takes an input string of *H* (heads) and *T* (tails) and produces an output of 1 every time the string HTHH occurs.
> For example, if the input string is: `H, H, H, H, H, T, H, H, T, H, H, H, H, H, T, H, H, T`, the output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

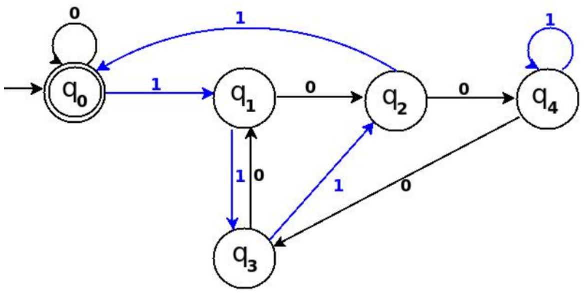
a)



b) 6.

T12

> Now consider a set of strings which only consists of 0s and 1s. Shown below is a diagram of a state machine that takes a string which contains an even number of 0s and an even number of 1s. (eg. 000110, 10100011 will be taken, 01011 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, 01011,111000 are not divisible by 5)
> *Hint: The number of states can be derived from the remainder when devided by 5.*
>
> 
>



T13

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

$2^8 * 8 * 8 = 2^{14}\, bits = 2^{11}\, bytes$