中山大学软件学院 **2008** 级软件工程专业(2010学年秋季学期)

《Digital Circuits and Logic Design》期末试题(A卷)

参考答案

1. Multiple choice questions (2 points for each question, 20 points total).

(1) If the period of a clock signal is 500 ps, the frequency is __c

a. 20 MHz
b. 200 MHz
c. 2 GHz
d. 20 GHz

(2) The time measurement between the 90% point on the trailing edge of a pulse to the 10% point on the trailing edge of the pulse is called the __b

a. rise time
b. fall time
c. period
d. pulse width

(3) When two positive signed numbers are added, the result may be larger than the size of the original numbers, creating overflow. This condition is indicated by__a

a. a change in the sign bit
b. a carry out of the sign position
c. a zero result
d. smoke

(4) Assume that a floating point number is represented in binary. If the sign bit is 1, the__a

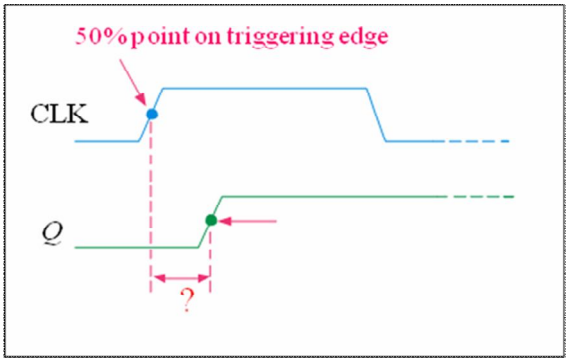
a. number is negative
b. number is positive
c. exponent is negative
d. exponent is positive

(5) A Boolean expression that is in standard SOP form is __*c*

a. the minimum logic expression
b. contains only one product term
c. has every variable in the domain in every term
d. none of the above

(6) The time interval illustrated is called __b

a. t_{PHL}
b. t_{PLH}
c. set-up time
d. hold time



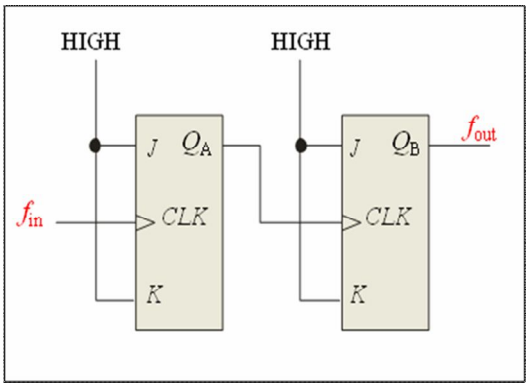
(7) The application illustrated is a__d__

a. astable multivibrator

b. data storage device

c. frequency multiplier

d. frequency divider



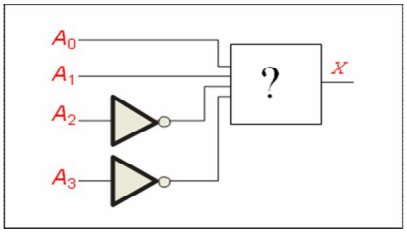
(8) Assume you want to decode the binary number 0011 with an active-HIGH decoder. The missing gate should be __d__

a. an AND gate

b. an OR gate

c. a NAND gate

d. a NOR gate



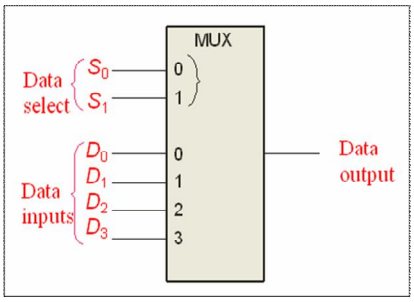
(9) If the data select lines of the MUX are S_1S_0 = 11, the output will be __d__

a. LOW

b. HIGH

c. equal to D_0

d. equal to D_3



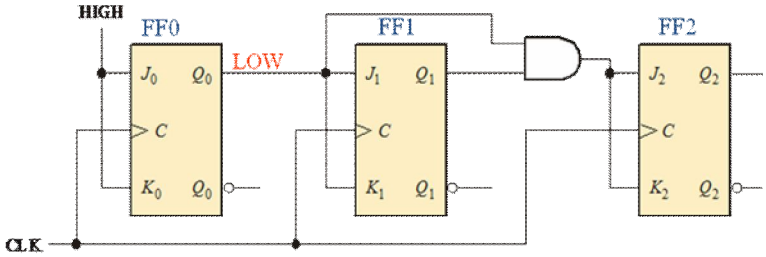
(10)Assume Q_0 is LOW. The next clock pulse will cause __b__

a. FF1 and FF2 to both toggle

b. FF1 and FF2 to both latch

c. FF1 to latch; FF2 to toggle

d. FF1 to toggle; FF2 to latch



2. Please solve the following number conversion problems (total 15 points)

(1) Convert binary number 10111000.10101 to decimal (2 points)

(2) Convert decimal number 100.625 to binary (2 points)

(3) Express decimal number -85 as 8-bit number in 1's complement form (2 points)

(4) Express decimal number -85 as 8-bit number in 2's complement form (2 points)

(5) Convert decimal number 126 to (3 points)

(a) Hexadecimal number

(b) Octal number

(c) 8421 BCD number

(6) Convert the decimal number 3.248 x 10^4 to a single-precision floating point binary number. Please show your procedure (4 points)

---

(1) 184.65625

(2) 110 0100.101

(3) 1010 1010

(4) 1010 1011

(5) (a) H7E (b) O176 (c) 0001 0010 0110

(6) 0 10001101 111_1101_1100_0000_0000_0000

---

3. **We need to design a circuit with 4 inputs(marked as A, B,C,D) and 2 outputs X, Y. Based on system analysis, we should have true table as shown in the following (total 20 points):**

A	B	C	D	*X*	*Y*
0	0	0	0	*0*	*1*
0	0	0	1	*1*	*0*
0	0	1	0	*1*	*1*
0	0	1	1	*1*	*0*
0	1	0	0	*0*	*1*
0	1	0	1	*1*	*0*
0	1	1	0	*1*	*1*
0	1	1	1	*1*	*0*
1	0	0	0	*0*	*1*
1	0	0	1	*1*	*0*
1	0	1	0	*1*	*1*
1	0	1	1	*1*	*1*
1	1	0	0	*1*	*1*
1	1	0	1	*1*	*1*
1	1	1	0	*0*	*1*
1	1	1	1	*1*	*1*

(1) Express the above circuit's outputs in minimum Sum-of-Product (SOP) form (4 points)
(2) Show the implementation of the circuit in SOP form by using only AND, OR, or NOT gates (3 points)
(3) Show the implementation of the circuit in SOP form by using only NAND gates (3 points)
(4) Express the above circuit's outputs in minimum Product-of-Sum (POS) form (4 points)
(5) Show the implementation of the circuit in POS form by using only AND, OR, or NOT gates (3 points)
(6) Show the implementation of the circuit in POS form by using only NOR gates (3 points)

---

(1)

$$X = \overline{A}\,\overline{C}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D}$$

$$Y = \overline{A}D + \overline{B}\,\overline{C}D$$

(2)

$$X = \overline{A}\,\overline{C}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D}$$

$$Y = \overline{A}D + \overline{B}\,\overline{C}D$$

The diagram is omitted.

(3)

$$X = \overline{\overline{A}\,\overline{C}\,\overline{D}} \bullet \overline{\overline{B}\,\overline{C}\,\overline{D}} \bullet \overline{AB\overline{C}\,\overline{D}}$$

$$Y = \overline{\overline{A}D} \bullet \overline{\overline{B}\,\overline{C}D}$$

The diagram is omitted.

(4)

$$X = (A + C + D)(B + C + D)(\overline{A} + \overline{B} + \overline{C} + D)$$
$$Y = (A + \overline{D})(B + C + \overline{D})$$

(5)

$$X = (A + C + D)(B + C + D)(\overline{A} + \overline{B} + \overline{C} + D)$$
$$Y = (A + \overline{D})(B + C + \overline{D})$$

The diagram is omitted.

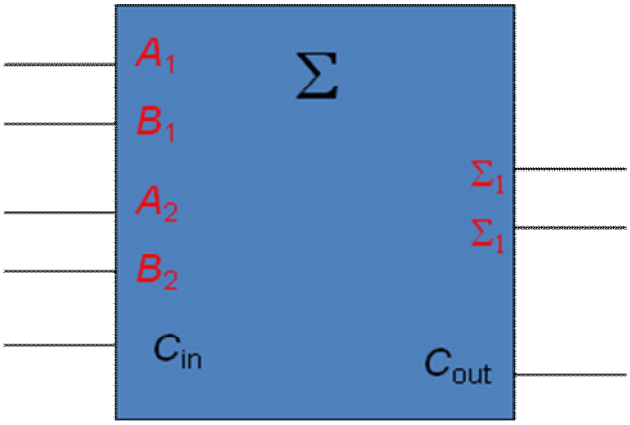
(6)

$$X = \overline{\overline{(A + C + D)} + \overline{(B + C + D)} + \overline{(\overline{A} + \overline{B} + \overline{C} + D)}}$$
$$Y = \overline{\overline{(A + \overline{D})} + \overline{(B + C + \overline{D})}}$$

The diagram is omitted.

---

4. **Design a 2-bit look-ahead-carry full adder using only basic logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR). (total 15 points)**



Denote $C_{out1} = A_1 B_1 + (A_1 + B_1) C_{in}$

$\Sigma_1 = A_1 \oplus B_1 \oplus C_{in}$

$\Sigma_2 = A_2 \oplus B_2 \oplus C_{in1}$

$C_{out} = A_2 B_2 + (A_2 + B_2) C_{in1}$

---

5. **Design a Moore FSM to detect a bit sequence 11001. If such sequence is found, output will be set to 1, otherwise, output will be set to 0. Overlap sequence, if exists, should be detected. Please use J-K flip flop for the system. (total 30 points).**
   - **(1) Derive state diagram (4 points)**
   - **(2) Encode states (state assignment) (2 points)**
   - **(3) Create state-assigned state table (4 points)**
   - **(4) Express J-K Flip-Flop excite table to use (3 points)**
   - **(5) Determine minimum Flip-Flop input equations and FSM output equation(s) (8 points)**
   - **(6) Verify that there is no state that will stay outside the FSM for ever (2 points)**
   - **(7) Draw circuit diagram (7 points)**

Keypoints.

-- ‘S0’ indicates no ‘1’s has been input yet. So the output at this state is ‘0’.

S1: only one ‘1’ has been input; output ‘0’.

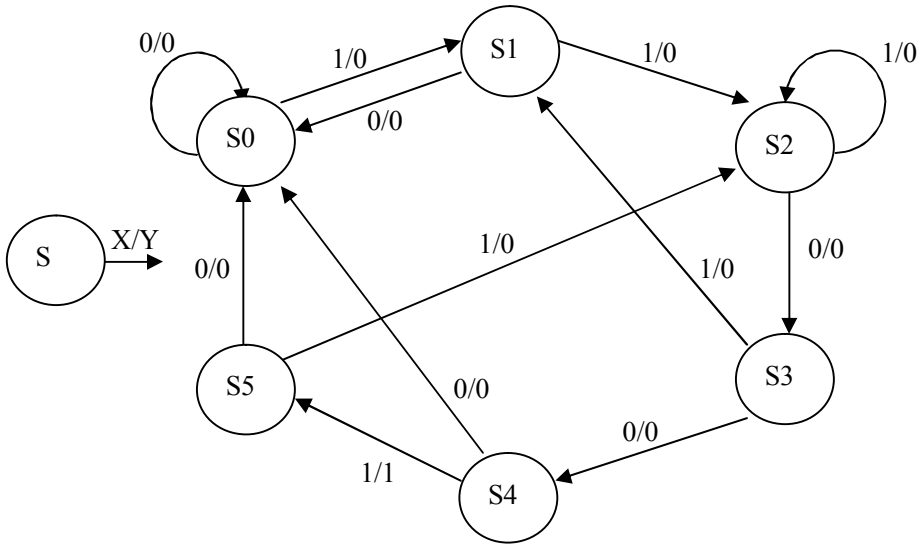
S2: input ‘11’; output ‘0’.

S3: input ‘110’; output ‘0’.

S4: input ‘1100’; output ‘0’.

S5: input ‘11001’; output ‘1’.

-- Denote X as the input and Y the output. The state diagram is shown as follow.



-- Three JK Flip-flops are used in this design. The outputs of JK Flip-flops will indicate the states shown in the above diagram.

-- ‘000’ – ‘S0’; ‘001’—‘S1’; ’010’—‘S2’; ‘011’ –‘S3’; ‘100’ – ‘S4’; ‘101’ –‘ S5’. Others are inactive states. Incorporating this state-assignment, the state diagram above can be translated into the Next-State Karnaugh Map.

Q_2Q_1 \ Q_0X	00	01	11	10
00	000/0	001/0	010/0	000/0
01	010/0	011/0	001/0	100/0
11	000/0	101/1	000/0	010/0
10	XXX/X	XXX/X	XXX/X	XXX/X

--

$$Q_2^{n+1} = Q_2^n \overline{Q_0^n} X + \overline{Q_2^n} Q_1^n Q_0^n \overline{X}$$

$$Q_1^{n+1} = \overline{Q_2^n} Q_1^n \overline{Q_0^n} + \overline{Q_1^n} Q_0^n X + Q_2^n Q_0^n \overline{X}$$

$$Q_0^{n+1} = \overline{Q_0^n} X + \overline{Q_2^n} Q_1^n X$$

$$Y = Q_2^n \overline{Q_0^n} X$$

--

$$J_2 = Q_1^n Q_0^n \overline{X} \qquad K_2 = \overline{\overline{Q_0^n} X}$$

$$J_1 = Q_0^n X + Q_2^n Q_0^n \overline{X} \qquad K_1 = \overline{\overline{Q_2^n}\,\overline{Q_0^n} + Q_2^n Q_0^n \overline{X}}$$

$$J_0 = X \qquad K_0 = \overline{\overline{Q_2^n} Q_1^n X}$$

-- The circuit design is capable of booting itself.

-- The logic diagram is omitted.