**Chapter Four: Combinational Logic Circuits**

1. (65)16 is equivalent to 8-bit ( ? )gray.

**Solution:**

65 🡪 0110 0101 (binary) 🡪 0 101 0111 (gray code)

2. **True or False:** The output of the circuit shown in Fig.1 is F = πABC ( 0, 1, 4 ).



Fig.1 芯片内部C为最高位。函数A为最高位。芯片为3-8译码器。

**Solution:** False.

F = ∑ABC (2, 3, 5,7) 🡪 minterms

F = πABC ( 0, 1, 4, 6 ) 🡪 maxterms

3. **True or False:** 74x283 is a 4-bit binary adder. Given S/A = 1, A3A2A1A0 = 1100, B3B2B1B0 = 1001, for the circuit shown in Fig.2, then S3S2S1S0 = ( 0111 ).

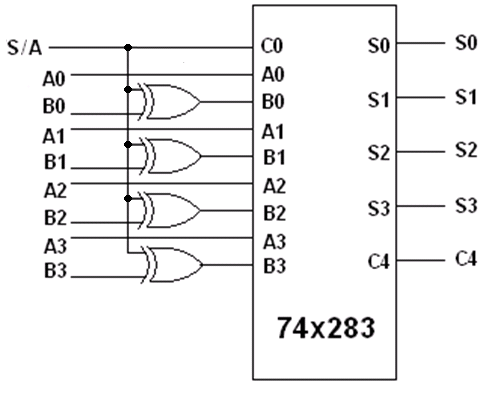


Fig.2 C0为进位输入信号。芯片为超前进位加法器。

**Solution:** False.

C0=1 (carry input)

B3=1⊕1=0, B2=0⊕1=1, B1=0⊕1=1, B0=1⊕1=0 (input to chip)

1100 (A3A2A1A0)

0110 (B3B2B1B0: input to chip)

+ 1

\_\_\_\_\_\_\_\_\_\_\_

10011 sum is 0011, carry output is 1.

S3S2S1S0 = ( 0011 )

4. **True or False:** 74x85 is a 4-bit magnitude comparator. If the input ABCD shown in Fig.3 is 0100, the output F is ( 1 ).



Fig.3 芯片为数值比较器。LT小于less than，EQ等于equal，GT大于greater than。

**Solution:** False.

A is equal to B on the lower bits.

A3A2A1A0 = 0110, B3B2B1B0 = ABCD = 0100

A3A2A1A0 > B3B2B1B0

LT=0, EQ=0, GT=1

F=0

**5.** A combination logic circuit is shown in Figure IV.

(1) **True or False:** Its logical expression is F = (A+C+D) + A’BCD + AB’C’D.

(2) **True or False:** The **canonical sum** of F(A, B, C, D) is F = ABC’D’ + A’B’C’D’ + A’BCD + AB’C’D.

(3) Please fill out the Karnaugh Map (Table IV) of F(A, B, C, D) . **Write the “1” blocks.**

(4) If the inputs (A B C D) are 8421-BCD codes, minimize the logic function F(A, B, C, D) into minimal **AND-OR** expression using Karnaugh Map (Table IV). **Write the “d” blocks**.

(5) Minimize the logic function F(A, B, C, D) into minimal hazard free **AND-OR** expression.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table IV | | | | |
| **CD**  **AB** | **00** | **01** | **11** | **10** |
| **00** |  |  |  |  |
| **01** |  |  |  |  |
| **11** |  |  |  |  |
| **10** |  |  |  |  |



**Solution:**

(1) False. The logical expression is F = (A+C+D)’ + A’BCD + AB’C’D

(2) False. The canonical sum is F = A’BC’D’ + A’B’C’D’ + A’BCD + AB’C’D

F = (A+C+D)’ + A’BCD + AB’C’D (from question 1)

= A’C’D’ + A’BCD + AB’C’D

= A’BC’D’ + A’B’C’D’+ A’BCD + AB’C’D

(3) Karnaugh Map of F(A, B, C, D): see “1” blocks in Table IV

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table IV | | | | |
| **CD**  **AB** | **00** | **01** | **11** | **10** |
| **00** | **1** |  |  |  |
| **01** | **1** |  | **1** |  |
| **11** | **d** | **d** | **d** | **d** |
| **10** |  | **1** | **d** | **d** |

(4) The inputs (A B C D) are 8421-BCD codes.

Karnaugh Map of F(A, B, C, D): see “d” blocks in Table IV

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table IV | | | | |
| **CD**  **AB** | **00** | **01** | **11** | **10** |
| **00** | **1** |  |  |  |
| **01** | **1** |  | **1** |  |
| **11** | **d** | **d** | **d** | **d** |
| **10** |  | **1** | **d** | **d** |

The minimal **AND-OR** expression is F = A’C’D’ + BCD+AD

(5) The minimal hazard free **AND-OR** expression is F = A’C’D’ + BCD+AD

6. **True or False.** The hazard in combinational logic is caused by（ **Minterms** ）.

**Solution:** False.

The delay of gates

7. **True or False.** The function of the circuit shown in the following figure is ( **one bit full adder** ).



**Solution:** False.

NAND gate, AND gate, NOR gate in the figure

F1=(XY)’X=(X’+Y’)X=XY’ (greater than, X > Y)

F3=(XY)’Y=(X’+Y’)Y=X’Y (less than, X <Y)

F2=(XY’+X’Y)’=(XY’)’(X’Y)’=(X’+Y)(X+Y’)=XY+X’Y’ (XNOR, equal to, X =Y)

One bit data comparator

8. **True or False.** To let the 7 outputs abcdefg=0011111 in the active high seven-segment LED, the inputs I3I2I1I0 should be ( 0011 ).



**Solution**: False. LEDs a and b are off, it is 6 in decimal. The binary is 0110.

9. To solve the hazards in the function, all redundant terms should be added are ( ? ).

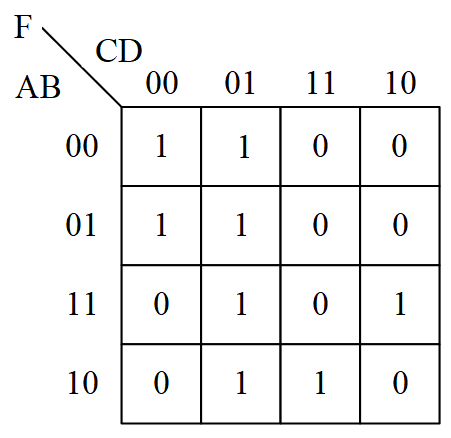
**Solution:**

(,,)

10. Combinational Logic Design.

Please use the 4-to-1 line data selector to implement the logic function F(A,B,C,D）= ∑m (0, 1, 4, 5, 9, 11, 13, 14). Draw the Karnaugh map of the logic function, write out the function for each data input, and draw the circuit diagram.

**Solution:**





**D0**=,**D1**=,**D2**=*D*,**D3**=