# §1-2

1. Convert the following numbers with the indicated bases to decimal:

(a) (EX) $(132.3)_4$

(b) (HW) $(425.3)_6$



# §1-2

2. In each of the following cases, determine the radix ***r***:

(a) (EX) $(24)_r = (18)_{10}$

(b) (HW) $(231)_r = (91)_{10}$

# §1-2

3. Perform the following binary arithmetic operations:

(a) (EX)

i. $1011 + 0110$ ii. $110 \times 011$

(b) (HW)

i. $1011001 + 1011010$ ii. $01101 \times 11011$



# §1-3 & 1-4

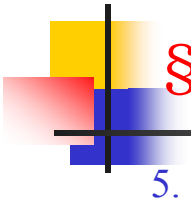
4. Convert the following numbers from the given base to the other three bases listed in the table.

(a) (EX)

| Decimal | Binary | Octal | Hexadecimal |
|---|---|---|---|
| 19.5 | ? | ? | ? |
| ? | ? | 63.75 | ? |

(b) (HW)

| Decimal | Binary | Octal | Hexadecimal |
|---|---|---|---|
| 237.875 | ? | ? | ? |
| ? | ? | 156.375 | ? |

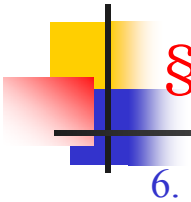


# §1-5

5. Obtain the 1's and 2's complements of the following 8-bit binary numbers:
   (a) (EX) 01000110, 11011010

   (b) (HW) 01010011, 10011000

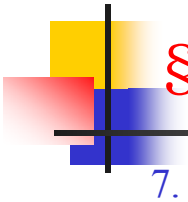




# §1-5

6. (a) (EX)
   i. Perform the indicated subtraction with the following unsigned binary numbers by **1's** complement addition:
      11101 – 10001 00101 – 10100
   ii. Perform the indicated subtraction with the following unsigned binary numbers by **2's** complement addition:
      11101 – 10001 00101 – 10100

   (b) (HW) Repeat (a) for
   i. 10110101 – 01001101, 01001101 – 10110101
   ii. 10110101 – 01001101, 01001101 – 10110101



# §1-6

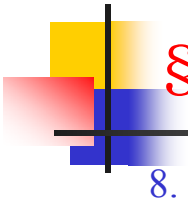
7. (a) (EX) The following binary numbers have a sign in the leftmost position and, if negative, are in 2's complement form. Perform the indicated arithmetic operations and indicate if ***overflow*** occurs for each computation. (Hint: Perform subtraction by 2's complement addition.)

   i. 001011 + 100110 ii. 110001 – 010010

   (b) (HW) Repeat (a) for

   i. 10110101 – 01001101 ii. 01001101 + 10110101





# §1-7

8. (a) (EX)

   Represent the decimal numbers 25 and 87 in BCD, and then show the steps necessary to form their sum.

   (b) (HW)

   Represent the decimal numbers 376 and 843 in BCD, and then show the steps necessary to form their sum.

# §2-4

9. Demonstrate the validity of the following identities by means of
   i. truth tables
   ii. postulates or proven theorems
   iii. Venn diagrams

   (a) (EX) $XY + XZ + X'YZ' = XZ + YZ'$

   (b) (HW) $XYZ' + (X' + Z)(Y + Z') = X'Z' + Y$



## (a) $XY + XZ + X'YZ' = XZ + YZ'$

### i. truth tables

| X Y Z | XY | XZ | X'YZ' | XZ + YZ' | | |
|---|---|---|---|---|---|---|
| 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 0 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 1 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 1 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 0 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 0 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 1 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 1 1 | 1 | 1 | 0 | 0 | 1 | 1 |

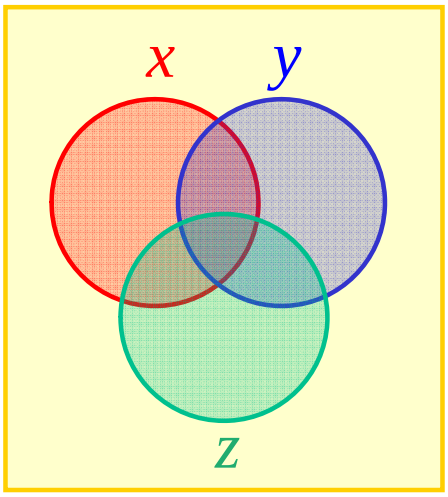
(a) $XY + XZ + X'YZ' = XZ + YZ'$

ii. postulates or proven theorems

$XY + XZ + X'YZ'$

$=$

iii. Venn diagram






## §2-5

10. Draw the logic diagram for each of the following Boolean equations.

(a) (EX) $F(X, Y, Z) = XY + XZ + X'YZ'$

$= XZ + YZ'$

(b) (HW) $F(X, Y, Z) = XYZ' + (X' + Z)(Y + Z')$

$= X'Z' + Y$

# §2-5

* Give an example (which may be designed as a combinational circuit) by yourself. Describe the problem, define the input and output variables, derive the Boolean equation(s), and simplify the Boolean equation(s) of the example.



# §2-5

11. Simplify the following Boolean expressions to a minimum number of literals according to the identities of Boolean algebra. Write the particular identities used in each step. For the simplified function you derive, how many literals does it have?

(a) (EX) $F(X, Y, Z) = XY + XZ + X'YZ'$

(b) (HW) $F(X, Y, Z) = XYZ' + (X' + Z)(Y + Z')$

(c) (HW) $F(A, B, C, D) = A'BCD + A'BCD' + A'C'D + AB'D + ACD' + ABC + (A + B' + C + D)'$

| Postulate/Theorem | (a) | (b) |
|---|---|---|
| Postulate 2 | $x + 0 = x$ | $x \cdot 1 = x$ |
| Postulate 5 | $x + x' = 1$ | $x \cdot x' = 0$ |
| Theorem 1 | $x + x = x$ | $x \cdot x = x$ |
| Theorem 2 | $x + 1 = 1$ | $x \cdot 0 = 0$ |
| Theorem 3, involution | $(x')' = x$ | |
| Postulate 3, commutative | $x + y = y + x$ | $xy = yx$ |
| Theorem 4, associative | $x + (y + z) = (x + y) + z$ | $x(yz) = (xy)z$ |
| Postulate 4, distributive | $x(y + z) = xy + xz$ | $x + (yz) = (x + y)(x + z)$ |
| Theorem 5, DeMorgan | $(x + y)' = x'y'$ | $(xy)' = x' + y'$ |
| Theorem 6, absorption | $x + xy = x$ | $x(x + y) = x$ |
| Theorem *, consensus | $xy + x'z + yz = xy + x'z$ | $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$ |

# §2-5

12. Complement the following functions by (i) applying DeMorgan's theorem and (ii) by using duals:

(a) (EX) XY + XZ + X′YZ′

(b) (HW) XYZ′ + (X′ + Z) (Y + Z′)



# §2-6

13. Obtain the truth table of the following functions, and express each function in sum-of-minterms and product-of-maxterms form:

(a) (EX) XY + XZ + X′YZ′

(b) (HW) XYZ′ + (X′ + Z) (Y + Z′)

---

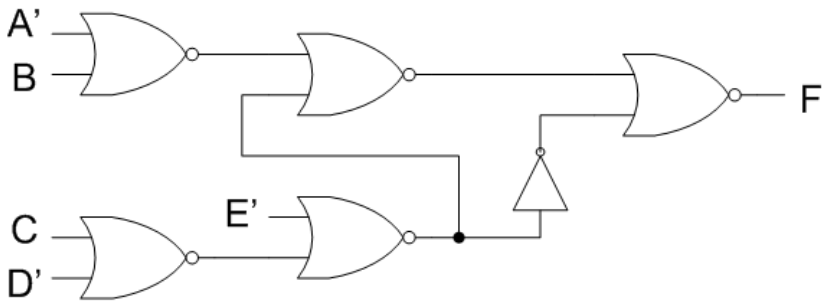
(a)

| *X Y Z* | *F* |
|---|---|
| 0 0 0 | |
| 0 0 1 | |
| 0 1 0 | |
| 0 1 1 | |
| 1 0 0 | |
| 1 0 1 | |
| 1 1 0 | |
| 1 1 1 | |

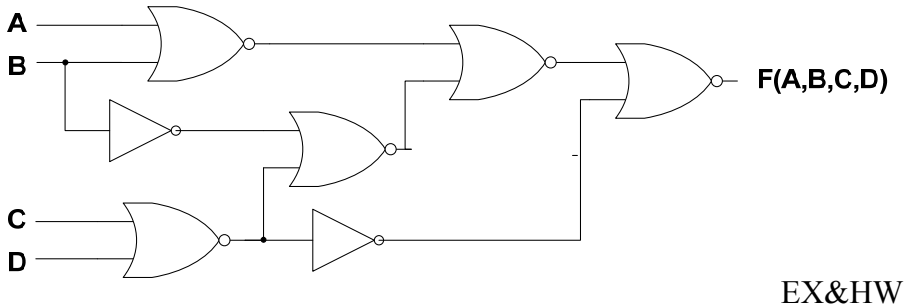
# Technology Parameters

14. Assume that a NOR gate has propagation delay $t_{pd\_NOR} = 0.078$ ns and an inverter has a propagation delay $t_{pd\_inv} = 0.052$ ns. What is the propagation delay of the longest path through the following circuit.

(a) (EX)




(b) (HW)






# Technology Parameters

15. An integrated circuit logic family has NAND gates with a fan-out of 8 standard loads and buffers with a fan-out of 16 standard loads. Sketch a schematic showing how the output signal of a single NAND gate can be applied to *N* other gate inputs using as few buffers as possible. Assume that each input is one standard load.

(a) (EX) N = 38

(b) (HW) N = 53

## §3-2

16. Simplify the following Boolean function by using Karnaugh maps (K-maps), and calculate the gate input counts of *F* before and after the optimization:

(a) (EX) i. $F(X, Y, Z) = \Sigma m(1, 3, 4, 5, 6, 7)$

ii. $F(X, Y, Z) = XY + XZ + X'YZ'$

(b) (HW) $F(X, Y, Z) = XYZ' + (X' + Z)(Y + Z')$

---

(a)

| X \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

| X \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |



## §3-3

17. Optimize the following Boolean function into sum-of-products by using K-map. Find all prime implicants and essential prime implicants, and apply the selection rule:

(a) (EX) $F(A, B, C, D) = \Sigma m(1,5,6,7,11,12,13,14,15)$

(b) (HW) $F(A, B, C, D) = A'BCD + A'BCD' + A'C'D + AB'D + ACD' + ABC + (A + B' + C + D)'$

---

(a)

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

# §3-4

18. Optimize the following function into (i) sum-of-products and (ii) product-of-sums forms by using K-map, and calculate the gate input count for each form:

(a) (EX) $F(A, B, C, D) = \Sigma m(1,5,6,7,11,12,13,14,15)$

(b) (HW) $F(A, B, C, D) = A'BCD + A'BCD' + A'C'D + AB'D + ACD' + ABC + (A + B' + C + D)'$

---

(a)

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |



# §3-5

19. Optimize the following Boolean function *F* together with don't-care condition *d* into (i) sum-of-products and (ii) product-of-sums forms by using K-map. Express each simplified function in "Σ*m*" notation, and determine the gate input count of it.

(a) (EX) $F(W, X, Y, Z) = \Sigma m(5,7,11,14,15)$,
$d(W, X, Y, Z) = \Sigma m(1,6,12,13)$

(b) (HW) $F(W, X, Y, Z) = \Sigma m(0,2,5,8,14,15)$,
$d(W, X, Y, Z) = \Sigma m(4,7,10,13)$

---

(a)

| WX \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

| WX \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

# Quine-McCluskey Method

20. Optimize the following Boolean function *F* together with don't-care condition *d* by using Quine-McCluskey method.

(a) (EX) $F(X, Y, Z) = \Sigma m(1,5,6),\ d(X, Y, Z) = \Sigma m(2,7)$

(b) (HW) i. $F(W, X, Y, Z) = \Sigma m(5,7,11,14,15),$
$d(W, X, Y, Z) = \Sigma m(1,6,12,13)$

ii. $F(W, X, Y, Z) = \Sigma m(0,2,5,8,14,15),$
$d(W, X, Y, Z) = \Sigma m(4,7,10,13)$

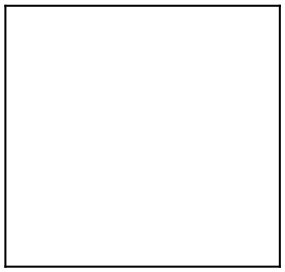
---

(a)

**Implication Table**

| Column 1 | Column 2 | Column 3 |
|---|---|---|
| | | |

**Implication Chart**





# Multiple-Level Circuit Optimization

21. Use decomposition to find a minimum gate input count, multiple-level implementation for each of the following functions using AND and OR gates and inverters. Draw the logic diagram and calculate the gate input counts of *F* before and after the decomposition.

(a) (EX) $F(A, B, C, D) = ABC' + A'BC + A'CD + AC'D$

(b) (HW) $F(A, B, C, D) = AC + BC + A'BD + AB'D$

# §3-6

22. Simplify the Boolean function and implement it

(a) (EX) F(A, B, C) = Σm(1, 3, 4, 5, 6)

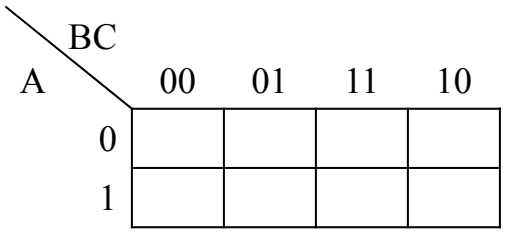
i. by 2-level NAND gates ii. by 2-level NOR gates

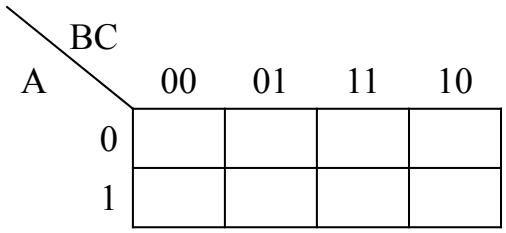
(b) (HW) F(A, B, C, D) = Σm(1,5,6,7,11,12,13,14,15)

i. by 2-level NAND gates ii. by 2-level NOR gates

---

(a)






# §3-7

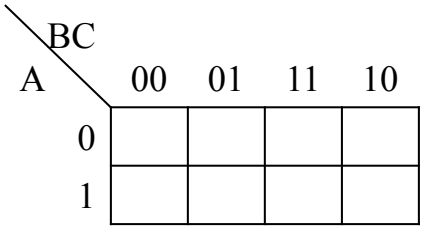
23. Simplify the following functions with AND-NOR, NAND-AND, OR-NAND, NOR-OR 2-level forms:

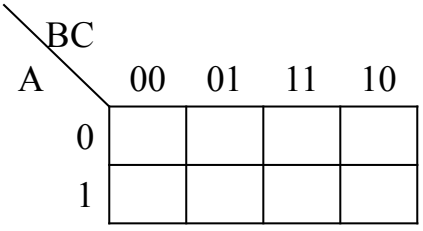
(a) (EX) F(A, B, C) = Σm(1, 3, 4, 5, 6)

(b) (HW) F(A, B, C, D) = Σm(1,5,6,7,11,12,13,14,15)

---

(a)



## §3-8

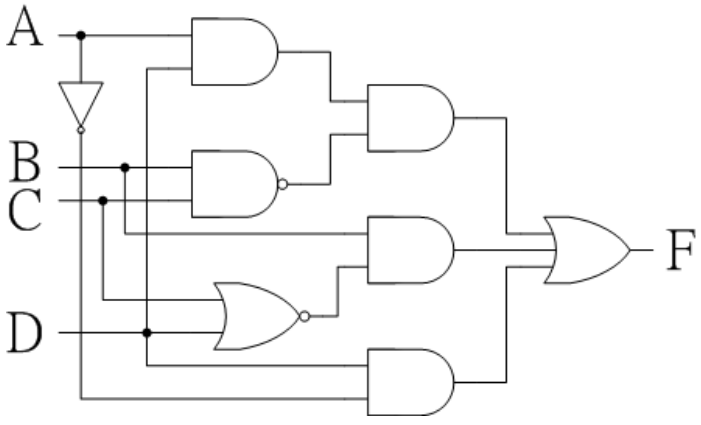
24. Consider a 3-bit message to be transmitted together w/ an odd parity bit. Design the odd-parity generation function *P* for the transmitter and the odd-parity checking function *C* for the receiver. Assume that the output of the checker is equal to 1 if error occurs.



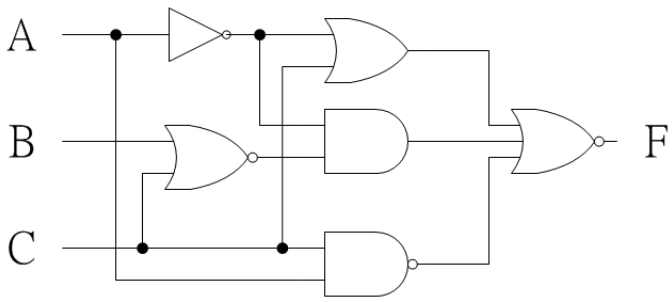
## §4-3

25. Analyze the following circuit to obtain the simplified Boolean expression in sum-of-products form for output *F* in terms of the input variables:

(a) (EX)




(b) (HW)



# §4-4

26. Design a combinational circuit by the following steps:
    i. Develop the truth table of the circuit.
    ii. Apply two-level optimization. Derive both the sum-of products (SoP) and product-of-sums (PoS) forms and calculate the gate input count for each form.
    iii. Repeat ii by applying multiple-level optimization.

(a) (EX) The combinational circuit accepts a 4-bit binary number (*w x y z*) and generate an output (*f*) which is equal to 1 only if the input is greater than twelve or less than three.

(b) (HW) The combinational circuit accepts a two 2-bit binary numbers $A_1A_0$ and $B_1B_0$ and generate the product ($P_3P_2P_1P_0$) of them.



(a)

| *w* | *x* | *y* | *z* | *f* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 1 | 1 | |

The combinational circuit accepts a 4-bit binary number (w x y z) and generate an output (f) which is equal to 1 only if the input is greater than twelve or less than three.

| *wx* \ *yz* | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

| *wx* \ *yz* | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

# §4-5

27. (EX&HW)

(a) Design a half subtractor (HS) which has 2 input variables and 2 output variables. The input variables are the minuend (被減數) $X_i$ and the subtrahend (減數) $Y_i$, and the output variables produce the difference $D_i$ and the borrow-out $B_{i+1}$ .

i. List the truth table of the circuit.

ii. Derive the simplified output equations in sum-of-products form.

iii. Map the equations into exclusive-OR (XOR) operations and other basic gates, if possible.

---

(a)

| Xi | Yi | Bi+1 | Di |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |



# §4-5

27. (EX&HW)

(b) Design a full subtractor (FS) which has 3 input variables and 2 output variables. The input variables are the minuend (被減數) $X_i$, the subtrahend (減數) $Y_i$, and the borrow-in $B_i$, and the output variables produce the difference $D_i$ and the borrow-out $B_{i+1}$ .

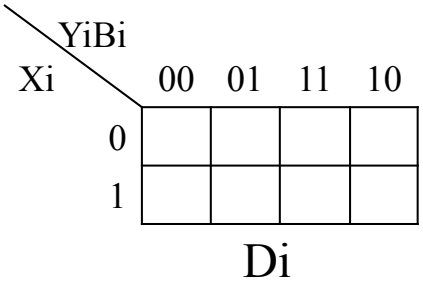
i. List the truth table of the circuit.

ii. Derive the simplified output equations in sum-of-products form.

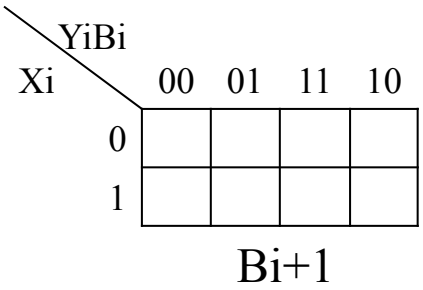
iii. Map the equations into exclusive-OR (XOR) operations and other basic gates, if possible.

(b)

| Xi | Yi | Bi | Bi+1 | Di |
|---|---|---|---|---|
| 0 | 0 | 0 | | |
| 0 | 0 | 1 | | |
| 0 | 1 | 0 | | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |




Di




Bi+1



# §4-5

27. (EX&HW)

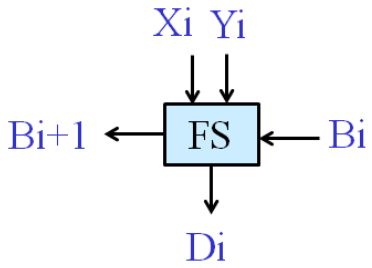
(c) Design a 4-bit ripple-borrow subtractor (RBS).

i. Draw the block diagram of the 4-bit RBS using the full subtractor designed in (b) as a basic block.

ii. Assume that AND and OR gates have a propagation delay of 10*ns*, XOR gate has a propagation delay of 20*ns*, and the propagation delay of an inverter is ignored. What is the ideal total propagation delay time of an *n*-bit RBS in *ns*.

---

(c)



# §4-5

27. (EX&HW)

(d) Design a 4-bit borrow-look-ahead (BLA) subtractor.

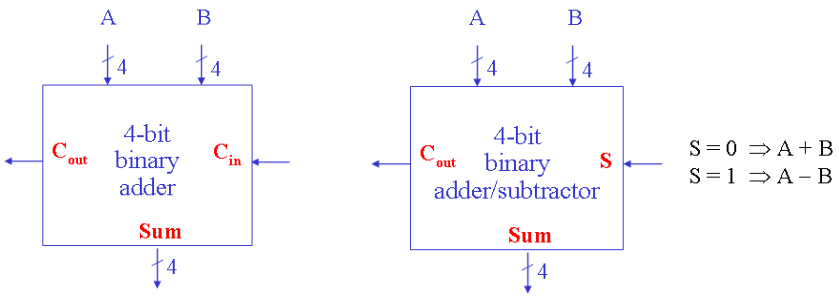
i. Derive the equation for each borrow generate $G_i$, borrow propagate $P_i$, output borrow $B_{i+1}$, and difference bit $D_i$ , where $i = 0, 1, 2, 3$.

ii. Assume that the propagation delays of XOR, AND, OR, and NOT gates are the same as that describe in (c). What is the ideal total propagation delay time of an *n*-bit single-level BLA subtractor in *ns*? Why?



# §4-6

28. (EX& HW) Given the excess-3 code as follows, design an excess-3 adder that adds two decimal digits in excess-3 code and a carry-in bit, and generates a carry out bit $C_{out}$ and excess-3 coded sum bits $S_3S_2S_1S_0$. First, add these two digits and carry in bit as binary numbers. If its carry out $K_{out}$ is equal to 1, then add 3 to its sum $Z_3Z_2Z_1Z_0$; otherwise, subtract 3 from the sum for correct results. Explain why the algorithm described above is correct and draw the block diagram of the excess-3 adder. Use binary adder and binary adder-subtractor as basic building blocks.




| Decimal digits | Excess-3 code |
|---|---|
| 0 | 0011 |
| 1 | 0100 |
| 2 | 0101 |
| 3 | 0110 |
| 4 | 0111 |
| 5 | 1000 |
| 6 | 1001 |
| 7 | 1010 |
| 8 | 1011 |
| 9 | 1100 |

<Ans>

| K | Z3 | Z2 | Z1 | Z0 | C | S3 | S2 | S1 | S0 | Sum |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | 0 |
| | | | | | | | | | | 1 |
| | | | | | | | | | | 2 |
| | | | | | | | | | | 3 |
| | | | | | | | | | | 4 |
| | | | | | | | | | | 5 |
| | | | | | | | | | | 6 |
| | | | | | | | | | | 7 |
| | | | | | | | | | | 8 |
| | | | | | | | | | | 9 |
| | | | | | | | | | | 10 |
| | | | | | | | | | | 11 |
| | | | | | | | | | | 12 |
| | | | | | | | | | | 13 |
| | | | | | | | | | | 14 |
| | | | | | | | | | | 15 |
| | | | | | | | | | | 16 |
| | | | | | | | | | | 17 |
| | | | | | | | | | | 18 |
| | | | | | | | | | | 19 |



# §4-9

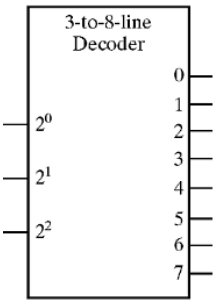
29. (a) (EX) A combinational circuit is defined by the following Boolean functions:

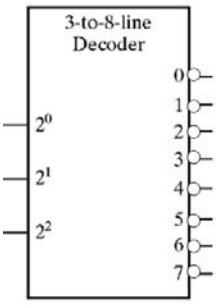
$f1(x, y, z) = \Sigma m(1, 4, 5)$

$f2(x, y, z) = \Sigma m(0, 2, 3, 4, 6, 7)$

i. Design the circuit using a 3-to-8-line decoder with ***active HIGH*** outputs and external gates. Minimize the number of inputs of each gate.

ii. Design the circuit using a 3-to-8-line decoder with ***active LOW*** outputs and external gates. Minimize the number of inputs of each gate.




(b) (HW) Repeat (a) for the following functions:

$f1(x, y, z) = \Sigma m(0, 2, 3, 5, 7)$

$f2(x, y, z) = \Sigma m(1, 2, 4)$

# §4-10

30. (a) (EX) Design a 4-input priority encoder with inputs D3, D2, D1, and D0, and outputs A1, A0, and V. Assume that input D0 has the highest priority and input D3 has the lowest priority.

i. Derive the truth table of the priority encoder.

ii. Derive the Boolean expression for each output.

(b) (HW) Repeat (a) for a hepta (7)-binary priority encoder. Assume that 6 has the highest priority and 0 has the lowest priority.

---

(a)

| D0 | D1 | D2 | D3 | A1 | A0 | V |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | | |
| 1 | x | x | x | | | |
| 0 | 1 | x | x | | | |
| 0 | 0 | 1 | x | | | |
| 0 | 0 | 0 | 1 | | | |

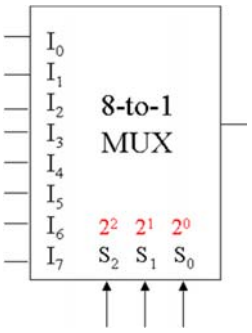


# §4-11

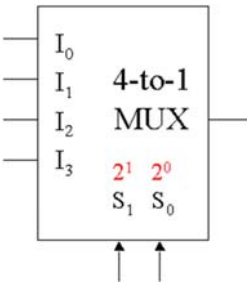
31. (a) (EX) Given the following function, Implement the function *f* and draw the block diagram using

$f(w, x, y, z) = \Sigma m(3, 4, 6, 7, 8, 9, 10, 11, 13, 14)$

i. a 8×1 multiplexer and external gates, if necessary. Connect inputs *w*, *x*, and *y* to the MUX selection lines. Show the truth table for deriving the inputs of the MUX.




ii. a 4×1 multiplexer and external gates, if necessary. Connect *w* and *x* to the MUX selection line. Show the truth table for deriving the inputs of the MUX.

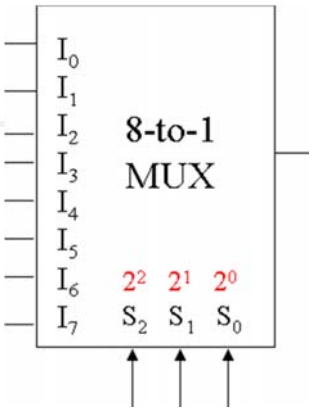


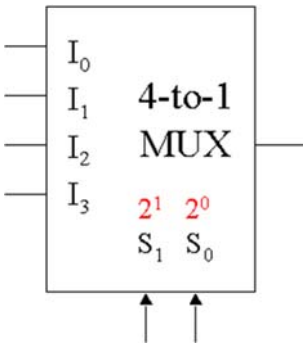

(b) (HW) Repeat (a) for the following function:

$f(w, x, y, z) = \Sigma m(2, 5, 6, 8, 9, 12, 13, 14)$

(a) $f(w, x, y, z) = \Sigma m(3, 4, 6, 7, 8, 9, 10, 11, 13, 14)$

| *w* | *x* | *y* | *z* | *f* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 1 | |

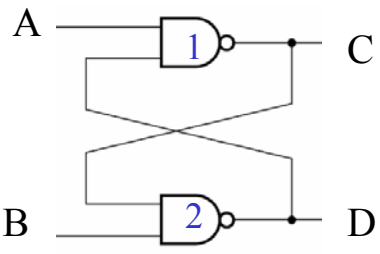


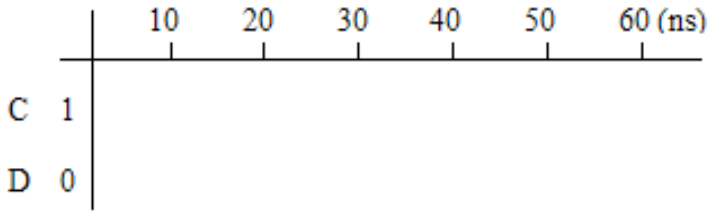






# §5-3

32. Given two cross-couple NAND gates, assume that the propagation delay (*PD*) of each gate is 10 ns. Draw the timing waveform for outputs C and D, and estimate the time required for the outputs to become stable in the following conditions:



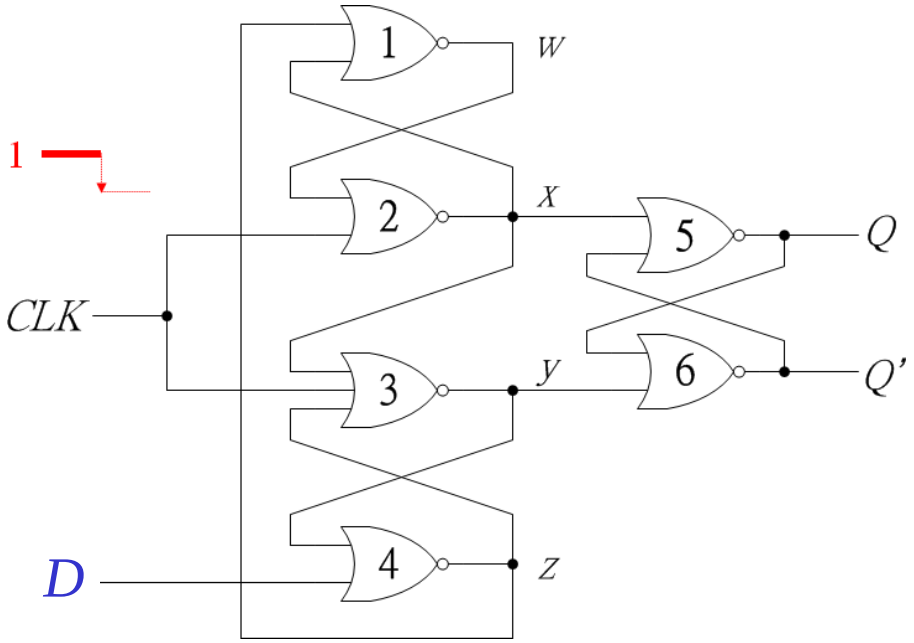

(a) (EX) A = 1, B = 0, C = 1, and D = 0

(b) (HW) i. A = 0, B = 1, C = 0, and D = 1

ii. A = 0, B = 0, C = 0, and D = 1

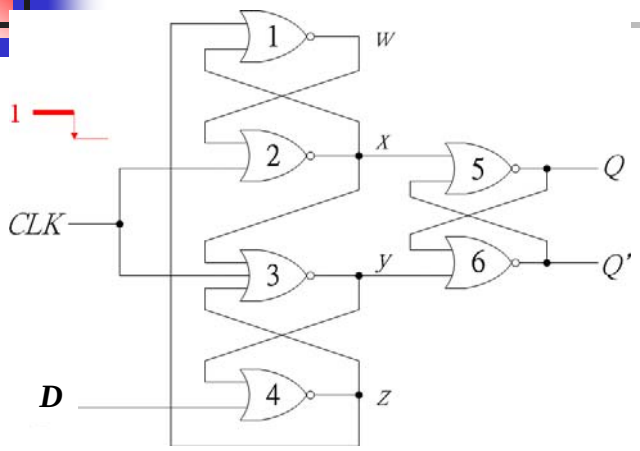
iii. A = 1, B = 1, C = 1, and D = 1

33. (EX & HW) Given the following logic diagram, explain why it is a negative-edge-triggered D flip-flop.

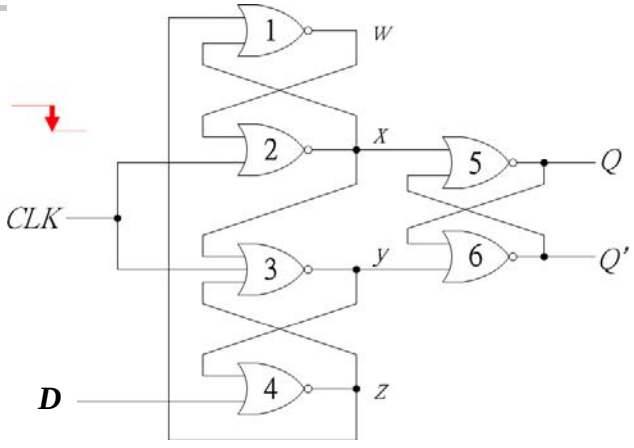



<Ans>

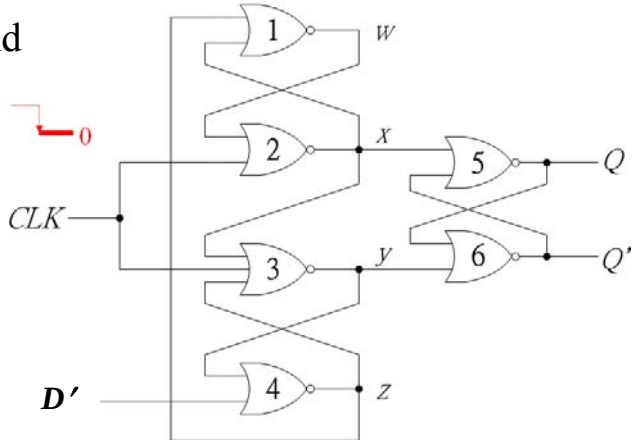
i. When clock is HIGH:




ii. When clock goes HIGH-to-LOW:




iii. When clock is LOW and $D$ is changed to $D'$ :



# §5-5

34. (EX & HW) A synchronous sequential circuit with two D flip-flops *A* and *B*, two inputs *X* and *Y*, and one output *Z* is specified by the following equations:

$$D_A = \overline{X}A + XY$$
$$D_B = \overline{X}A + XB$$
$$Z = A\overline{B}$$

(a) Draw the logic diagram of the circuit.

(b) Derive the next state equations.

(c) Complete the transition/state table of the circuit.

(d) Draw the state diagram.

(e) Starting from state 00 in the state diagram, determine the state transitions and output sequence that will be generated when an input sequence of XY = 01, 11, 11, 00, 10, 01, 11, 10 is applied.

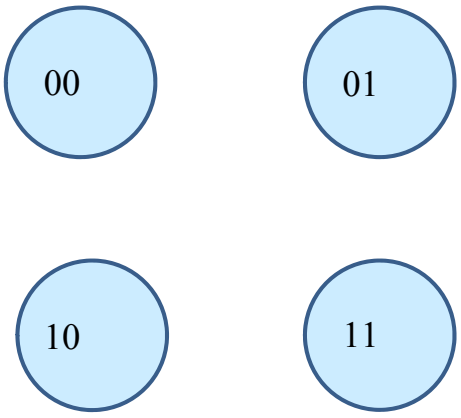


<Ans>

Transition/state table:

| Present state | | Input | | Next state | | Output |
|---|---|---|---|---|---|---|
| A | B | X | Y | A+ | B+ | Z |
| 0 | 0 | 0 | 0 | | | |
| 0 | 0 | 0 | 1 | | | |
| 0 | 0 | 1 | 0 | | | |
| 0 | 0 | 1 | 1 | | | |
| 0 | 1 | 0 | 0 | | | |
| 0 | 1 | 0 | 1 | | | |
| 0 | 1 | 1 | 0 | | | |
| 0 | 1 | 1 | 1 | | | |
| 1 | 0 | 0 | 0 | | | |
| 1 | 0 | 0 | 1 | | | |
| 1 | 0 | 1 | 0 | | | |
| 1 | 0 | 1 | 1 | | | |
| 1 | 1 | 0 | 0 | | | |
| 1 | 1 | 0 | 1 | | | |
| 1 | 1 | 1 | 0 | | | |
| 1 | 1 | 1 | 1 | | | |

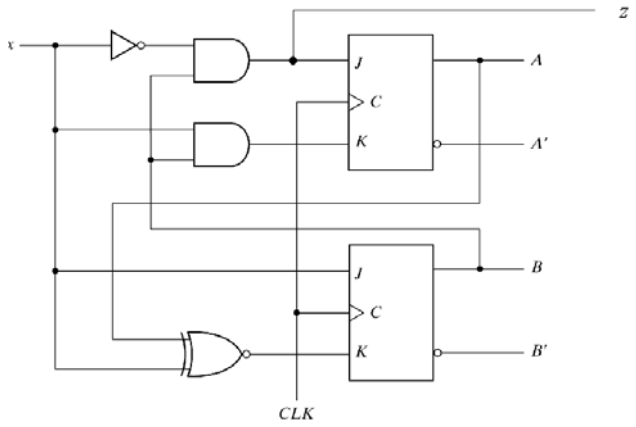
State diagram:




| Cycle | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| *xy* | 01 | 11 | 11 | 00 | 10 | 01 | 11 | 10 | |
| AB | 00 | | | | | | | | |
| Z | | | | | | | | | |

# §5-5

35. (EX & HW) Analyze the following synchronous sequential circuit, which has one input *x* and one output *z*, step by step:




(a) Derive the memory input equations and the output equation.

(b) Derive the next state equations.

(c) Complete the transition/state table of the circuit.

(d) Draw the state diagram.

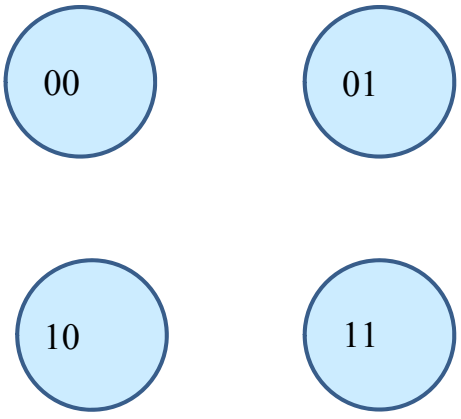


<Ans>

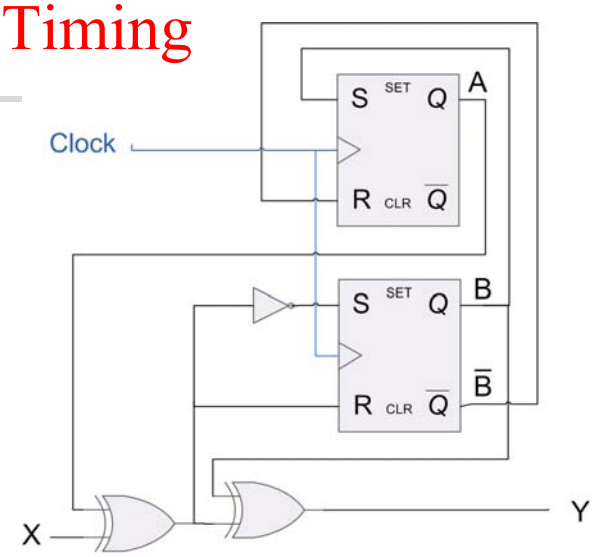
Transition/state table:

| Present state | | Input | Next state | | Output |
|---|---|---|---|---|---|
| A | B | X | A+ | B+ | 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 | | | |

State diagram:



# 補充資料：Sync Seq Ckt Timing




36. (EX & HW) A sequential circuit is given in the following figure. The timing parameters for the gates and flip-flops are as follows:

Inverter: $t_{pd} = 0.5$ ns

XOR Gate: $t_{pd} = 2.0$ ns

Flip-flop: $t_{pd} = 2.0$ ns,

$t_s = 1.0$ ns, and

$t_h = 0.25$ ns

(a) Find the longest path delay from an external circuit input passing through gates only to an external circuit output.

(b) Find the longest path delay in the circuit from an external input to positive clock edge.

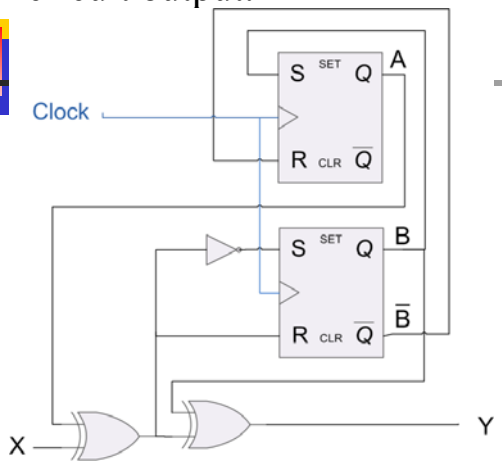
(c) Find the longest path delay from positive clock edge to output.

(d) Find the longest path delay from positive clock edge to positive clock edge.

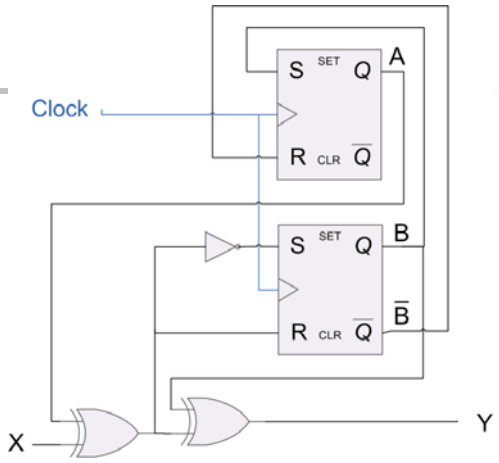
(e) Determine the maximum frequency of operation of the circuit in megahertz (MHz).



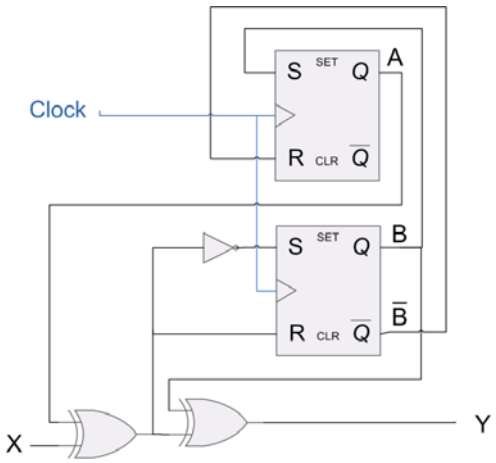
(a) external circuit input → external circuit output:



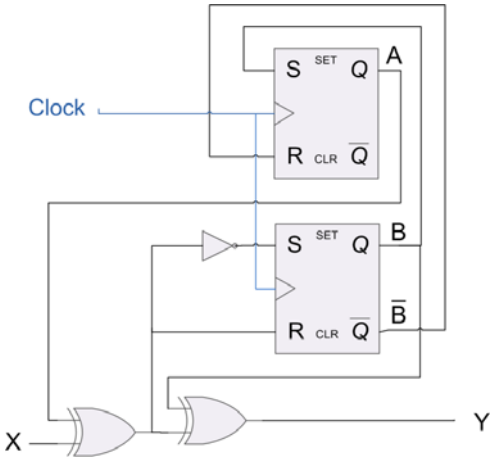

(b) external input → positive clock edge:




(c) positive clock edge → output:




(d) positive clock edge → positive clock edge:



# §5-7 & 5-8

37. (EX & HW) Design a Moore-type recognizer which has one input (X) and one output (Z). The output is asserted (= 1) whenever the input sequence …010… has been observed, as long as the sequence …100… has not been seen since the last reset.

(a) Draw the state diagram of the recognizer.

(b) Starting from the initial state of your design, determine the state transitions and output sequence that will be generated when an input sequence of 00111010100101 is applied.

---

<Ans>

| Cycle | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | |
| State | A | | | | | | | | | | | | | | |
| Output | | | | | | | | | | | | | | | |

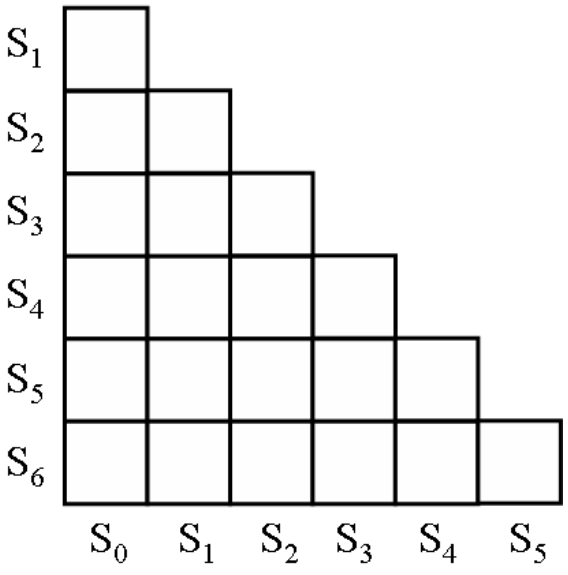
# 補充資料：State Reduction

38. (EX & HW) Given the following state table,

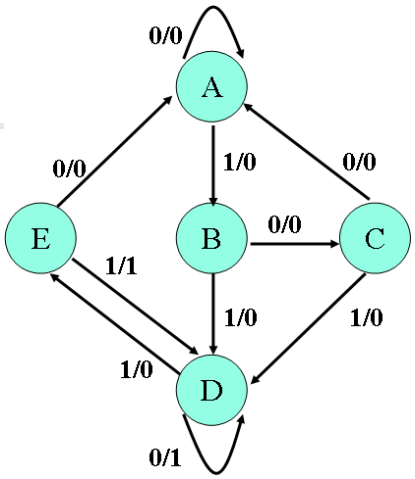
(a) Use the row matching method to find the equivalent states.

(b) Use the implication chart method to find the equivalent states. Tabulate the reduced state table.

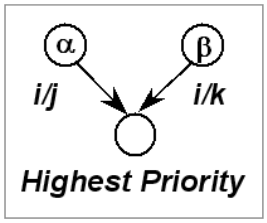
| Present state | Next state | | Output | |
|---|---|---|---|---|
| | x = 0 | x = 1 | x = 0 | x = 1 |
| $S_0$ | $S_1$ | $S_4$ | 0 | 0 |
| $S_1$ | $S_1$ | $S_2$ | 0 | 0 |
| $S_2$ | $S_1$ | $S_6$ | 0 | 0 |
| $S_3$ | $S_1$ | $S_3$ | 0 | 0 |
| $S_4$ | $S_5$ | $S_4$ | 0 | 0 |
| $S_5$ | $S_5$ | $S_2$ | 0 | 0 |
| $S_6$ | $S_5$ | $S_3$ | 0 | 1 |

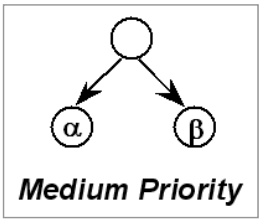


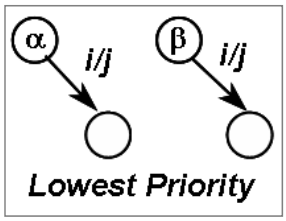
# 補充資料：State Assignment




39. (EX & HW) Given a state diagram and the illustration of the three guidelines for state assignment,
   (a) List the adjacent states for each of the guidelines.
   (b) Assign states into a state map according to the guidelines and check on the fulfilled adjacent conditions.




State map:

| $Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |



# §5-7 & 5-8

40. (EX & HW) Given the following state (transition) table, treat the unused state as don't-care conditions.

   Implement the circuit with **D** flip-flops. Derive the minimized Boolean equations in sum-of-products form for the flip-flop inputs and the output Z .

| Present state | | Input | Next state | | Output |
|---|---|---|---|---|---|
| $Q_A$ | $Q_B$ | X | $Q_A$+ | $Q_B$+ | Z |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |

<Ans>

| A \ Bx | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

| A \ Bx | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

| A \ Bx | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

# §5-7 & 5-8

41. (EX & HW) Given the following state (transition) table, treat the unused state as don't-care conditions. .

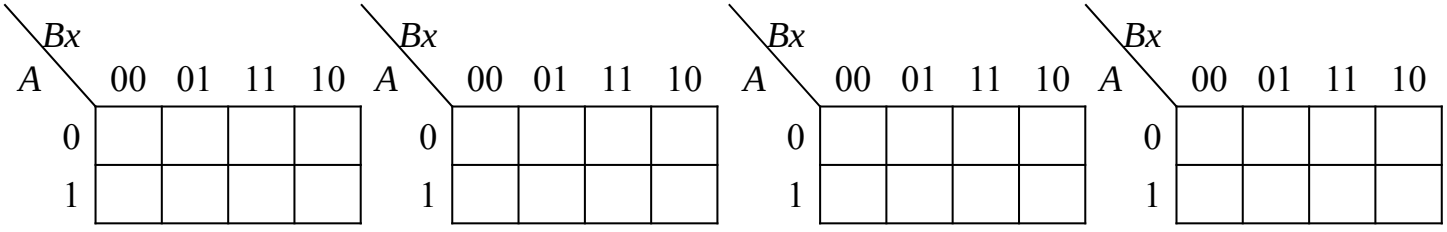
Implement the circuit with **JK** flip-flops. Complete the excitation table shown above and derive the minimized Boolean equations in sum-of-products form for the flip-flop inputs .

| Present state | | Input | Next state | | Output | Excitation Table Flip-Flop Inputs | | | |
|---|---|---|---|---|---|---|---|---|---|
| $Q_A$ | $Q_B$ | X | $Q_A$+ | $Q_B$+ | Z | $J_A$ | $K_A$ | $J_B$ | $K_B$ |
| 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 1 | 0 | 1 | 0 | | | | |
| 0 | 1 | 0 | 1 | 0 | 0 | | | | |
| 0 | 1 | 1 | 1 | 0 | 1 | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | | | | |
| 1 | 0 | 1 | 0 | 1 | 0 | | | | |



<Ans>

| Present state | | Input | Next state | | Output | Excitation Table Flip-Flop Inputs | | | |
|---|---|---|---|---|---|---|---|---|---|
| $Q_A$ | $Q_B$ | X | $Q_A$+ | $Q_B$+ | Z | $J_A$ | $K_A$ | $J_B$ | $K_B$ |
| 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 1 | 0 | 1 | 0 | | | | |
| 0 | 1 | 0 | 1 | 0 | 0 | | | | |
| 0 | 1 | 1 | 1 | 0 | 1 | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | | | | |
| 1 | 0 | 1 | 0 | 1 | 0 | | | | |



# 補充資料：Word Problems

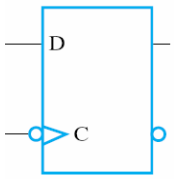
* (EX & HW) Give an example of synchronous sequential circuits by yourself. Describe the problem, define the input variables, output variables, and states, and draw the state diagram of the problem.



# §6-2

42. (EX & HW) Given the function table of a 2-bit shift register, derive the input equation for each D flip-flop of the register and draw the logic diagram of the register.

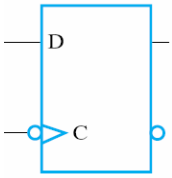
| CLK | $S_1$ | $S_0$ | $A_1^+$ | $A_0^+$ | Register Operation |
|---|---|---|---|---|---|
| ↓ | **0** | **0** | $\mathbf{A_1}$ | $\mathbf{A_0}$ | **No Change** |
| ↓ | **0** | **1** | $\mathbf{I_1}$ | $\mathbf{I_0}$ | **Parallel load** |
| ↓ | **1** | **0** | $\mathbf{A_0}$ | **SI** | **Shift left** |



# §6-3

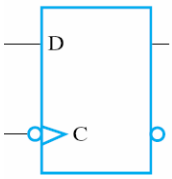
43. (EX & HW) Given the counting sequence of a 2-bit binary counter, design a **ripple** counter using negative-edge triggered D flip-flops. Derive and draw your logic circuit.

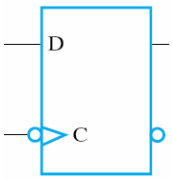
| $Q_1$ | $Q_0$ |
|---|---|
| 1 | 1 |
| 1 | 0 |
| 0 | 1 |
| 0 | 0 |




---

<Ans>



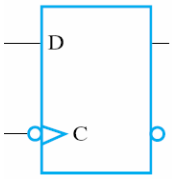






# §6-4

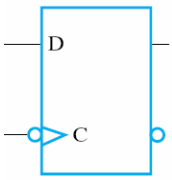
44. (EX & HW) Given the counting sequence of a 2-bit binary counter, design a **synchronous** counter using D flip-flops. Derive the flip-flop input equations and draw the logic diagram of the circuit.

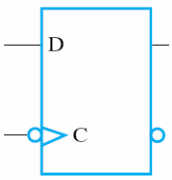
| $Q_1$ | $Q_0$ |
|---|---|
| 1 | 1 |
| 1 | 0 |
| 0 | 1 |
| 0 | 0 |




---

<Ans>

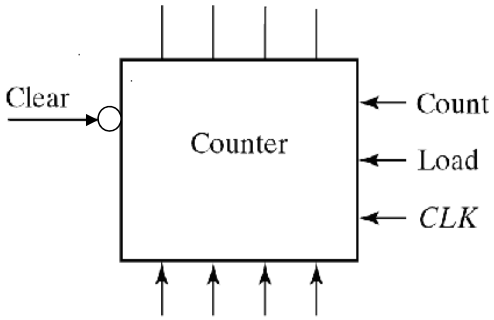


# §6-4

45. (EX & HW) Given the function table and the block diagram of a 4-bit binary up-counter, implement a counter that follows the sequence 0101 through 1011 and then repeats using the *load* input.

| Clear | CLK | Load | Count | Function |
|---|---|---|---|---|
| **0** | × | × | × | **Clear to 0** |
| **1** | ↑ | **1** | × | **Load inputs** |
| **1** | ↑ | **0** | **1** | **Count next binary state** |
| **1** | ↑ | **0** | **0** | **No change** |



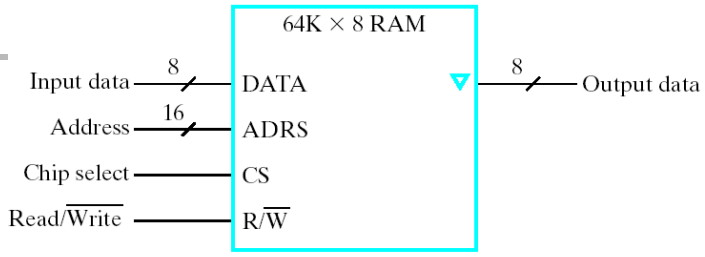



# §7-3

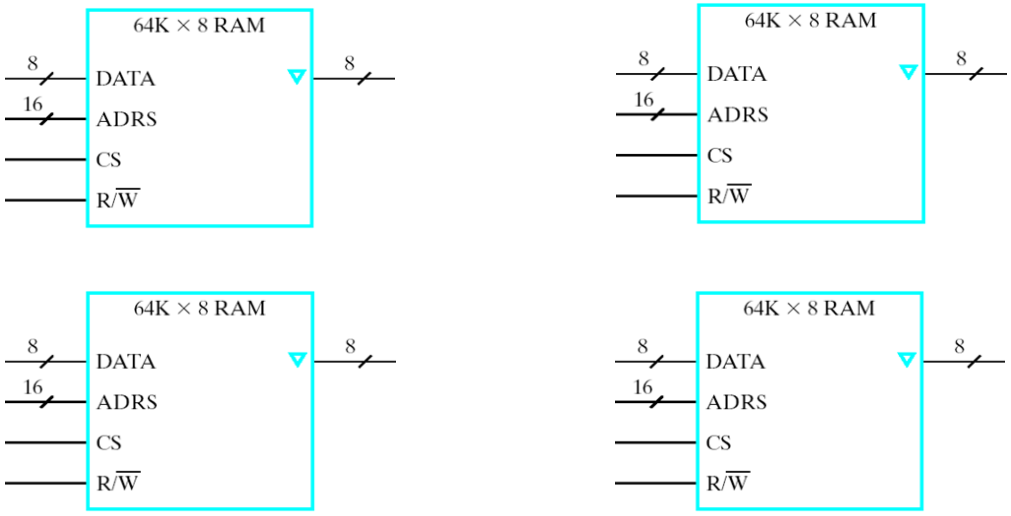
46. (EX & HW) An 8K×4 RAM chip is to be constructed.
    (a) If linear decoding scheme is used for the chip, what is the size of the decoder? Describe the hardware requirement of the decoder including the number of AND gates and the number of inputs per gate.
    (b) If coincident decoding scheme is used by splitting the internal decoder into row select and column select, what is the size of each decoder? Assume that the RAM cell array is as square as possible. Describe the hardware requirement of the decoders the same way as that in (a).

# §7-3

47. (EX & HW) Construct a 128K×16 RAM by using 64K×8 RAM ICs.




<Ans>






# §7-5

48. (a) (EX) Design a combinational circuit for the following functions by using a ROM (Read-Only memory).

$A(X, Y, Z) = \Sigma m(0,1,3,6,7)$
$B(X, Y, Z) = \Sigma m(0,1,4,5)$
$C(X, Y, Z) = \Sigma m(2,5)$
$D(X, Y, Z) = \Sigma m(1,2,4,5,6,7)$

i. Tabulate the truth table of the ROM.

ii. Specify the size of the ROM, i.e., the number of words × the number of bits per word.

iii. Reduce the size of the ROM without using external gates, if possible. Assume that the complements of the input variables are available.

(b) (HW) Repeat (a) for a combinational circuit with three inputs and three outputs. The input of the circuit is a 3-bit number ($A_2$ $A_1$ $A_0$) and the output ($F_2$ $F_1$ $F_0$) is the Gray code of the input number.

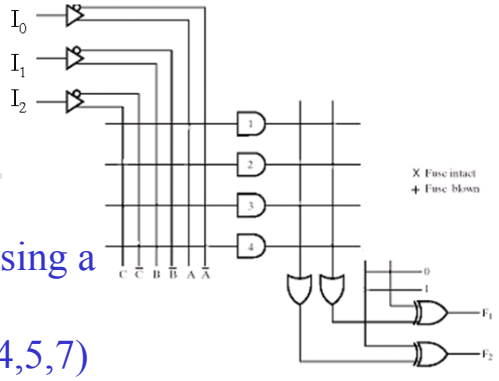
(a) A(X, Y, Z) = Σm(0,1,3,6,7)
B(X, Y, Z) = Σm(0,1,4,5)
C(X, Y, Z) = Σm(2,5)
D(X, Y, Z) = Σm(1,2,4,5,6,7)

ROM truth table:

| X | Y | Z | A | B | C | D |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | | |
| 0 | 0 | 1 | | | | |
| 0 | 1 | 0 | | | | |
| 0 | 1 | 1 | | | | |
| 1 | 0 | 0 | | | | |
| 1 | 0 | 1 | | | | |
| 1 | 1 | 0 | | | | |
| 1 | 1 | 1 | | | | |



# §7-6




49. (a) (EX) Implement the following two functions using a PLA (Programmable Logic Array):
A(x, y, z) = Σm(0,2,3,5,6), B(x, y, z) = Σm(1,4,5,7)

The outputs of the PLA are available in both "True" (non-complemented) and "Complement" form. An example of PLA is given below.

i. Derive the Boolean expressions for each of the function and its complement in SoP forms.

ii. Explain and specify the size required of the PLA as ***the number of inputs*** × ***the number of distinct product terms*** × ***the number of outputs***. Be sure to share product terms between functions.

iii. Derive the PLA programming table for the functions. (Note that the number of product terms should be minimized.)

(b) (HW) Repeat (a) for a the following two functions:
A(w, x, y, z) = Σm(2, 3, 8, 9, 10, 12, 13, 14)
B(w, x, y, z) = Σm(0, 1, 2, 4, 5, 6, 8)

(a)

A(x, y, z) = Σm(0,2,3,5,6)

| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

B(x, y, z) = Σm(1,4,5,7)

| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

PLA programming table:

| | | | | | outputs | |
|---|---|---|---|---|---|---|
| | | inputs | | | T/C | T/C |
| Product terms | | x | y | z | A | B |
| | 1 | | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | … | | | | | |



# §7-7

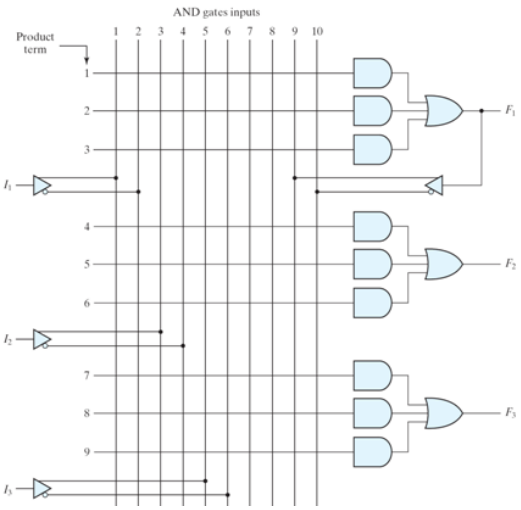
50. (a) (EX) Implement the following two functions using the PAL (Programmable Array Logic) given below.

A(x, y, z) = Σm(0, 2, 3, 5, 6),

B(x, y, z) = Σm(1, 4, 5, 7)

i. Derive the Boolean expressions of the functions.

ii. Derive the PAL programming table of the functions.



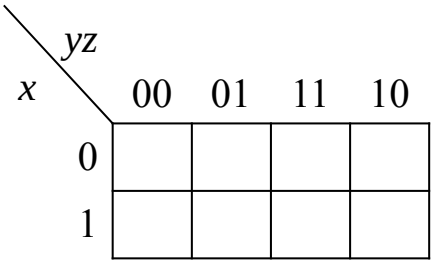

(b) (HW) Repeat (a) for a the following two functions:

A(w, x, y, z) = Σm(2, 3, 8, 9, 10, 12, 13, 14)

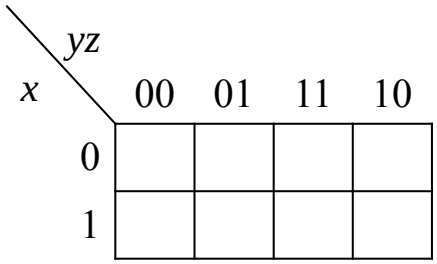
B(w, x, y, z) = Σm(0, 1, 2, 4, 5, 6, 8)

(a)

$A(x, y, z) = \Sigma m(0,2,3,5,6)$




$B(x, y, z) = \Sigma m(1,4,5,7)$




PAL programming table:

| Product terms | AND inputs | | | | outputs |
|---|---|---|---|---|---|
| | x | y | z | C | |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |