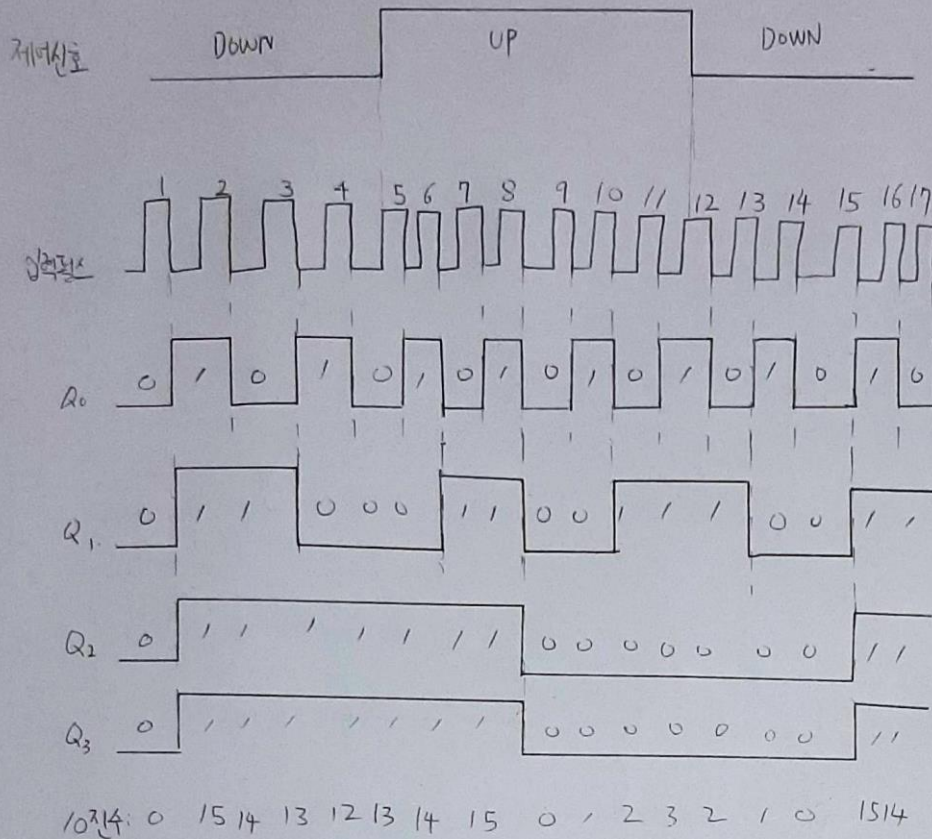


# REPORT

## 8,9 장 연습문제



8 10 <예제 8-2>에서 UP/down 신호 빈도에 따라 인가되는 경우, 즉 처음 4주기 동안에는 0(다운 카운팅), 다음 7주기는 '1'(업-카운팅), 그리고 마지막 5주기 동안에는 '0'(다운 카운팅)으로 세트되는 경우에 출력 파형들이 어떻게 발생할 것인지 그림 8-1과 같은 형태로 표현하라.



8.12 문제 8.11과 동일한 상태 시퀀스를 발생하는 mod-5 카운터를 JK-플립-플롭을 이용하여 설계하고, 문제 8.11의 결과 회로와 복잡도를 비교하라.

| 현재상태 |   |   | 다음상태 |   |   | 플립-플롭 입력들 |       |       |       |       |       |
|------|---|---|------|---|---|-----------|-------|-------|-------|-------|-------|
| A    | B | C | A    | B | C | $J_A$     | $K_A$ | $J_B$ | $K_B$ | $J_C$ | $K_C$ |
| 0    | 0 | 0 | 0    | 0 | 1 | 0         | X     | 0     | X     | 1     | X     |
| 0    | 0 | 1 | 0    | 1 | 0 | 0         | X     | 1     | X     | X     | 1     |
| 0    | 1 | 0 | 0    | 1 | 1 | 0         | X     | X     | 0     | 1     | X     |
| 0    | 1 | 1 | 1    | 0 | 0 | 1         | X     | X     | 1     | X     | 1     |
| 1    | 0 | 0 | 0    | 0 | 0 | X         | 1     | 0     | X     | 0     | X     |

BC 00 01 11 10

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | 0 | 0 | 0 | 1 | 0 |
| 1 | X | X | X | X | X |

$J_A = BC$

BC 00 01 11 10

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | 0 | 0 | X | X | X |
| 1 | 0 | X | X | X | X |

$J_B = C$

BC 00 01 11 10

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | 0 | 1 | X | X | 1 |
| 1 | 0 | X | X | X | X |

$J_C = A'$

BC 00 01 11 10

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | 0 | X | X | X | X |
| 1 | 1 | X | X | X | X |

$K_A = 1$

BC 00 01 11 10

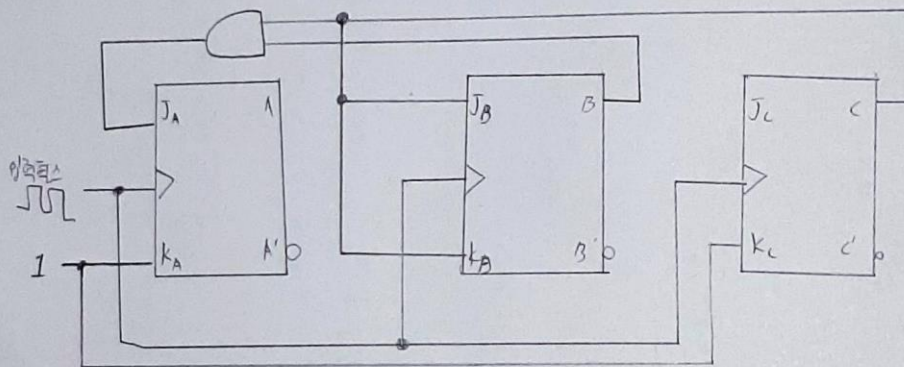
|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | 0 | X | X | X | 0 |
| 1 | X | X | X | X | X |

$K_B = C$

BC 00 01 11 10

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | 0 | X | 1 | 1 | X |
| 1 | X | X | X | X | X |

$K_C = 1$





8.14 3-비트 그레이 코드 카운터를 T 플립플롭을 이용하여 설계하라

| 현재상태 |   |   | 다음상태 |   |   | 플립 플롭 입력 |       |       |
|------|---|---|------|---|---|----------|-------|-------|
| A    | B | C | A    | B | C | $T_A$    | $T_B$ | $T_C$ |
| 0    | 0 | 0 | 0    | 0 | 1 | 0        | 0     | 1     |
| 0    | 0 | 1 | 0    | 1 | 1 | 0        | 1     | 0     |
| 0    | 1 | 1 | 0    | 1 | 0 | 0        | 0     | 1     |
| 0    | 1 | 0 | 1    | 1 | 0 | 1        | 0     | 0     |
| 1    | 1 | 0 | 1    | 1 | 1 | 0        | 0     | 1     |
| 1    | 1 | 1 | 1    | 0 | 1 | 0        | 1     | 0     |
| 1    | 0 | 1 | 1    | 0 | 0 | 0        | 0     | 1     |
| 1    | 0 | 0 | 0    | 0 | 0 | 1        | 0     | 0     |

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

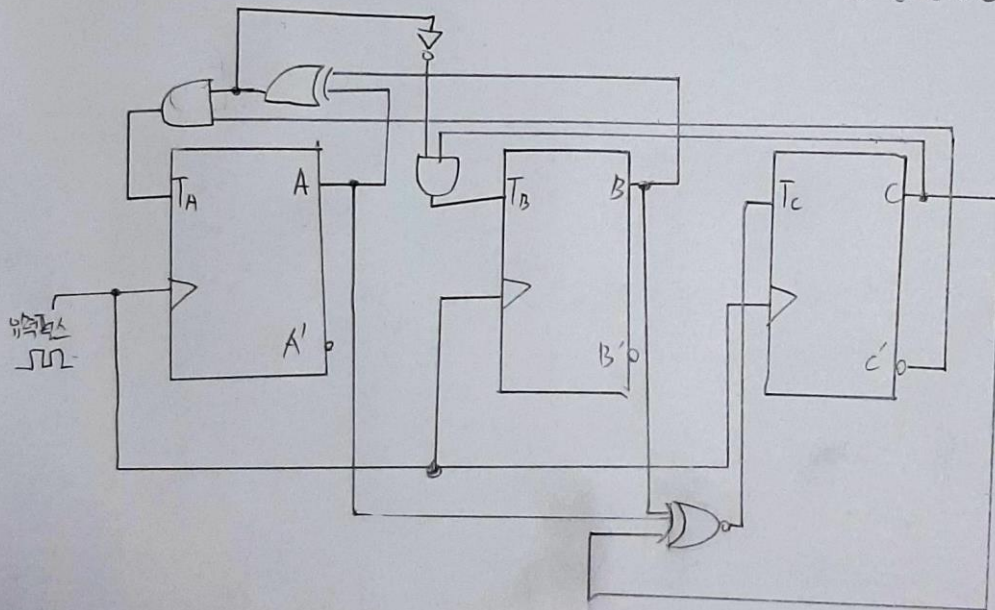
$$T_A = AB'C' + A'BC' = C'(A \oplus B)$$

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

$$T_B = A'B'C + ABC = (A \oplus B)'C$$

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

$$T_C = (A \oplus B \oplus C)'$$



8.16 동시 BCD 카운터를 T 플립-플롭을 이용하여 설계하는 경우에 대한 입력 함수들을 구하고, 그 결과가 8.3.4절에서 보인 입력 함수들과 일치하는지 확인하라.

| 현재 상태 |   |   |   | 다음 상태 |   |   |   | 플립-플롭 입력 |       |       |       |
|-------|---|---|---|-------|---|---|---|----------|-------|-------|-------|
| A     | B | C | D | A     | B | C | D | $T_A$    | $T_B$ | $T_C$ | $T_D$ |
| 0     | 0 | 0 | 0 | 0     | 0 | 0 | 1 | 0        | 0     | 0     | 1     |
| 0     | 0 | 0 | 1 | 0     | 0 | 1 | 0 | 0        | 0     | 1     | 1     |
| 0     | 0 | 1 | 0 | 0     | 0 | 1 | 1 | 0        | 0     | 0     | 1     |
| 0     | 0 | 1 | 1 | 0     | 1 | 0 | 0 | 0        | 1     | 1     | 1     |
| 0     | 1 | 0 | 0 | 0     | 1 | 0 | 1 | 0        | 0     | 0     | 1     |
| 0     | 1 | 0 | 1 | 0     | 1 | 1 | 0 | 0        | 0     | 1     | 1     |
| 0     | 1 | 1 | 0 | 0     | 1 | 1 | 1 | 0        | 0     | 0     | 1     |
| 0     | 1 | 1 | 1 | 1     | 0 | 0 | 0 | 1        | 1     | 1     | 1     |
| 1     | 0 | 0 | 0 | 1     | 0 | 0 | 1 | 0        | 0     | 0     | 1     |
| 1     | 0 | 0 | 1 | 0     | 0 | 0 | 0 | 1        | 0     | 0     | 1     |

| AB \ CD |    |    |    |    |
|---------|----|----|----|----|
|         | 00 | 01 | 11 | 10 |
| 00      | 0  | 0  | 0  | 0  |
| 01      | 0  | 0  | 1  | 0  |
| 11      | X  | X  | X  | X  |
| 10      | 0  | 1  | X  | X  |

$$T_A = AD + BCD$$

| AB \ CD |    |    |    |    |
|---------|----|----|----|----|
|         | 00 | 01 | 11 | 10 |
| 00      | 0  | 0  | 1  | 0  |
| 01      | 0  | 0  | 1  | 0  |
| 11      | X  | X  | X  | X  |
| 10      | 0  | 0  | X  | X  |

$$T_B = CD$$

| AB \ CD |    |    |    |    |
|---------|----|----|----|----|
|         | 00 | 01 | 11 | 10 |
| 00      | 0  | 1  | 1  | 0  |
| 01      | 0  | 1  | 1  | 0  |
| 11      | X  | X  | X  | X  |
| 10      | 0  | 0  | X  | X  |

$$T_C = AD$$

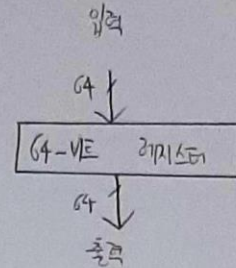
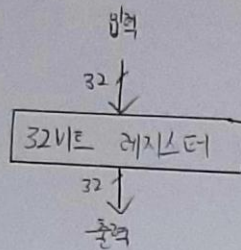
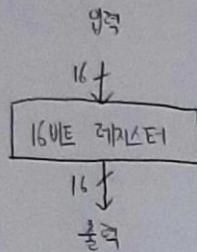
| AB \ CD |    |    |    |    |
|---------|----|----|----|----|
|         | 00 | 01 | 11 | 10 |
| 00      | 1  | 1  | 1  | 1  |
| 01      | 1  | 1  | 1  | 1  |
| 11      | X  | X  | X  | X  |
| 10      | 1  | 1  | X  | X  |

$$T_D = 1$$

8.3.4절의 입력 함수들과 일치한다.



8.18 16비트, 32비트 및 64비트 병렬 레지스터에 대한 간략화된 블록도를 도시하라.



8.20 데이터 '0000'이 저장되어 있는 4-비트 레지스터에 새로운 데이터 '1010'을 적재하고자 한다. 아래와 같은 시프트 동작을 사용하는 장치에 데이터 입력 순서를 각각 설명하고 풀면하라.

(1) 우측 시프트 - 입력 순서: 0, 1, 0, 1

| 클럭기 \ 데이터 | $d_3$ | $d_2$ | $d_1$ | $d_0$ |
|-----------|-------|-------|-------|-------|
| 초기 상태     | 0     | 0     | 0     | 0     |
| $t_1$     | 0     | 0     | 0     | 0     |
| $t_2$     | 1     | 0     | 0     | 0     |
| $t_3$     | 0     | 1     | 0     | 0     |
| $t_4$     | 1     | 0     | 1     | 0     |

(2) 좌측 시프트 - 입력 순서: 1, 0, 1, 0

| 클럭기 \ 데이터 | $d_3$ | $d_2$ | $d_1$ | $d_0$ |
|-----------|-------|-------|-------|-------|
| 초기 상태     | 0     | 0     | 0     | 0     |
| $t_1$     | 0     | 0     | 0     | 1     |
| $t_2$     | 0     | 0     | 1     | 0     |
| $t_3$     | 0     | 1     | 0     | 1     |
| $t_4$     | 1     | 0     | 1     | 0     |

8.22 문제 8.20을 시프트 동작이 연속적으로 두 번씩 수행되는 경우에 대하여 다시 풀라.

(1) 우측 시프트: 10110101  $\rightarrow$  01011010  $\rightarrow$  00101101

(2) 좌측 시프트: 10110101  $\rightarrow$  01101010  $\rightarrow$  11010100

8.24 B레지스터의 초기값 비트는 A 레지스터와 B 레지스터의 초기 비트에 같아지며 두 레지스터가 모두 좌측 시프트 동작을 한다.

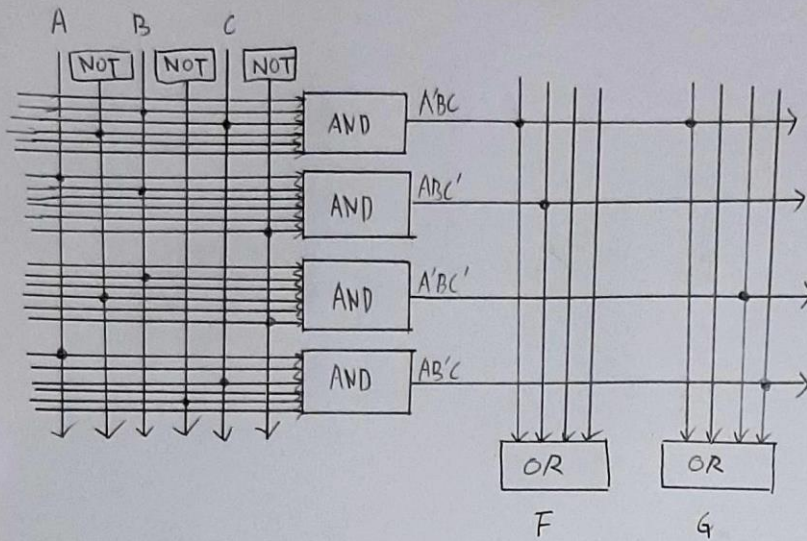
|       | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $B_4$ | $B_3$ | $B_2$ | $B_1$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 초기상태  | 0     | 0     | 0     | 0     | 1     | 0     | 1     | 0     |
| $t_1$ | 0     | 0     | 0     | 1     | 0     | 1     | 0     | 1     |
| $t_2$ | 0     | 0     | 1     | 0     | 1     | 0     | 1     | 0     |
| $t_3$ | 0     | 1     | 0     | 1     | 0     | 1     | 0     | 1     |
| $t_4$ | 1     | 0     | 1     | 0     | 1     | 0     | 1     | 0     |



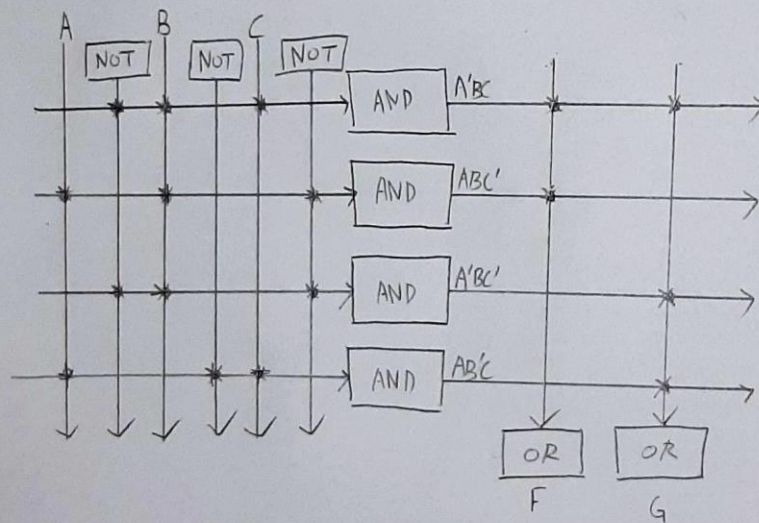
9.1

$$F = A'BC + ABC'$$

$$G = A'BC' + A'BC + AB'C$$

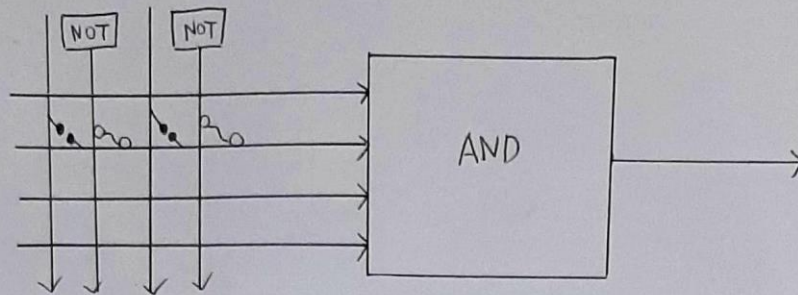


9.2





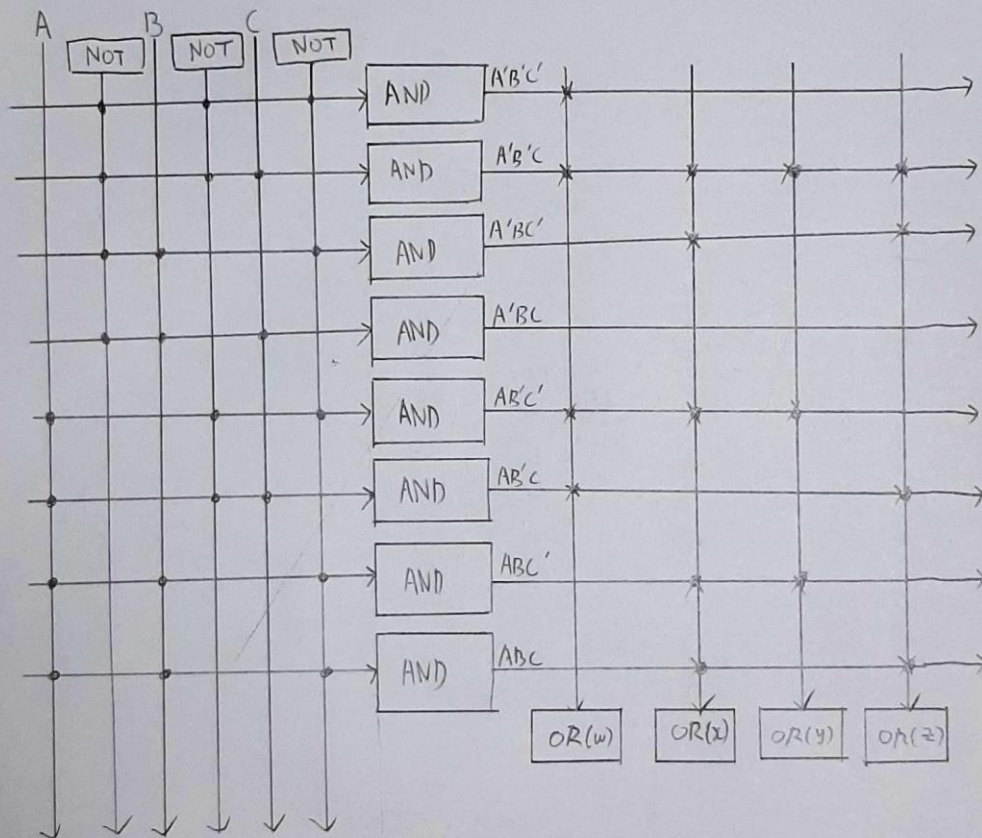
9.3



9.4

$$w = \sum(0, 1, 4, 5), \quad x = \sum(1, 2, 4, 6, 7)$$

$$y = A'B'C + AB'C' + ABC', \quad z = A'B'C + A'BC' + AB'C + ABC$$



9.5

| 주변 |   |   | 데이터 |   |   |   |
|----|---|---|-----|---|---|---|
| A  | B | C | w   | x | y | z |
| 0  | 0 | 0 | 1   | 0 | 0 | 0 |
| 0  | 0 | 1 | 1   | 1 | 1 | 1 |
| 0  | 1 | 0 | 0   | 1 | 0 | 0 |
| 0  | 1 | 1 | 0   | 0 | 0 | 1 |
| 1  | 0 | 0 | 1   | 1 | 1 | 0 |
| 1  | 0 | 1 | 1   | 0 | 0 | 1 |
| 1  | 1 | 0 | 0   | 1 | 1 | 0 |
| 1  | 1 | 1 | 0   | 1 | 0 | 1 |

9.6

$$w = A'BC + AB'C + ABD' + ABCD$$

$$x = A + C'D' + ABCD'$$

$$y = A'BD' + A'CD + ABCD$$

$$z = A'BCD' + A'BC'D + A'BCD + ABCD'$$

, 그림 9-8의 좌측 AND 배열에서 상위 4개의 AND 게이트 입력단기 각각  $A'BC$ ,  $AB'C$ ,  $ABD'$ ,  $ABCD$ 에 대응되는 피라점에 1을 표시하여 접속된다는 것을 표시한다. 그리고 우측 OR 배열의 첫 번째 출력을  $w$ 로 표시한다. 같은 방법으로 위의  $x, y, z$  함수에 대응되는 접속점을 표시하고 우측 OR 게이트의 출력들을 각각  $x, y, z$ 로 표시한다.

9.7

$$x = A'B'C' + AD + BC$$

$$y = A'CD + ACD' + A'BD + A'B'C' + AD + BC$$

$$z = BC + AD + A'B'C' + A'CD' + BCD$$

위 함수들에서  $y$ 와  $z$ 를 구현할 때, 밑줄 친 부분은 그림 9-9에서와 같이 출력  $x$ 를 이용하여 구현된다.

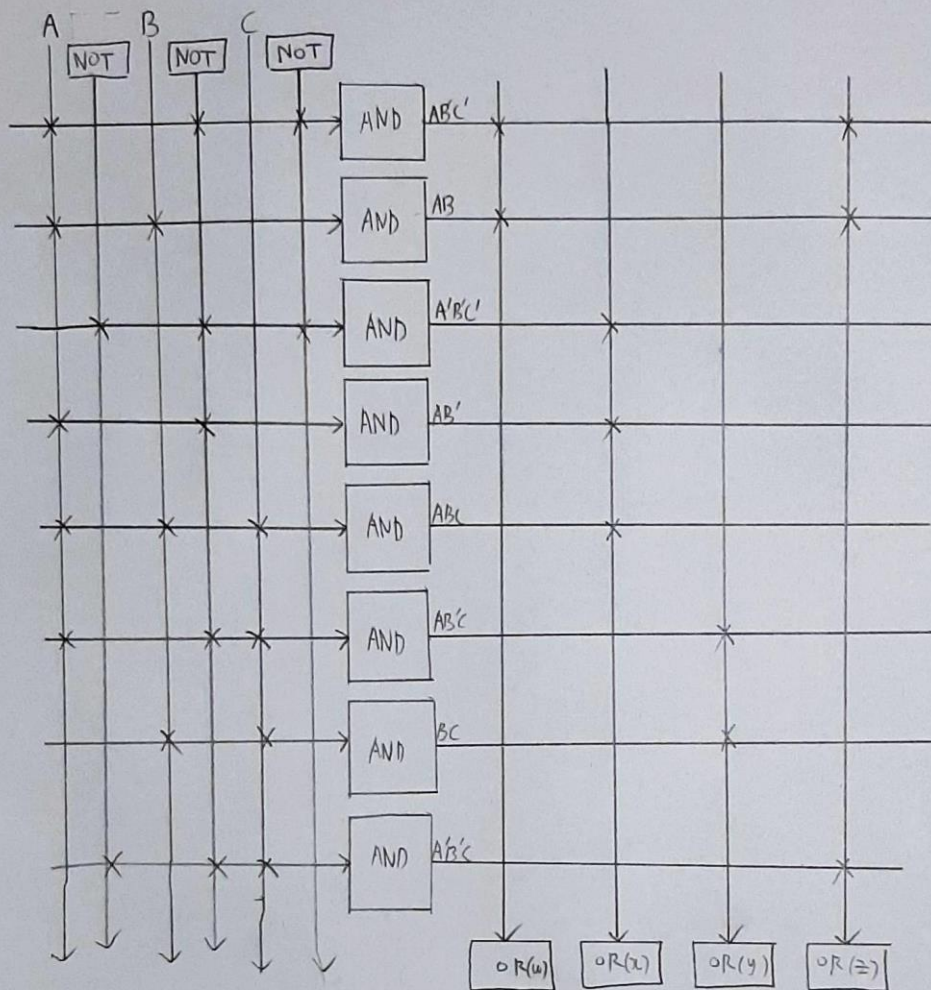


9.8 입력  $x$ 를 좌측 A입력선, 입력  $y$ 를 B입력선으로 각각 연결하고

상단 AND 게이트 선에서 A와 Q 교차점을 접속(X)시키며,

하단 AND 게이트 선에서 B 교차점을 접속시키면 된다.

9.9



9.10 CPLD와 FPGA의 차이점

CPLD는 기본블록(CLB)으로서 PAL이나 PLA를 사용하며, FPGA에 비하여 밀도가 낮다.

FPGA는 CLB를 기본블록으로 사용하는데, CLB는 CPLD의 CLB보다 더 단순한 구조를 가지고 있지만 그 수가 훨씬 더 많다. 또한 내부 상호연결망 및 기억장치도 포함하고 있다.