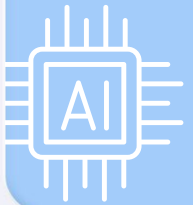
7 강.

조합논리회로(2)



컴퓨터과학과 강지훈 교수

학습 목차

7 강

contents

제5장. 조합논리회로

5.4
여러 가지 조합논리회로

5.4.1 코드 변환기

- 디지털 시스템의 코드 변환기
  - 디지털 시스템은 특정 목적을 달성하기 위해 구성요소들이 서로 정보를 교환함
  - 서로 정보를 교환하는 요소들이 서로 다른 코드 체계를 사용한다면, 상대의 데이터를 읽고 교환할 수 있도록 코드를 변환해주어야 함
  - 코드 변환기는 2진 코드를 다른 형태의 2진 코드로 바꾸는 작업을 수행함
  - 코드 변환기의 예
    - BCD to Excess-3(3-초과) 코드 변환기
    - BCD to 9의 보수 변환기

5.4.1 코드 변환기

- BCD to Excess-3(3-초과) 코드 변환기
  - BCD 코드를 Excess-3 코드로 바꿀 때 둘 다 4비트를 이용해 10진수 표현
  - 즉, 4개의 입력, 4개의 출력이 필요함

10진수	BCD8421	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

변환기 설계 과정

1. 진리표 작성
2. 카르노 도표를 이용해 간소화
3. 출력 부울함수 유도
4. 논리 회로도 작성

5.4.1 코드 변환기

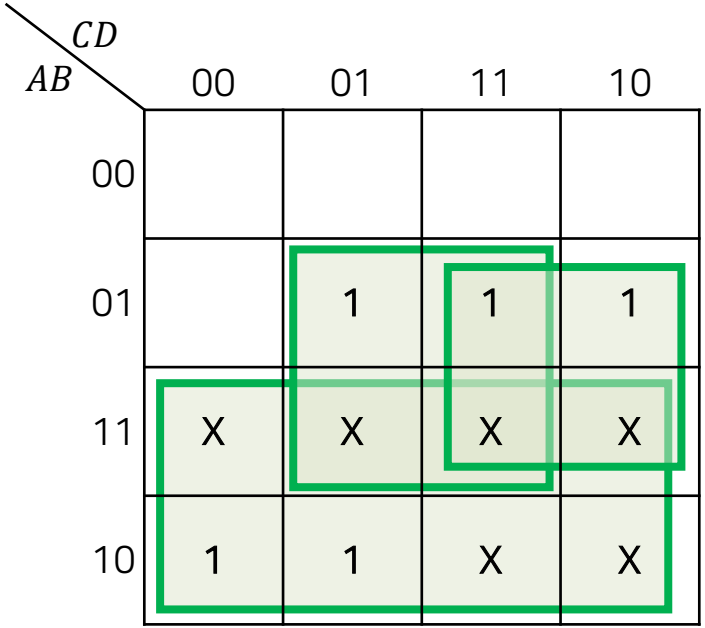
- 코드 변환기를 위한 진리표 작성
  - BCD는 0~9까지만 표현하기 때문에 4비트로 표현할 수 있는 나머지 숫자 10~15는 무관조건 처리

입력 BCD 코드					출력 3-초과 코드			
10진수	*A*	*B*	*C*	*D*	*W*	*X*	*Y*	*Z*
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

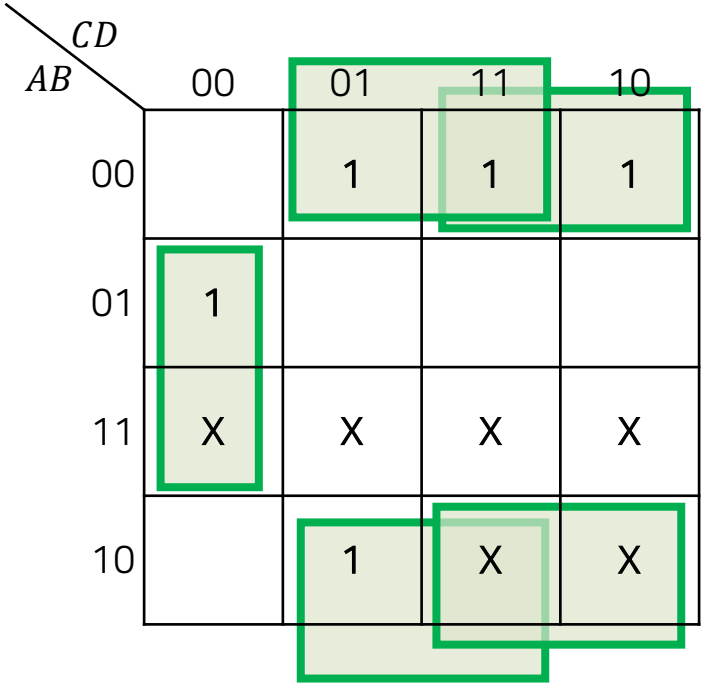
• 카르노 도표를 이용한 간소화 및 부울함수 도출(1)

AB \ CD	00	01	11	10
00				
01				
11	X	X	X	X
10			X	X

- 카르노 도표를 이용한 간소화 및 부울함수 도출(2)



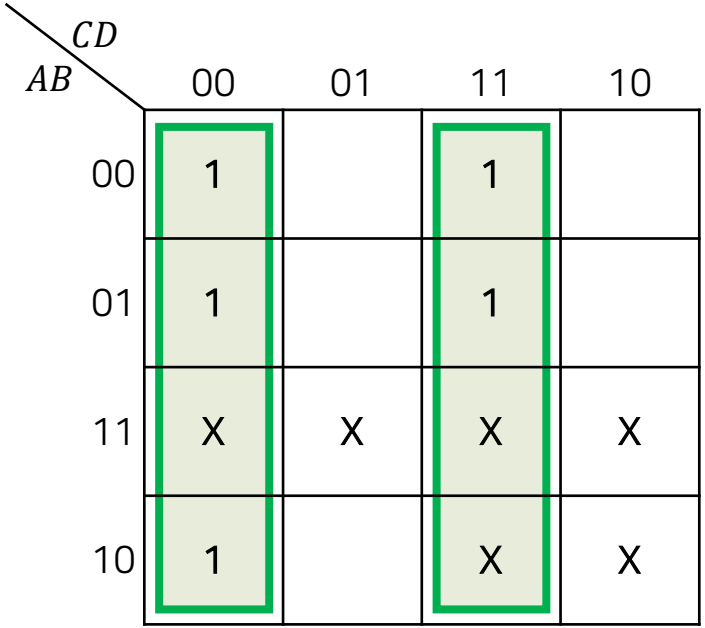
$$W = A + BC + BD$$



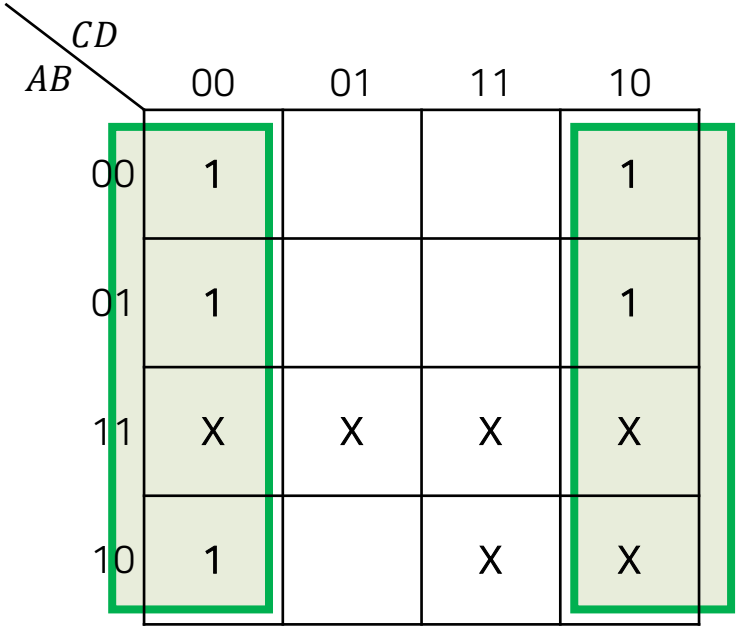
$$X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$$

5.4.1 코드 변환기

- 카르노 도표를 이용한 간소화 및 부울함수 도출(3)



$$Y = CD + \bar{C}\bar{D}$$



$$Z = \bar{D}$$

5.4.1 코드 변환기

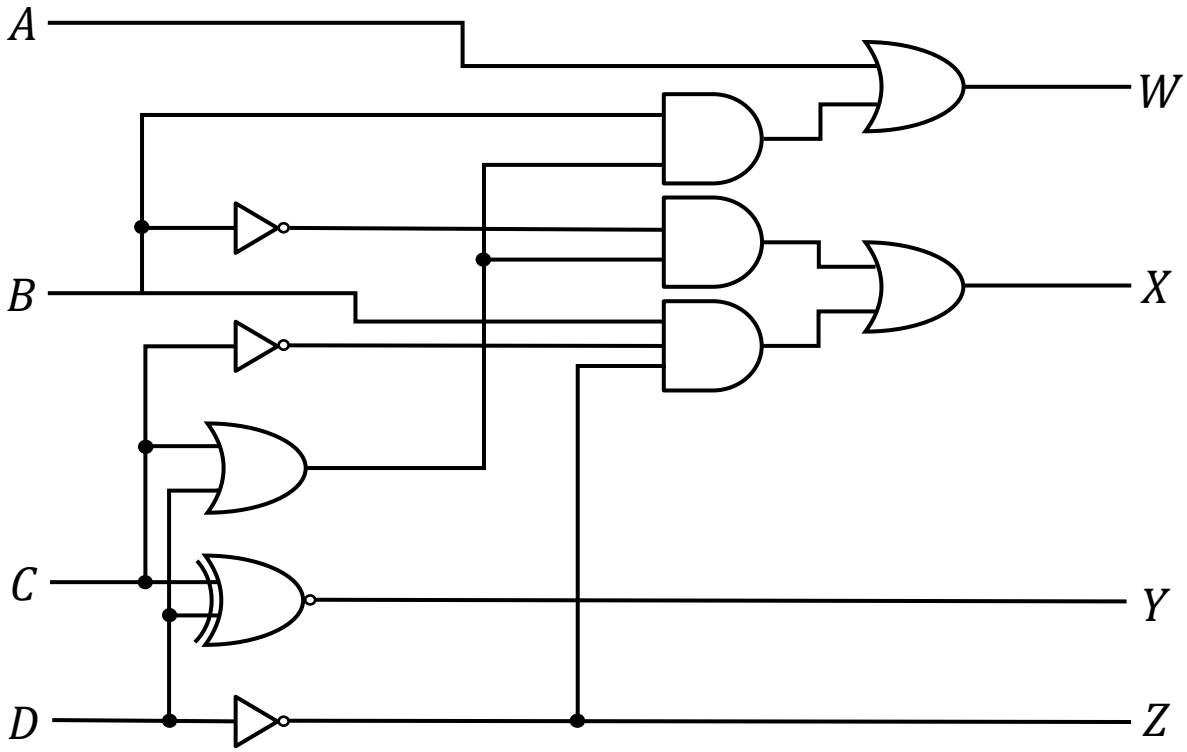
- 논리회로도 작성

$$W = A + BC + BD$$
$$= A + B(C + D)$$

$$X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$$
$$= \bar{B}(C + D) + B\bar{C}\bar{D}$$

$$Y = CD + \bar{C}\bar{D} = \overline{C \oplus D}$$

$$Z = \bar{D}$$

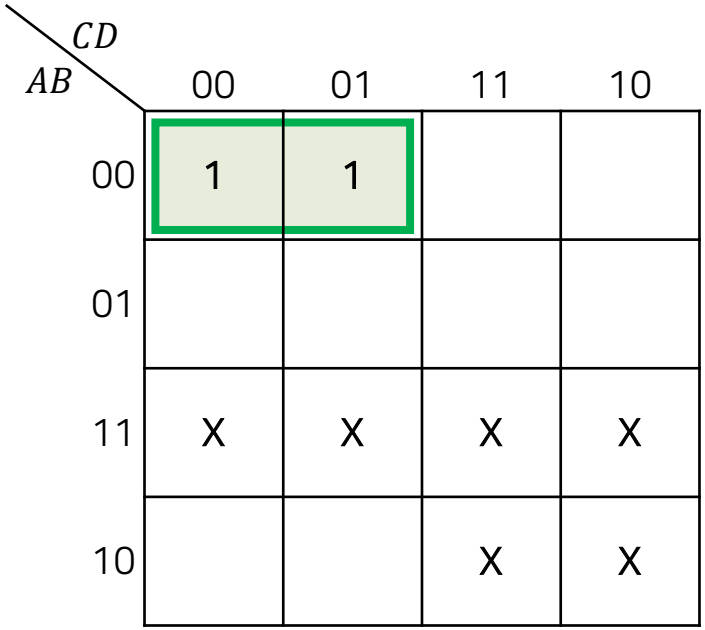


- BCD to 9의 보수 변환기
  - BCD 코드가 입력되면 9의 보수로 변환
  - 10~15는 무관조건 처리

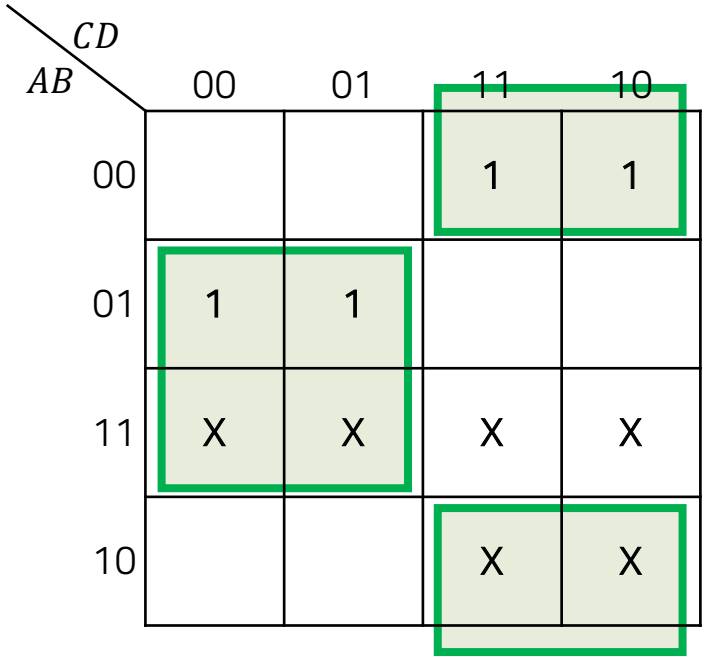
10진숫자	A	B	C	D	9의 보수	W	X	Y	Z
0	0	0	0	0	9	1	0	0	1
1	0	0	0	1	8	1	0	0	0
2	0	0	1	0	7	0	1	1	1
3	0	0	1	1	6	0	1	1	0
4	0	1	0	0	5	0	1	0	1
5	0	1	0	1	4	0	1	0	0
6	0	1	1	0	3	0	0	1	1
7	0	1	1	1	2	0	0	1	0
8	1	0	0	0	1	0	0	0	1
9	1	0	0	1	0	0	0	0	0

5.4.1 코드 변환기

- 카르노 도표를 이용한 간소화 및 부울함수 도출(1)



$$W = \bar{A}\bar{B}\bar{C}$$



$$X = B\bar{C} + \bar{B}C$$

• 카르노 도표를 이용한 간소화 및 부울함수 도출(2)

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

$Y = C$

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

$Z = \overline{D}$

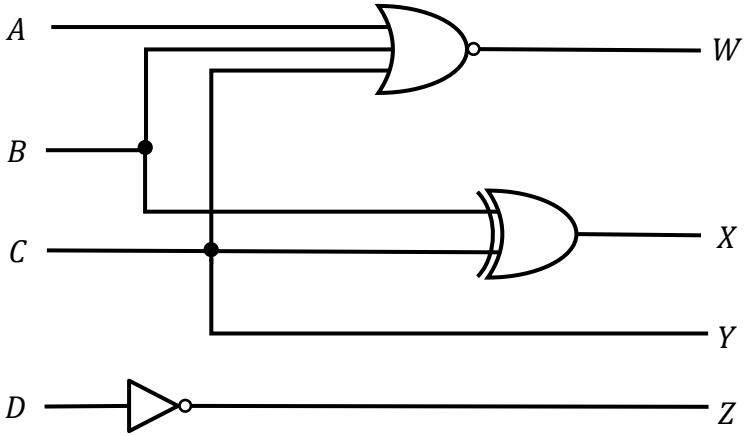
- 논리회로도 작성

$$W = \bar{A}\bar{B}\bar{C}$$
$$= \overline{A + B + C}$$

$$X = B\bar{C} + \bar{B}C$$
$$= B \oplus C$$

$$Y = C$$

$$Z = \bar{D}$$



5.4.2 패리티 발생기/검사기

- 패리티 비트
  - 2진 데이터를 주고 받을 때 노이즈나 회로상 문제로 인해 에러가 발생할 수 있음
  - 하지만, 디지털 시스템은 이게 에러인지 자체적으로 식별할 수 없음
  - 따라서, 에러를 검출하기 위한 장치를 추가해줘야 함
  - 패리티 비트는 데이터 전송 중 오류 검출을 위해 추가하는 보조 비트를 말함
    - 과거에는 패리티 비트가 매우 대중적인 오류 검출 방법이었지만 현재는 간단한 오류 검출을 위해 사용됨 -> 현재에는 더 정교한 방식을 많이 사용함
    - 예) 시리얼 통신, RAID 스토리지, 무선 통신 등

5.4.2 패리티 발생기/검사기

- 패리티 비트의 기본적인 작동 방식
  - 2진 데이터에서 1의 개수를 조정하여 오류 검출이 가능하도록 함
    - 패리티 비트는 보통 데이터에 1 비트가 추가되고, 별개의 비트로 취급됨
  - 짝수 패리티 비트
    - 2진 데이터와 패리티 비트에 포함된 1의 개수가 짝수가 되도록 패리티 비트 값을 설정

데이터 (4비트)	1의 개수	짝수 패리티 비트	전송 데이터 (5비트)
1010	2	0	1010 **0**

  - 홀수 패리티 비트
    - 2진 데이터와 패리티 비트에 포함된 1의 개수가 홀수가 되도록 패리티 비트 값을 설정

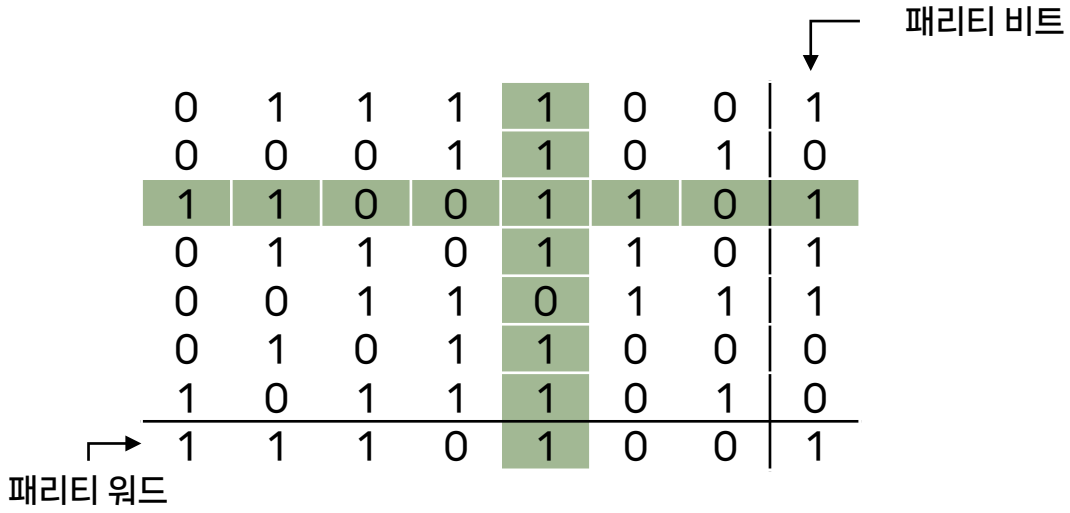
데이터 (4비트)	1의 개수	홀수 패리티 비트	전송 데이터 (5비트)
1010	2	1	1010 **1**

- 패리티 비트 부가 방식 예시

10진숫자	BCD	짝수 패리티의 BCD	1의 개수	BCD	홀수 패리티의 BCD	1의 개수
0	0000	0000 0	0	0000	0000 1	1
1	0001	0001 1	2	0001	0001 0	1
2	0010	0010 1	2	0010	0010 0	1
3	0011	0011 0	2	0011	0011 1	3
4	0100	0100 1	2	0100	0100 0	1
5	0101	0101 0	2	0101	0101 1	3
6	0110	0110 0	2	0110	0110 1	3
7	0111	0111 1	4	0111	0111 0	3
8	1000	1000 1	2	1000	1000 0	1
9	1001	1001 0	2	1001	1001 1	3

5.4.2 패리티 발생기/검사기

- 이중 패리티 검출 방식
  - 일반 패리티 검출 방식은 2개 이상의 비트 값에 에러가 있을 시 검출할 수 없음
  - 이러한 경우 이중 패리티 검출 방식을 사용함
    - 코드를 한 묶음으로 하여, 그 묶음의 수직방향과 수평 방향으로 패리티 비트를 부가



- 이중 패리티 검출 방식 예시
  - 4번째 행의 코드 0110110 에 에러가 발생하여 코드 0111110 으로 된 경우

0	1	1	1	1	0	0	1
0	0	0	1	1	0	1	0
1	1	0	0	1	1	0	1
0	1	1	1	1	1	0	1
0	0	1	1	0	1	1	1
0	1	0	1	1	0	0	0
1	0	1	1	1	0	1	0
1	1	1	0	1	0	0	1

패리티 에러가 검출됨

패리티 에러가 검출됨

- 패리티 비트와 패리티 워드를 통해 에러가 발생한 위치 확인 가능

5.4.2 패리티 발생기/검사기

- 패리티 검출을 위한 조합 논리회로
  - 패리티 발생기(parity generator)
    - 에러 검출 및 정정을 위해 송신 측에서 전송 정보에 패리티 비트를 부가하기 위해 패리티 비트를 생성하는 회로
  - 패리티 검사기(parity checker)
    - 수신 측에서 송신된 정보의 에러 검출을 위해 패리티 비트를 검사하는 회로

- 홀수 패리티 발생기 설계(1)
  - 3비트 데이터에 하나의 패리티 비트를 부가하는 홀수 패리티 발생기 설계

1. 진리표 작성

3비트 정보			패리티 비트
X	*Y*	*Z*	*P*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

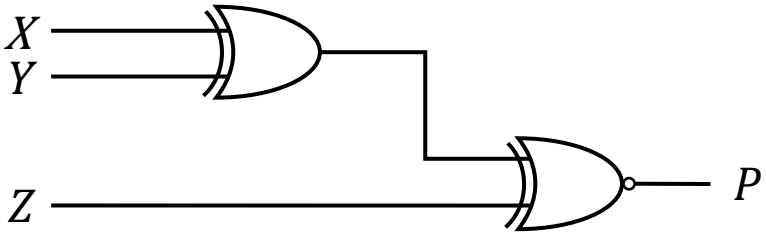
• 홀수 패리티 발생기 설계(2)

2. 카르노 도표를 이용해 출력 부울함수 유도

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

$$
\begin{aligned}
P &= \bar{X}\bar{Y}\bar{Z} + \bar{X}YZ + X\bar{Y}Z + XY\bar{Z} \\
&= \bar{X}(\bar{Y}\bar{Z} + YZ) + X(Y\bar{Z} + \bar{Y}Z) \\
&= \bar{X}(\overline{Y \oplus Z}) + X(Y \oplus Z) \\
&= \overline{X \oplus Y \oplus Z}
\end{aligned}
$$

3. 논리 회로도 작성



• 패리티 검사기 설계(1)

1. 진리표 작성

전송된 4비트				패리티 에러검사
W	X	Y	Z	P
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

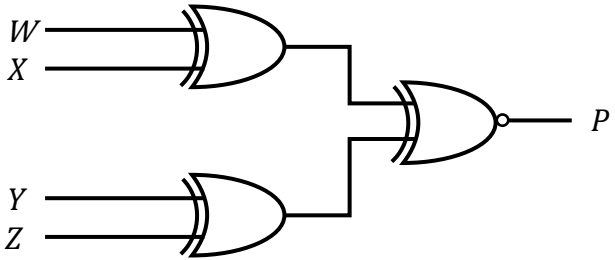
- 패리티 검사기 설계(2)

2. 카르노 도표를 이용해 출력 부울함수 유도

WX \ YZ	00	01	11	10
00	1		1	
01		1		1
11	1		1	
10		1		1

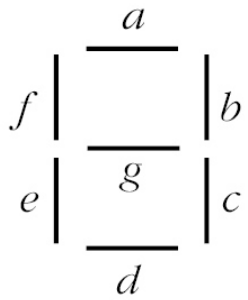
$$P = \bar{W}\bar{X}\bar{Y}\bar{Z} + \bar{W}\bar{X}YZ + \bar{W}X\bar{Y}Z + \bar{W}XY\bar{Z} + W\bar{X}\bar{Y}Z + W\bar{X}Y\bar{Z} + WX\bar{Y}\bar{Z} + WXYZ$$
$$= \overline{W \oplus X \oplus Y \oplus Z}$$

3. 논리 회로도 작성

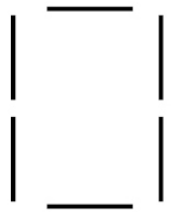


BCD-7 세그먼트 디스플레이

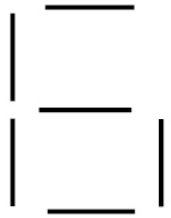
- BCD – 7 세그먼트
  - BCD를 입력 받아 7 세그먼트 디스플레이에 숫자를 표시하는 디코더
    - 입력된 코드를 해석(디코딩)하여 특정 출력을 활성화 하는 역할
  - 7 세그먼트는 7개의 LED 또는 LDC로 이루어진 숫자 표시 장치



세그먼트 표시기



a, b, c, d, e, f
세그먼트로
표시된 10진수 0

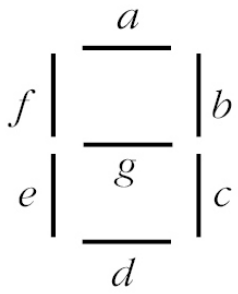


a, c, d, e, f, g
세그먼트로
표시된 10진수 6

BCD-7 세그먼트 디스플레이

• BCD – 7 세그먼트 디스플레이 설계(1)

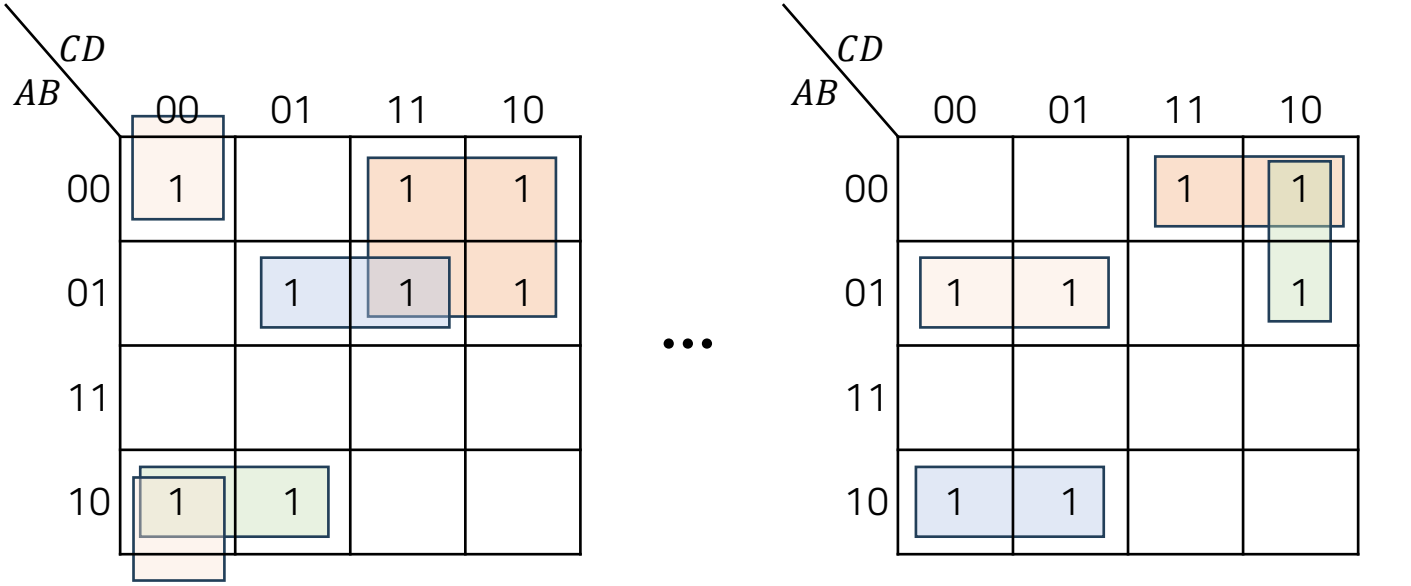
1. 진리표 작성



BCD 입력				세븐 세그먼트 출력						
A	*B*	*C*	*D*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
나머지 모든 입력				0	0	0	0	0	0	0

• BCD – 7 세그먼트 디스플레이 설계(2)

2. 카르노 도표를 이용해 출력 부울함수 유도

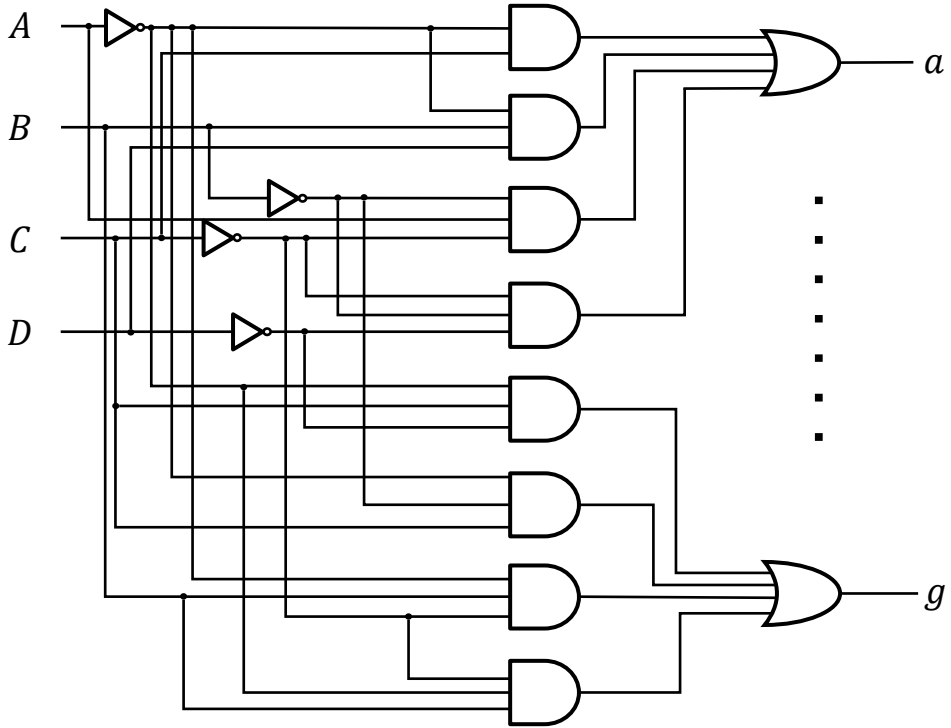


$$a = \bar{A}C + \bar{A}BD + \bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}$$

$$g = \bar{A}C\bar{D} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

• BCD – 7 세그먼트 디스플레이 설계(3)

3. 논리 회로도 작성



제5장. 조합논리회로

5.5

MSI를 이용한 조합논리회로(1)

5.5 MSI를 이용한 조합논리회로

- 실제 회로 제작에서는..
  - 효율적인 회로를 만들기 위해서는 필요한 게이트의 수를 최소화 해야함
  - 하지만, 처음부터 끝까지 개별 논리 게이트를 직접 배치하고 최적화 설계하는 것은 비효율적임
  - 왜 비효율 적일까?

    회로 제작에 소요되는 비용과 시간이 증가함

5.5 MSI를 이용한 조합논리회로

- **실제 회로를 만들 때에는 어떠한 기능을 위해 게이트, 배선, 트랜지스터 등의 집합이 이미 최적화되어 만들어져 패키징 되어있는 집적회로(IC)를 많이 사용함**
  - 이미 검증되어 만들어진 논리 회로를 사용한다면, 그만큼 설계, 테스트, 검증에 소요되는 비용과 설계 시간이 크게 단축됨
- **이미 만들어진 회로를 사용하여 다양한 조합논리회로를 설계할 수 있음**
  - 프로그래밍의 예를 들면 일종의 라이브러리로 비유할 수 있음

5.5 MSI를 이용한 조합논리회로

- MSI(Medium Scale Integrated circuits) 장치(MSI 소자)
  - 여러 개의 논리 게이트의 조합으로 특정한 기능을 수행하는 중간 규모의 집적회로
  - 단순한 논리 게이트를 조합하여 더 복잡한 기능을 수행하도록 모듈로 설계됨
  - 이미 검증된 최적화된 설계를 제공하기 때문에 설계 생산성을 향상하고 유지보수가 용이해짐
  - 논리 블록 형태로 사용되기 때문에 설계자가 처음부터 끝까지 직접 할 필요 없음
    - 대표적인 MSI – 인코더, 디코더, 멀티플렉서, 디멀티플렉서

5.5.1 인코더/디코더

- 인코더
  - **부호화 되지 않은 입력을 받아서 부호화 된 출력을 내보내는 부호화기**
    - 데이터를 특정 형식으로 변환하는 장치
    - 10진수나 8진수를 입력 받아 2진수나 BCD 코드로 변환하는 조합논리회로
  - **기본적으로 N개의 입력을 받아 $\log_2 N$개의 출력으로 변환함**
    - 즉, 2^n개의 입력과 n개의 출력을 가짐(경우에 따라 추가적인 출력이 존재할 수 있음)
    - 출력은 입력값에 대응하는 2진 코드를 생성함

5.5.1 인코더/디코더

- 8진수를 2진수로 바꾸는 인코더 회로 설계
  - 8진수(0~7)를 표현하기 위한 8개의 신호를 2진수로 변환하는 인코더
  - 사람에게 8진수는 단순한 숫자이지만 디지털 시스템에서는 이를 전기적인 신호로 표현해야 함
    - 사람은 8진수 하나를 숫자로 이해하지만 회로에서는 해당 숫자를 특정 신호로 표현해야 함
  - 이를 위해 일반적으로 1-hot encoding 방식, 즉 8개의 개별적인 신호 중 하나만 활성화 하는 방식으로 각 숫자를 표현하는 방법을 사용함
    - 하나의 신호만 1, 나머지는 0
    - 0~7까지 신호를 표현할 때 하나의 독립적인 신호에 매핑함

• 인코더 설계(1)

1. 진리표 작성

입력								출력		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

그 이외의 조건은 무관조건

• 인코더 설계(2)

2. 출력함수 유도

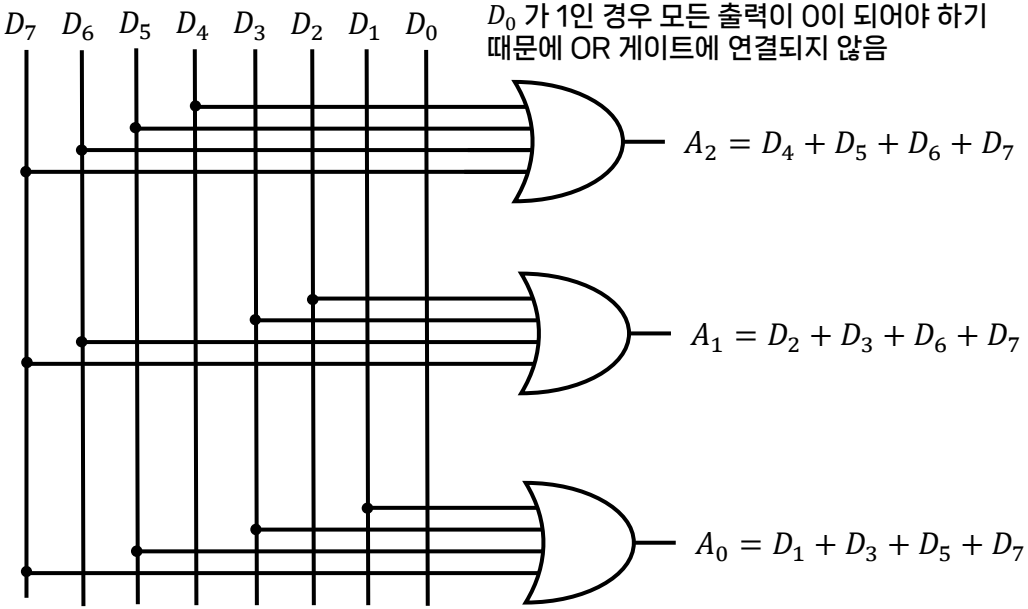
$$A_0 = D_1 + D_3 + D_5 + D_7$$
$$A_1 = D_2 + D_3 + D_6 + D_7$$
$$A_2 = D_4 + D_5 + D_6 + D_7$$

입력								출력		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

3. 논리 회로도 작성

임의의 순간에 하나의 입력만 1이어야 함

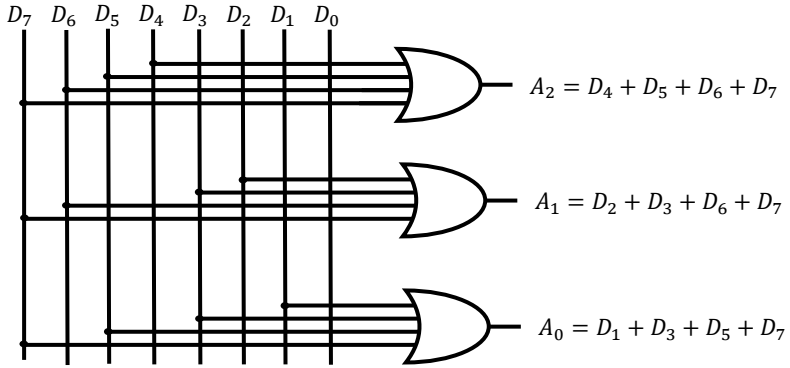
D_0 가 1인 경우 모든 출력이 0이 되어야 하기 때문에 OR 게이트에 연결되지 않음



5.5.1 인코더/디코더

• 우선순위 인코더

- 설계한 인코더는 오직 1개의 값만 1이어야 하며, 그렇지 않으면 의미가 없어짐
- 예를 들어 D_3, D_6가 동시에 1이 되면 A_2, A_1, A_0는 1 1 1이 되지만, 이는 2진수로 3도 6도 아님



- 이런 문제를 해결하기 위해 입력 값의 첨자가 높을 수록 높은 우선순위를 준다면 D_3, D_6이 동시에 1이 되어도 D_6이 더 높은 우선순위를 가지게 됨
  - 결과적으로 출력이 110이 되어 10진수 6을 나타낼 수 있음

내용 정리

Summary Contents

01 다양한 조합논리회로

- 코드 변환기
- 패리티 발생/검사기
- BCD-7 세그먼트 디스플레이

02 MSI를 이용한 조합논리회로

- 인코더

AI

다음시간에는

8강.

조합논리회로(3)