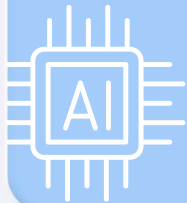




디지털논리회로 [Digital Logic Circuits]

6강.

# 조합논리회로(1)



컴퓨터과학과 **강지훈** 교수



## 제5장 | 조합논리회로

contents

# 학습 목차

## ▶ 6 강

### 01 조합논리회로의 분석과 설계

- 조합논리회로 개요
  - 조합논리회로의 분석
  - 조합논리회로의 설계
- 

### 02 기본 연산회로

- 가산기
- 감산기
- 가·감산기

## 6강. 조합논리회로(1)



### 제5장. 조합논리회로

#### 5.1

## 조합논리회로 개요 및 분석과 설계



## 5.1.1 조합논리회로의 개요

### • 개요

출력이 현재 입력 신호의 논리적 "조합"에 따라 즉시 결정

조합논리회로  
(combinational  
logic circuit)

- 저장요소 없음
- 현재 입력만으로 출력 결정

디지털  
논리회로

출력이 현재 입력 뿐 아니라 이전 상태와 같이  
시간적 "순서"에 따라 결정

순서논리회로  
(sequential  
logic circuit)

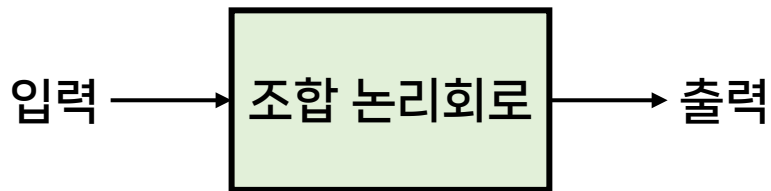
- 조합논리회로에 저장요소  
(ex: Flip-Flop 등)를 추가
- 저장요소의 상태, 입력변수에  
따라 출력 결정



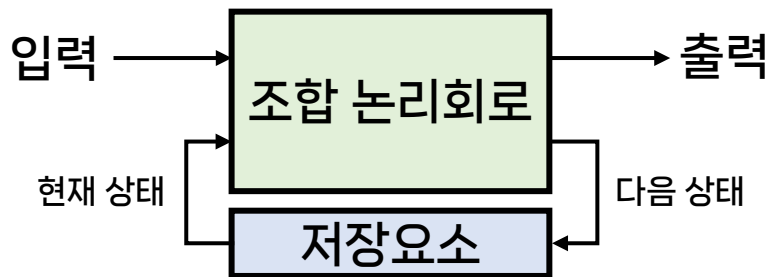
## 5.1.1 조합논리회로의 개요



### • 조합논리회로와 순서논리회로의 블록도



조합논리회로의 블록도



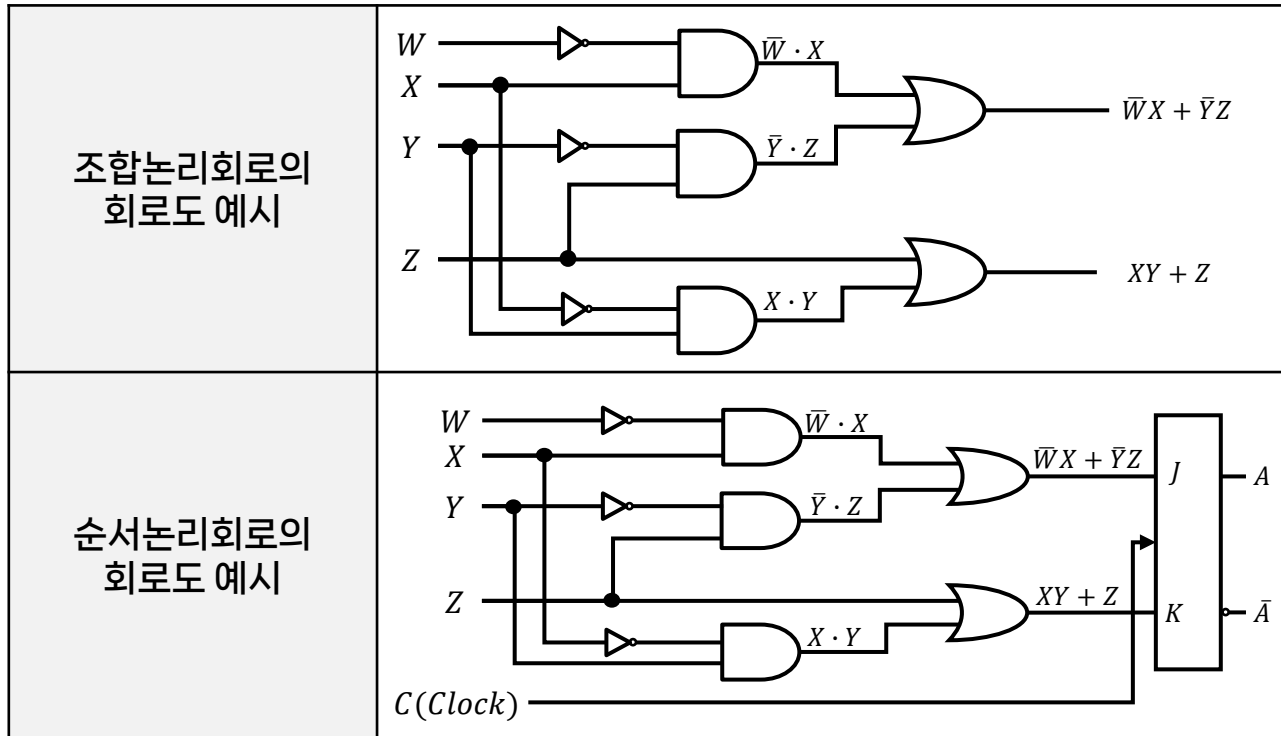
순서논리회로의 블록도



## 5.1.1 조합논리회로의 개요



### • 조합논리회로와 순서논리회로

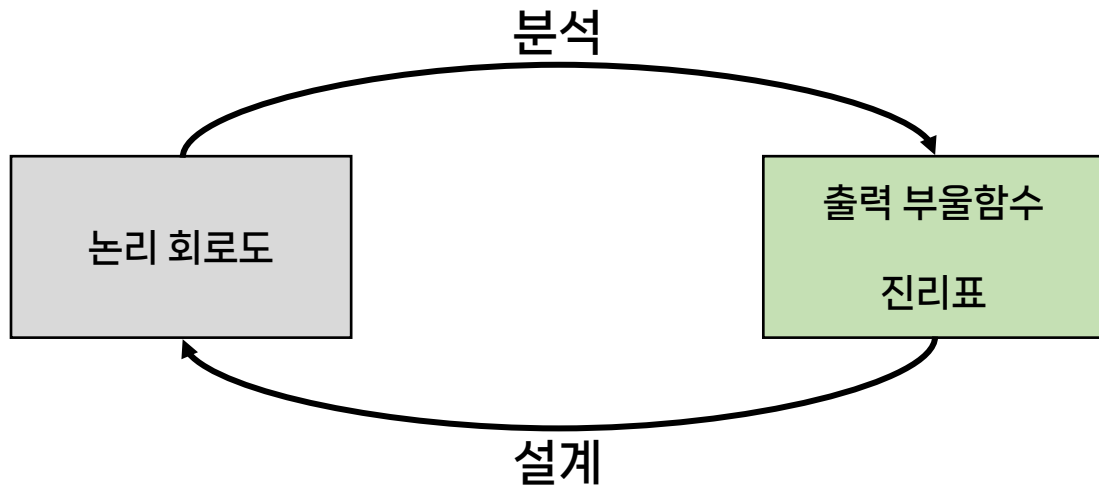




## 5.2 조합논리회로의 분석과 설계



### • 회로의 분석과 설계





## 5.2.1 조합논리회로의 분석



### • 조합논리회로의 분석

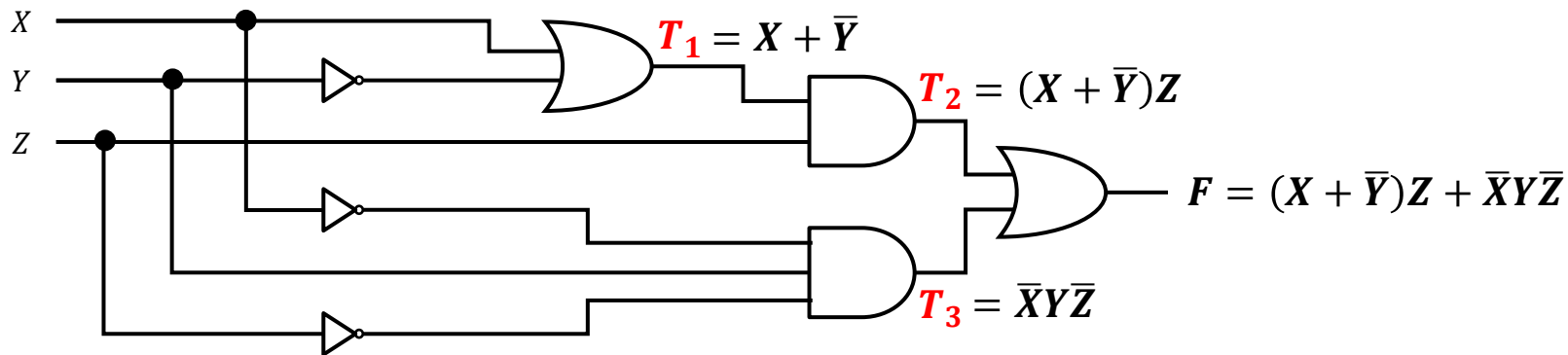
- 주어진 논리회로에 대해 입출력 관계를 구하는 것
  - 입력, 출력은 2진수 기반
  - 2진수 기반의 입력, 논리식, 출력을 구하여 분석을 수행
  - 즉, 진리표나 부울함수를 도출하여 분석을 수행함





## 5.2.1 조합논리회로의 분석

### • 논리회로도에서 부울함수 유도



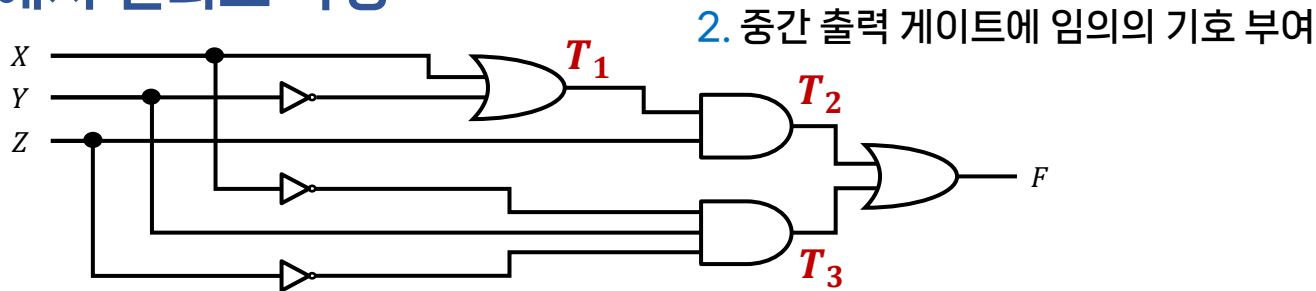
1. 각 중간 출력 게이트에 임의의 기호를 부여하고 부울함수 도출
2. 도출된 부울함수를 입력으로 하는 게이트에 다시 임의의 기호를 부여하여 부울함수 도출
3. 출력에 대한 부울함수를 얻을 때 까지 두 번째 단계를 반복



## 5.2.1 조합논리회로의 분석



### • 논리회로도에서 진리표 작성



| X | Y | 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 |

1. 입력변수에 대한 2진수 조합 리스트 작성

| $T_1$ | $T_2$ | $T_3$ |
|-------|-------|-------|
| 1     | 0     | 0     |
| 1     | 1     | 0     |
| 0     | 0     | 1     |
| 0     | 0     | 0     |
| 1     | 0     | 0     |
| 1     | 1     | 0     |
| 1     | 0     | 0     |
| 1     | 1     | 0     |

3. 입력변수로만 이루어진 출력 게이트에 대한 진리표 작성

| F |
|---|
| 0 |
| 1 |
| 1 |
| 0 |
| 0 |
| 1 |
| 0 |
| 1 |

4. 모든 출력 란을 작성하고 최종 출력 값 정리

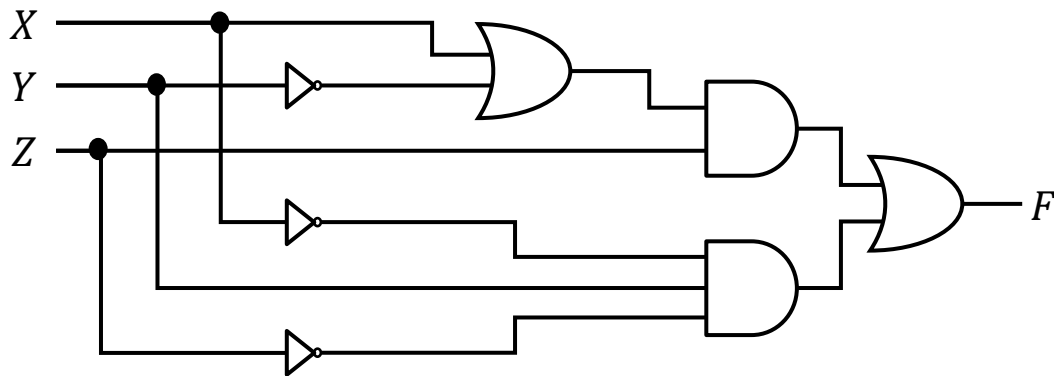




## 5.2.1 조합논리회로의 분석



### • 논리회로도와 진리표



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



## 5.2.2 조합논리회로의 설계



### • 문제에 대한 설명을 통해 부울함수와 논리회로도를 구하는 과정

1. 주어진 문제로부터 입력변수와 출력변수의 개수를 결정하고 간단한 블록도 작성
2. 입력 및 출력 변수의 관계를 정의하는 진리표 작성
3. 각 출력을 입력 변수의 함수로 나타내고 간소화
4. 논리회로도 작성

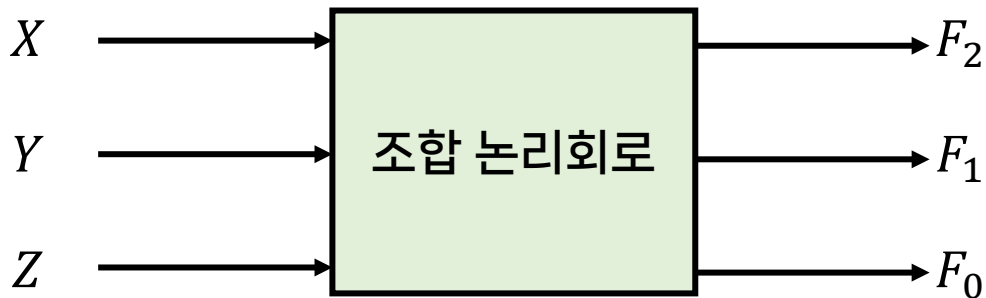


## 5.2.2 조합논리회로의 설계



- 3비트 2진수를 입력 받아 10진수로  
홀수인 것은 출력에 통과, 짝수인 것은 통과시키지 않고  
000으로 출력하는 조합논리회로 설계

1. 주어진 문제로부터 입력변수와 출력변수의 개수를 결정하고 간단한 블록도 작성





## 5.2.2 조합논리회로의 설계

### 2. 입력 및 출력 변수의 관계를 정의하는 진리표 작성

3비트 2진수 입력 받아 10진수로 홀수인 것만 출력으로 통과, 짝수는 000

| X | Y | 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 |

0  
1  
2  
3  
4  
5  
6  
7

|     | $F_2$ | $F_1$ | $F_0$ |
|-----|-------|-------|-------|
| 0   | 0     | 0     | 0     |
| 1 → | 0     | 0     | 1     |
| 2   | 0     | 0     | 0     |
| 3 → | 0     | 1     | 1     |
| 4   | 0     | 0     | 0     |
| 5 → | 1     | 0     | 1     |
| 6   | 0     | 0     | 0     |
| 7 → | 1     | 1     | 1     |



## 5.2.2 조합논리회로의 설계

### 3. 각 출력을 입력 변수의 함수로 나타내고 간소화

| X | Y | Z | $F_2$ | $F_1$ | $F_0$ |
|---|---|---|-------|-------|-------|
| 0 | 0 | 0 | 0     | 0     | 0     |
| 0 | 0 | 1 | 0     | 0     | 1     |
| 0 | 1 | 0 | 0     | 0     | 0     |
| 0 | 1 | 1 | 0     | 1     | 1     |
| 1 | 0 | 0 | 0     | 0     | 0     |
| 1 | 0 | 1 | 1     | 0     | 1     |
| 1 | 1 | 0 | 0     | 0     | 0     |
| 1 | 1 | 1 | 1     | 1     | 1     |

$$F_2 = \Sigma m(5, 7)$$

$$F_1 = \Sigma m(3, 7)$$

$$F_0 = \Sigma m(1, 3, 5, 7)$$

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

$$F_2 = XZ$$

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

$$F_0 = Z$$

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

$$F_1 = YZ$$



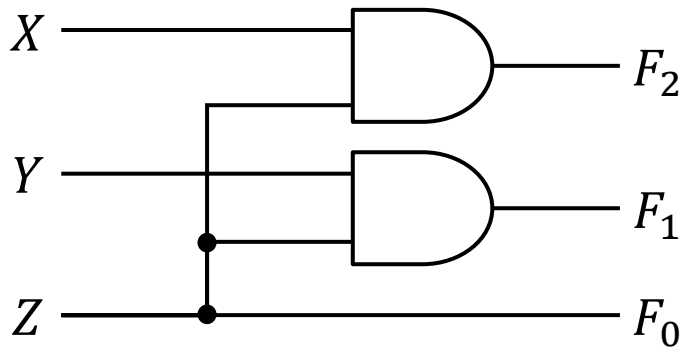
## 5.2.2 조합논리회로의 설계

### 4. 논리회로도 작성

$$F_2 = XZ$$

$$F_1 = YZ$$

$$F_0 = Z$$





## 6강. 조합논리회로(1)

### ➡ 제5장. 조합논리회로

# 5.3 기본 연산회로



## 5.3 기본 연산회로



### • 기본 연산회로

#### • 가·감·승·제, 사칙연산을 위한 산술연산회로

- 가산기: 덧셈 회로
- 감산기: 뺄셈 회로
- 승산기: 곱셈 회로
- 제산기: 나눗셈 회로



## 5.3.1 가산기

### • 가산기(Adder)

#### • 2진수 덧셈을 수행하는 조합논리회로

|                             |   |
|-----------------------------|---|
| 반가산기<br>(HA: Half<br>Adder) | <ul style="list-style-type: none"><li>• 두 개의 1비트 이진수를 더하는 논리회로</li><li>• 두 개의 비트를 입력 받아 합(Sum)과 자리올림(Carry)을 출력</li></ul>   |
| 전가산기<br>(FA: Full<br>Adder) | <ul style="list-style-type: none"><li>• 세 개의 1비트 이진수를 더하는 논리회로</li><li>• 자리올림도 고려하여 연산을 수행할 수 있음</li><li>• 따라서, 다중 비트 덧셈이 가능함</li><li>• 세 개의 비트를 입력 받아 합과 다음 자리올림을 출력</li></ul> |



## 5.3.1 가산기



### • 반가산기

두 비트의 덧셈( $S$ : 합,  $C$ : 올림)

$$\begin{array}{r} X \quad 0 \quad 0 \quad 1 \quad 1 \\ + Y \quad + 0 \quad + 1 \quad + 0 \quad + 1 \\ \hline C \ S \quad 0 \ 0 \quad 0 \ 1 \quad 1 \ 0 \end{array}$$

반가산기 진리표

| 입력  |     | 출력  |     |
|-----|-----|-----|-----|
| $X$ | $Y$ | $S$ | $C$ |
| 0   | 0   | 0   | 0   |
| 0   | 1   | 1   | 0   |
| 1   | 0   | 1   | 0   |
| 1   | 1   | 0   | 1   |

간소화를 위한 카르노 도표

| $X \backslash Y$ | 0 | 1 |
|------------------|---|---|
| 0                |   | 1 |
| 1                | 1 |   |

$$\begin{aligned} S &= \bar{X}Y + X\bar{Y} \\ &= X \oplus Y \end{aligned}$$

| $X \backslash Y$ | 0 | 1 |
|------------------|---|---|
| 0                |   |   |
| 1                |   | 1 |

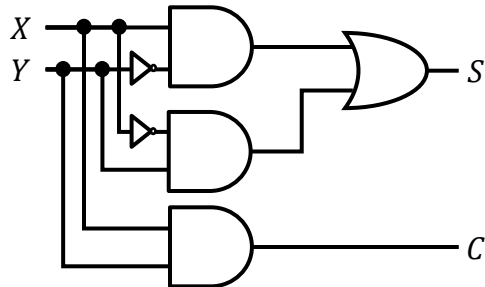
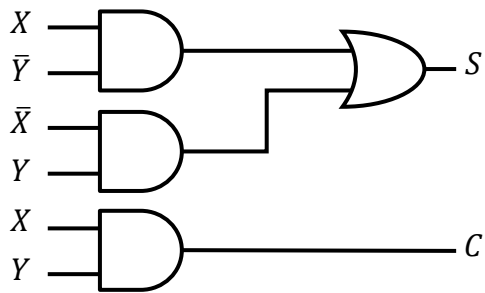
$$C = XY$$



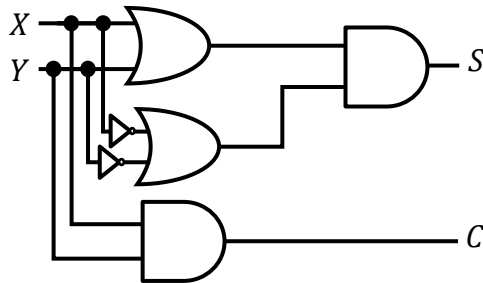
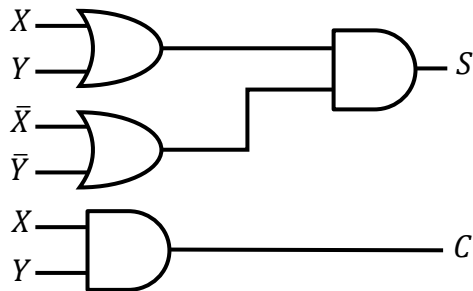
## 5.3.1 가산기

### • 여러 형태의 반가산기 논리회로도

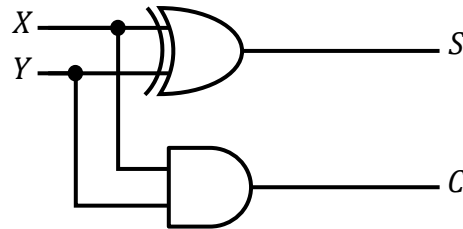
$$S = \bar{X}Y + X\bar{Y}$$
$$C = XY$$



$$S = (X + Y)(\bar{X} + \bar{Y})$$
$$C = XY$$



$$S = X \oplus Y$$
$$C = XY$$





## 5.3.1 가산기



### • 전가산기

세 비트의 덧셈( $S$ : 합,  $C$ : 올림)

$$\begin{array}{r}
 \begin{array}{cccc}
 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 1 & 1 \\
 C_{in} & + 0 & + 1 & + 0 & + 1 \\
 & \underline{0\ 0} & \underline{0\ 1} & \underline{0\ 1} & \underline{1\ 0} \\
 X & & & & \\
 + Y & & & & \\
 \underline{C\ S} & 1 & 1 & 1 & 1 \\
 & 0 & 0 & 1 & 1 \\
 & + 0 & + 1 & + 0 & + 1 \\
 & \underline{0\ 1} & \underline{1\ 0} & \underline{1\ 0} & \underline{1\ 1}
 \end{array}
 \end{array}$$

전가산기 진리표

| 입력  |     |              | 출력  |     |
|-----|-----|--------------|-----|-----|
| $X$ | $Y$ | $Z(=C_{in})$ | $S$ | $C$ |
| 0   | 0   | 0            | 0   | 0   |
| 0   | 0   | 1            | 1   | 0   |
| 0   | 1   | 0            | 1   | 0   |
| 0   | 1   | 1            | 0   | 1   |
| 1   | 0   | 0            | 1   | 0   |
| 1   | 0   | 1            | 0   | 1   |
| 1   | 1   | 0            | 0   | 1   |
| 1   | 1   | 1            | 1   | 1   |



## 5.3.1 가산기

### • 전가산기 카르노 도표

| 입력 |   |                        | 출력 |   |
|----|---|------------------------|----|---|
| X  | Y | Z (= C <sub>in</sub> ) | S  | C |
| 0  | 0 | 0                      | 0  | 0 |
| 0  | 0 | 1                      | 1  | 0 |
| 0  | 1 | 0                      | 1  | 0 |
| 0  | 1 | 1                      | 0  | 1 |
| 1  | 0 | 0                      | 1  | 0 |
| 1  | 0 | 1                      | 0  | 1 |
| 1  | 1 | 0                      | 0  | 1 |
| 1  | 1 | 1                      | 1  | 1 |

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

$$S = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$$

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

$$C = XY + XZ + YZ$$



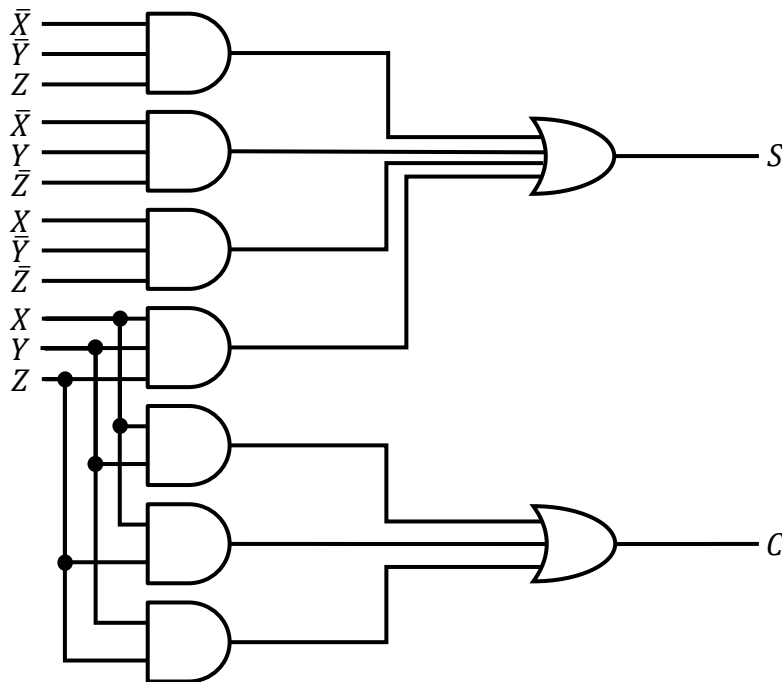
## 5.3.1 가산기



### • 전가산기의 논리회로도(1)

$$S = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$$

$$C = XY + XZ + YZ$$







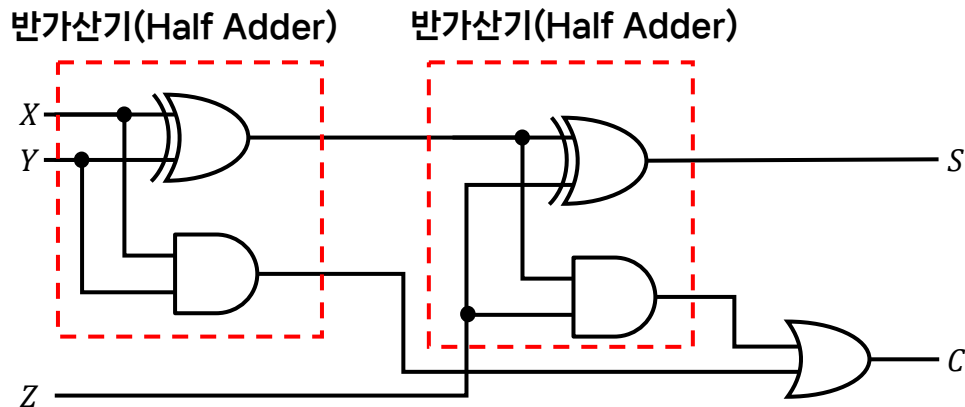
## 5.3.1 가산기



### • 전가산기의 논리회로도(1)

$$S = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ = X \oplus Y \oplus Z$$

$$C = XY + XZ + YZ = XY + Z(X \oplus Y)$$



2개의 반가산기와 1개의 OR 게이트로 구성

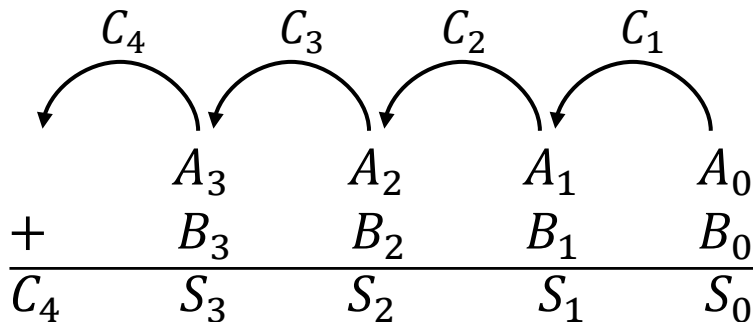


## 5.3.1 가산기



### • 직·병렬 가산기

- $n$  비트로 구성된 2진수의 가산 – 전가산기를 이용해 구현



직렬 방법(직렬 가산기): 회로구성비가 저렴

비트단위로 순차적 덧셈을 수행하여 느림

병렬 방법(병렬 가산기): 여러 개의 전가산기를 병렬로 연결

모든 비트를 동시에 더하여 속도가 빠름

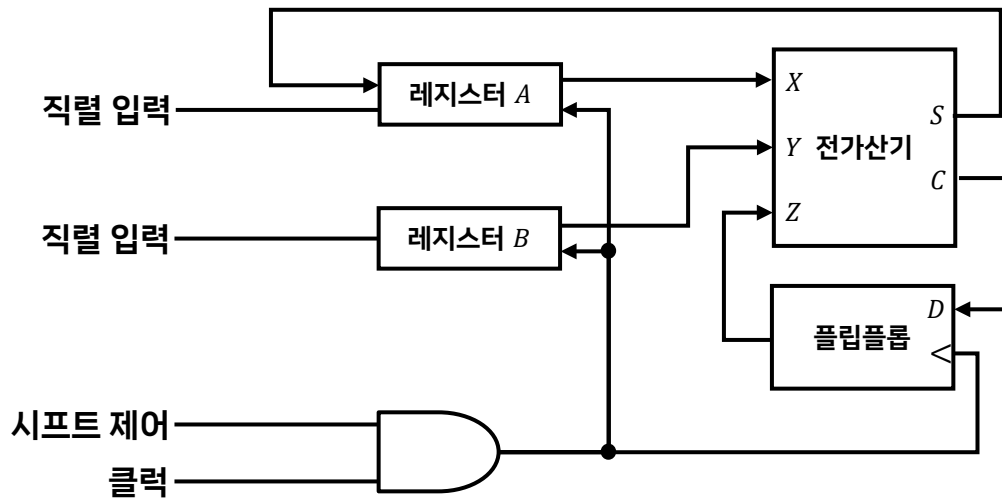




## 5.3.1 가산기

### • 직렬 가산기

#### • 전가산기와 저장요소로 구성



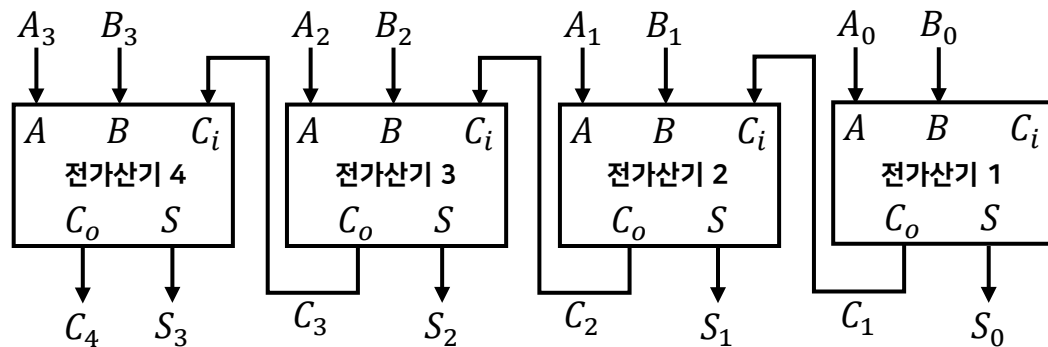


## 5.3.1 가산기



### • 4비트 병렬 가산기

- 전가산기를 연속적으로 연결하여 구성



- 위의 형태는 병렬 가산기 형태이지만, 여러 비트 값이 병렬 가산 되지는 않음
- 병렬 가산을 위해서 올림 수 미리보기 방식(Carry Look-Ahead) 등을 사용해 병렬 가산을 수행함



## 5.3.2 감산기

### • 감산기(Subtractor)

- 2진수의 뺄셈을 수행하는 조합논리회로

|                               |  |
|-------------------------------|--|
| 반감산기<br>(HS: Half Subtractor) | <ul style="list-style-type: none"><li>• 두 개의 1비트 이진수의 뺄셈을 수행하는 논리회로</li><li>• 두 개의 비트를 입력 받아 차이(Difference)와 빌림(Borrow)을 출력</li></ul>  |
| 전감산기<br>(FS: Full Subtractor) | <ul style="list-style-type: none"><li>• 세 개의 1비트 이진수의 뺄셈을 수행하는 논리회로</li><li>• 자리 빌림을 고려하여 연산을 수행할 수 있음</li><li>• 따라서, 다중 비트 뺄셈이 가능함</li><li>• 세 개의 비트를 입력 받아 차이와 새로운 자리 빌림을 출력</li></ul> |



## 5.3.2 감산기



### • 반감산기

두 비트의 뺄셈( $D$ : 차,  $B$ : 빌림 수)

|                    |                    |                    |                    |                    |
|--------------------|--------------------|--------------------|--------------------|--------------------|
| $X$                | 0                  | 0                  | 1                  | 1                  |
| $-Y$               | $-0$               | $-1$               | $-0$               | $-1$               |
| $\overline{B} \ D$ | $\overline{0} \ 0$ | $\overline{1} \ 1$ | $\overline{0} \ 1$ | $\overline{0} \ 0$ |

반가산기 진리표

| 입력  |     | 출력  |     |
|-----|-----|-----|-----|
| $X$ | $Y$ | $D$ | $B$ |
| 0   | 0   | 0   | 0   |
| 0   | 1   | 1   | 1   |
| 1   | 0   | 1   | 0   |
| 1   | 1   | 0   | 0   |

간소화를 위한 카르노 도표

|     |   |     |   |
|-----|---|-----|---|
|     |   | $Y$ |   |
|     |   | 0   | 1 |
| $X$ | 0 |     | 1 |
|     | 1 | 1   |   |

$$D = \bar{X}Y + X\bar{Y}$$

$$= X \oplus Y$$

|     |   |     |   |
|-----|---|-----|---|
|     |   | $Y$ |   |
|     |   | 0   | 1 |
| $X$ | 0 |     | 1 |
|     | 1 |     |   |

$$B = \bar{X}Y$$



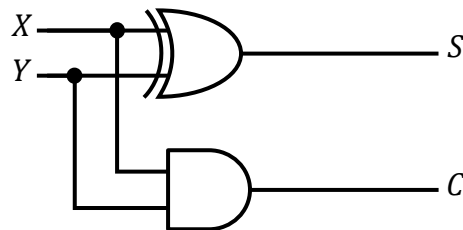
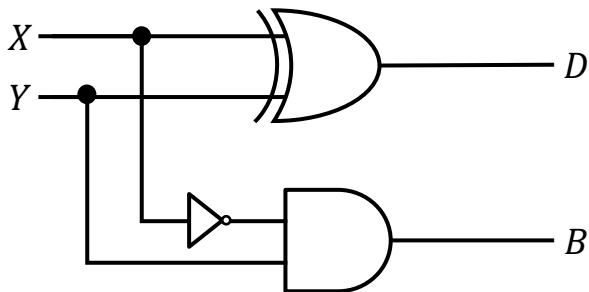
## 5.3.2 감산기



### • 반감산기의 논리회로도

$$D = \bar{X}Y + X\bar{Y} = X \oplus Y$$

$$B = \bar{X}Y$$



반가산기 논리회로도

$$S = X \oplus Y$$

$$C = XY$$



## 5.3.2 감산기



### • 전감산기

전감산기 진리표

| 입력  |     |                | 출력  |     |
|-----|-----|----------------|-----|-----|
| $X$ | $Y$ | $Z (= B_{in})$ | $D$ | $B$ |
| 0   | 0   | 0              | 0   | 0   |
| 0   | 0   | 1              | 1   | 1   |
| 0   | 1   | 0              | 1   | 1   |
| 0   | 1   | 1              | 0   | 1   |
| 1   | 0   | 0              | 1   | 0   |
| 1   | 0   | 1              | 0   | 0   |
| 1   | 1   | 0              | 0   | 0   |
| 1   | 1   | 1              | 1   | 1   |

전감산기 부울함수

$$D = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$$

$$= X \oplus Y \oplus Z$$

$$B = \bar{X}Y + \bar{X}Z + YZ$$

$$= \bar{X}Y + (\overline{X \oplus Y})Z$$





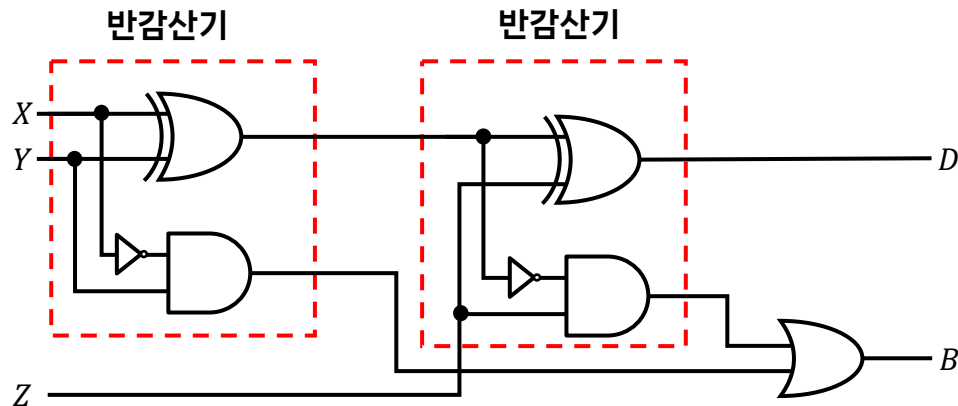
## 5.3.2 감산기



### • 전감산기의 논리회로도

$$D = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ = X \oplus Y \oplus Z$$

$$B = \bar{X}Y + \bar{X}Z + YZ = \bar{X}Y + (\bar{X} \oplus Y)Z$$



2개의 반감산기와 1개의 OR 게이트로 구성



## 5.3.3 병렬 가·감산기



### • 실제 컴퓨터에서는

- 2진 감산을 수행할 때 가산기를 사용함

- 사칙연산을 가산기와 시프트 연산 기반으로 수행함

- $X - Y$ 의 감산

- $Y$ 의 보수를 구해  $X$ 에 더함
- 1의 보수는 NOT을 이용해서 구할 수 있으며, 가산기에 입력 올림수로 1을 입력
- 비트 값을 반전하고 1을 더하여 2의 보수를 구할 수 있음

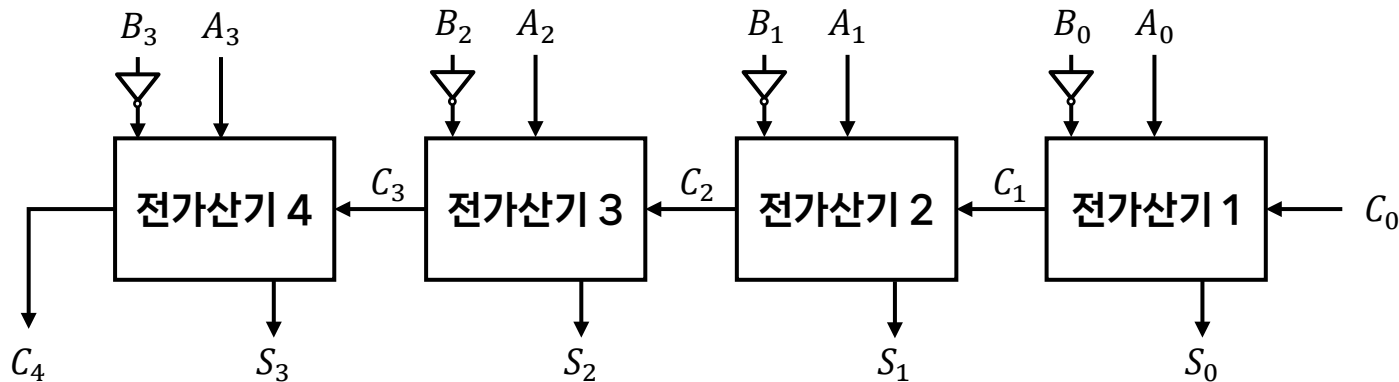


## 5.3.3 병렬 가·감산기



### • 4비트 병렬 감산기

$$\begin{aligned} A - B &= A + (B \text{의 } 2\text{의 보수}) \\ &= A + (\bar{B} + 1) \end{aligned}$$



병렬 가산기의  $B$ 단자와 가산기 사이에 NOT 게이트 추가(1의 보수)

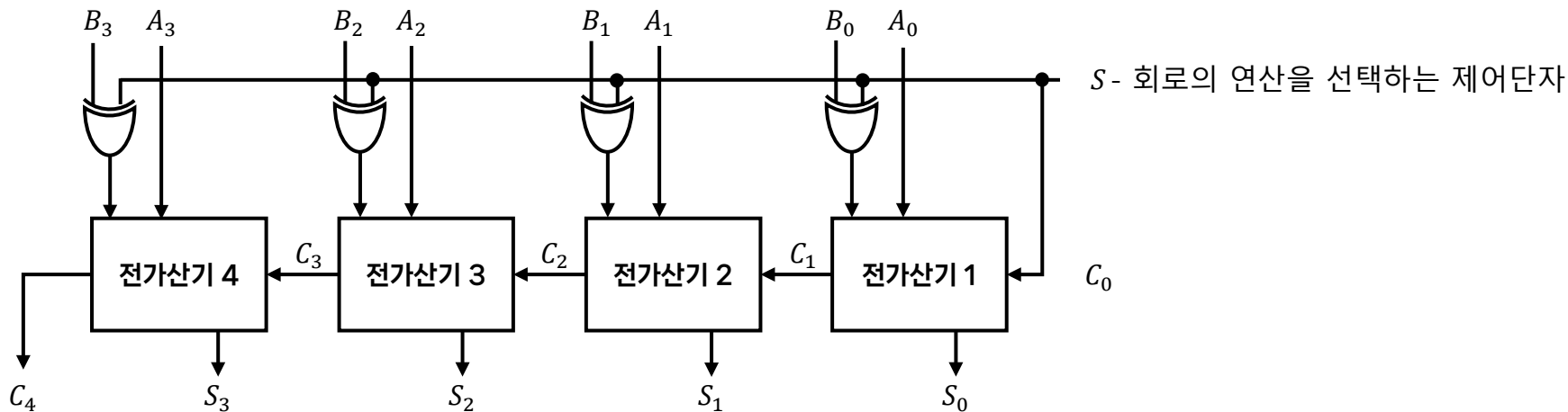
입력 올림 수  $C_0$ 는 1로 입력(2의 보수)



## 5.3.3 병렬 가·감산기

### • 4비트 가·감산기

- 덧셈과 뺄셈을 가산기만으로 수행



$S = 0$ : 가산연산 수행 (XOR의 출력이  $B \oplus 0 = B \cdot \bar{0} + \bar{B} \cdot 0 = B$ )

$S = 1$ : 감산연산 수행 (XOR의 출력이  $B \oplus 1 = B \cdot \bar{1} + \bar{B} \cdot 1 = \bar{B}$ )



# 내용 정리

Summary

Contents



## 6강 | 조합논리회로(1)



디지털 +  
논리회로

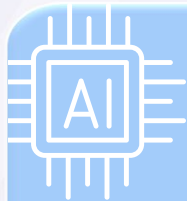


### 01 조합논리회로의 분석과 설계

### 02 기본 연산 회로

- 가산기
- 감산기
- 병렬가감산기

Digital Logic Circuits



다음시간에는

7강.

# 조합논리회로(2)