

전자계산기

정보처리산업기사

1장 / 논리 회로

정보처리산업기사

SECTION 38

볼 대수

불 대수

- 불 대수는 하나의 명제가 참 또는 거짓인가를 판단하는데 이용되는 수학적 방법으로 영국의 수학자(G. Boole)에 의해 개발되었다.
- 디지털 컴퓨터는 참과 거짓을 나타내는 1과 0의 두 가지 상태로만 표현하여 처리하는 2진 논리회로로 구성되었으므로, 이러한 논리회로를 간략화하여 표현할 때 불 대수가 사용된다.

AND

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

NOT

A	NOT A
0	1
1	0

불 대수의 기본 공식

- 교환법칙 : $A+B = B+A$, $A \cdot B = B \cdot A$
- 결합법칙 : $A+(B+C) = (A+B)+C$, $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- 분배법칙 : $A \cdot (B+C) = A \cdot B + A \cdot C$, $A + B \cdot C = (A+B) \cdot (A+C)$
- 멱등법칙 : $A+A = A$, $A \cdot A = A$
- 보수법칙 : $A+\bar{A} = 1$, $A \cdot \bar{A} = 0$
- 항등법칙 : $A + 0 = A$, $A+1 = 1$, $A \cdot 0 = 0$, $A \cdot 1 = A$
- 콘센서스 : $AB + BC + C\bar{A} = AB + C\bar{A}$
 $(A+B)(B+C)(C+\bar{A}) = (A+B)(C+\bar{A})$
- 드모르강 : $\overline{A+B} = \bar{A} \cdot \bar{B}$, $\overline{A \cdot B} = \bar{A} + \bar{B}$
- 복원법칙 : $\bar{\bar{A}} = A$

불 대수의 이해

- 불 대수에서는 한 개의 변수에 대입되는 값이 항상 0 또는 1이라는 것을 염두에 두고 생각하면 쉽게 결과를 알 수 있다. 무조건 기본 공식을 암기하려 하지 말고 변수에 대입될 수 있는 값을 생각해서 나올 수 있는 결과를 예측해 보세요.
- $A+0=A$: 불 대수이므로 A에 대입될 수 있는 값은 0 또는 1이다. 그러므로 만들어 질 수 있는 불식은 $1+0=1$, $0+0=0$ 뿐이다. 즉 A에 입력되는 값이 결과가 되므로 $A+0=A$ 가 된다.
- $A \cdot 0 = 0$: A에 대입될 수 있는 값은 0또는 1이므로 만들어질 수 있는 수식은 $1 \cdot 0=0$, $0 \cdot 0=0$ 뿐이다. 즉A에 무슨 값이 대입되어도 결과는 0이므로 $A \cdot 0 = 0$ 이다.
- $A+\bar{A} = 1$: A에 대입될 수 있는 값은 0또는 1이다. A가 1이라면 \bar{A} (A의 부정)은 0이되고 A가 0이라면 \bar{A} 는 1이 되어 $1+0=1$, $0+1=1$, 즉 항상 1이 되므로 $A+\bar{A} = 1$ 이다.
- $A \cdot \bar{A}=0$: A에 대입될 수 있는 값은 0또는 1이다. A가 1이라면 \bar{A} 는 0이되고 A가 0이라면 \bar{A} 는 1이 되어 $1 \cdot 0 = 0$, $0 \cdot 1 = 0$, 즉 항상 0이 되므로 $A \cdot \bar{A} = 0$ 이다.

논리식의 간소화

- 논리식의 간소화가 왜 필요할까?
- 합의 곱 표현을 곱의 합 표현으로 변환한다.
- 공통 인수를 뽑아 묶는다.
- 멍등법칙, 보수법칙, 항등법칙 등의 기본 공식 형태로 유도하여 줄여 나간다.
- 카르노 맵(카르노 도) 이용하기
 - 카르노 맵은 설계된 논리식을 도표로 표현하여 최소화하는 방법이다.
 - 카르노 맵은 변수(입력선)의 개수에 따라 표의 크기가 달라지며 칸의 위치에 따라서 각 칸의 불 함수가 정해진다.

카르노 맵(카르노 도) 이용하기

$\diagdown BC$	$\overline{B}\overline{C}$	$\overline{B}C$	$B\overline{C}$	BC
\overline{A}	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
A	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

$\diagdown B$	\overline{B}	B
\overline{A}	$\overline{A}\overline{B}$	$\overline{A}B$
A	$A\overline{B}$	AB

$\diagdown BC$	00	01	11	10
0	0	1	3	2
1	4	5	7	6

$\diagdown B$	0	1
0	0	1
1	2	3

출처 : <https://slidesplayer.org/slide/14821051/>









기출문제

SECTION 39

논리 게이트

논리 게이트

논리 게이트는 논리회로를 구성하는 기본적인 전자소자로, 0또는 1을 입력과 출력으로 사용한다.

게이트	기호	의미	진리표	논리식															
AND		입력신호가 모두 1일 때 1출력	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	$Y = A \cdot B$ $Y = AB$
A	B	Y																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OR		입력신호 중 1개만 1이어도 1출력	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	$Y = A + B$
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NOT		입력된 정보를 반대로 변환하여 출력	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0	$Y = A'$ $Y = \overline{A}$									
A	Y																		
0	1																		
1	0																		
BUFFER		입력된 정보를 그대로 출력	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	Y	0	0	1	1	$Y = A$									
A	Y																		
0	0																		
1	1																		
NAND		NOT + AND, 즉 AND의 부정	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	$Y = \overline{A \cdot B}$ $Y = \overline{AB}$
A	B	Y																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
NOR		NOT + OR, 즉 OR의 부정	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	$Y = \overline{A + B}$
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
XOR		입력신호가 모두 같으면 0, 한 개라도 틀리면 1출력	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	$Y = A \oplus B$ $Y = AB + \overline{A}\overline{B}$
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
XNOR		NOT + XOR, 즉 XOR의 부정	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1	$Y = A \odot B$ $Y = \overline{A \oplus B}$ $Y = AB + \overline{A}\overline{B}$
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

출처 : https://m.blog.naver.com/PostView.nhn?blogId=pusna25&logNo=100112058544&proxyReferer=https%3A%2F%2Fwww.google.com%2F&view=img_3

기출문제

SECTION 40

조합논리회로

반가산기 / 전가산기

조합논리회로

- 조합논리회로는 임의의 시간에서의 출력이 이전의 입력에는 관계없이 현재의 입력조합(0 또는 1)으로부터 직접 결정되는 논리회로이다.
- 이에 반해 순서논리회로는 외부로부터의 입력과 현재 상태에 따라 출력이 결정된다.
- 조합논리회로의 종류 : 반가산기, 전가산기, 병렬가산기, 반감산기, 전감산기, 디코더, 인코더, 멀티플렉서, 디멀티플렉서, 다수결회로, 비교기 등

논리회로 설계 방법

- 논리소자를 이용하여 입력되는 값으로 원하는 값을 출력하는 조합논리회로를 설계할 수 있다. 논리회로 설계 방법을 맞춰 2진수 2Bit를 더하는 반가산기를 설계해 본다.
- 1. 회로의 기능을 진리표로 나타낸다.
 - 반가산기는 2진수 2Bit를 입력받아 합과 자리올림수를 계산하는 조합논리회로이다. 입력 신호 A, B가 있을 때 출력으로 다음과 같은 합(SUM)과 자리올림(Carry)이 발생한다.

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- 2. 진리표로부터 논리식을 구한다.

논리회로 설계 방법

3. 논리식을 간소화한다.

$$\text{Sum} = \bar{A}B + A\bar{B} = A \oplus B$$

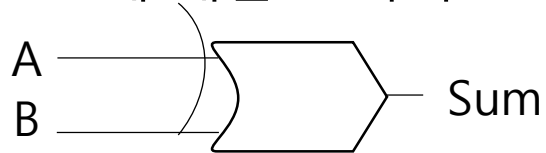
$$\text{Carry} = A \cdot B$$

◆ 카르노 맵을 이용하여 간략화 할 수도 있다.

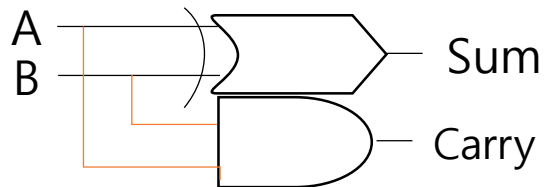
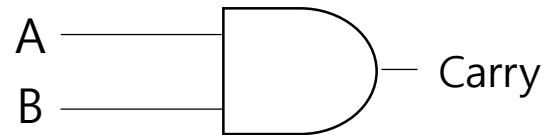
4. 간소화된 논리식을 논리소자로 그려서 연결시킨다.

$$\text{Sum} = \bar{A}B + A\bar{B} = A \oplus B \quad \text{Carry} = A \cdot B$$

Sum에 대한 논리회로



Carry에 대한 논리회로



두개를 합친 논리회로

반가산기(HA, Half Adder) (1/3)

- 반가산기는 아래 그림처럼 1Bit짜리 2진수 두개를 덧셈한 합(S)과 자리올림수 (C)를 구하는 조합논리회로이다.

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \quad 0 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 0 \quad 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 0 \quad 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 1 \quad 0 \end{array}$$

$$\begin{array}{r} A \\ + B \\ \hline C \quad S \end{array}$$

반가산기(HA, Half Adder) (2/3)

- 진리표

입력 A,B가 있을때 출력으로 S(합)와 C(자리올림) 가 있다.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

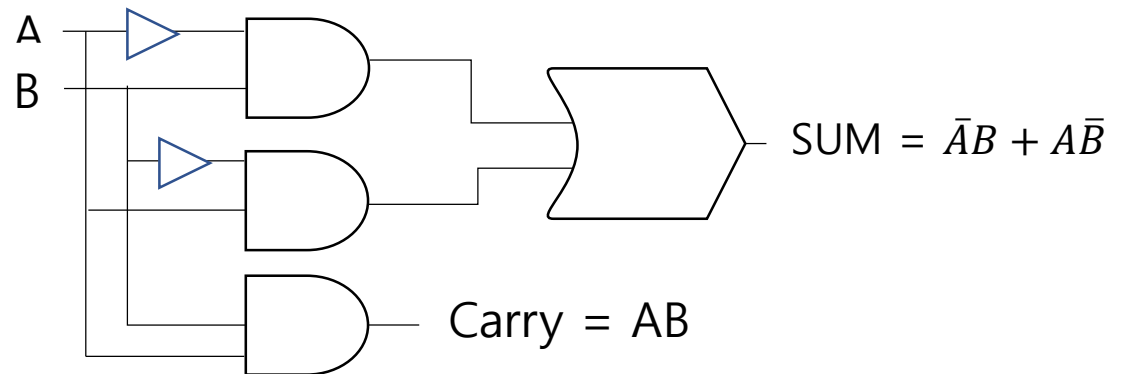
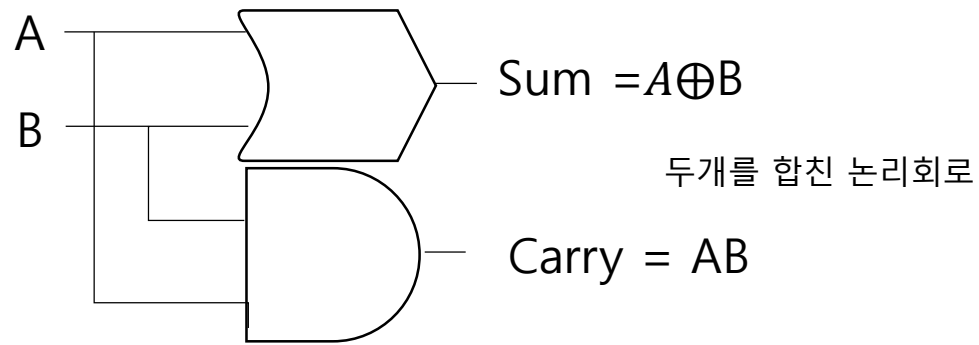
반가산기 (HA, Half Adder) (3/3)

논리식

출력이 1이 되는 항에 대해서 입력되는 값을 AND 연산하고 각각의 항을 OR 연산한다. 간략화가 가능한 경우 간략화한다.

$$C = A \cdot B$$

$$S = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$$



전가산기(FA, Full Adder) (1/3)

- 전가산기는 뒷자리에서 올라온 자리올림수(C_i)를 포함하여 1Bit 크기의 2진수 3자리를 더하여 합(S_i)과 자리올림수(C_{i+1})를 구하는 회로이다.

	0		1		0		1		0		1		0		1		C_i
	0		0		0		0		1		1		1		1		A_i
+	0	+	0	+	1	+	1	+	0	+	0	+	1	+	1	+	B_i
<hr/>		<hr/>		<hr/>		<hr/>		<hr/>		<hr/>		<hr/>		<hr/>		<hr/>	
	0		0		0		1		0		1		1		0		C_{i+1}
	0		1		1		0		1		0		1		1		S_i

전가산기(FA, Full Adder) (2/3)

진리표

입력으로 A,B 그리고 전 단계의 자리올림수가 C_i 가 있을 때 출력으로 합 S와 자리올림수 C_{i+1} 가 있다.

X	Y	Z	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

$$\begin{aligned} S &= \bar{X} \bar{Y} C_{in} + \bar{X} Y \bar{C}_{in} + X \bar{Y} \bar{C}_{in} + XY C_{in} \\ &= \bar{X} (\bar{Y} C_{in} + Y \bar{C}_{in}) + X (\bar{Y} \bar{C}_{in} + Y C_{in}) \\ &= \bar{X} (Y \oplus C_{in}) + X (\bar{Y} \oplus \bar{X}) \\ &= X \oplus (Y \oplus C_{in}) \end{aligned}$$

$$\begin{aligned} C_{out} &= \bar{X} Y C_{in} + X \bar{Y} C_{in} + XY \bar{C}_{in} + XY C_{in} \\ &= C_{in} (\bar{X} Y + X \bar{Y}) + XY (\bar{C}_{in} + C_{in}) \\ &= C_{in} (X \oplus Y) + XY \end{aligned}$$

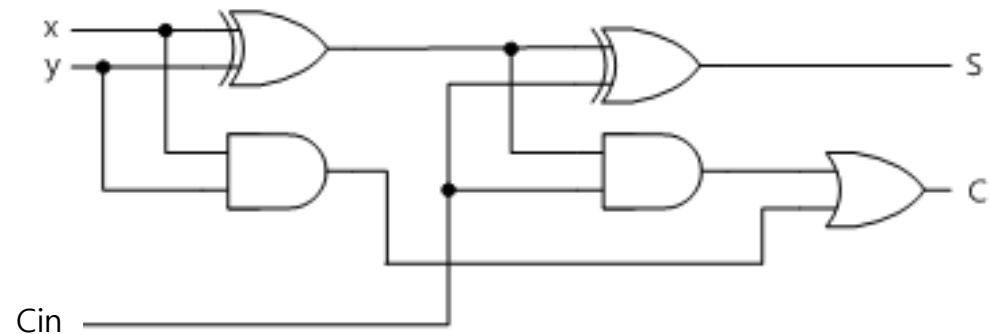
전가산기(FA, Full Adder) (3/3)

회로

전가산기는 두 개의 반가산기(HA)와 한 개의 OR Gate로 구성된다.

$$\begin{aligned} S &= \bar{X}\bar{Y}C_{in} + \bar{X}Y\bar{C}_{in} + X\bar{Y}\bar{C}_{in} + XYC_{in} \\ &= \bar{X}(\bar{Y}C_{in} + Y\bar{C}_{in}) + X(\bar{Y}\bar{C}_{in} + YC_{in}) \\ &= \bar{X}(Y \oplus C_{in}) + X(\bar{Y} \oplus \bar{C}_{in}) \\ &= X \oplus (Y \oplus C_{in}) \end{aligned}$$

$$\begin{aligned} C_{out} &= \bar{X}YC_{in} + X\bar{Y}C_{in} + XY\bar{C}_{in} + XYC_{in} \\ &= C_{in}(\bar{X}Y + X\bar{Y}) + XY(\bar{C}_{in} + C_{in}) \\ &= C_{in}(X \oplus Y) + XY \end{aligned}$$

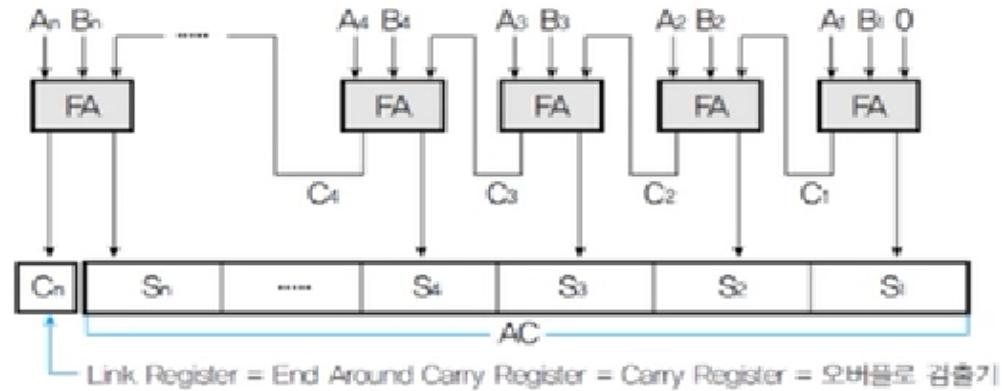


SECTION 41

기타 조합논리회로

병렬가산기(PA, Parallel Adder)

- 병렬가산기는 n Bit로 된 2진수 A, B 에 대한 덧셈을 n 개의 전가산기(FA)를 이용하여 구현한 실질적인 가산기이다.
- 전파지연을 줄이기 위해 Carry Look Ahead를 사용한다.



반감산기 (HS, Half Subtract)

- 반감산기는 1Bit짜리 2진수 2자리에 대한 감산을 하는 회로이다.

$$\begin{array}{r} 0 \\ - 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ - 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ - 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ - 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} A \\ - B \\ \hline D \end{array}$$

A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

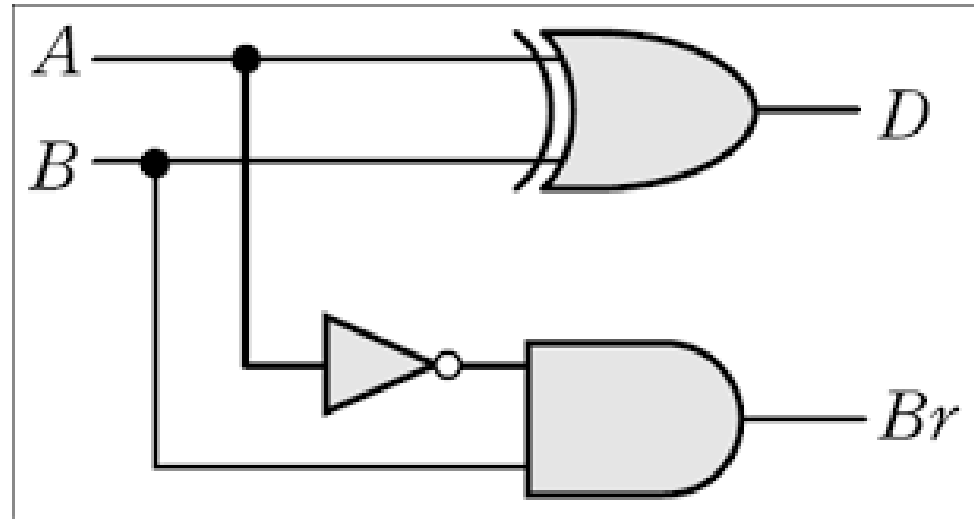
반감산기 (HS, Half Subtract)

논리식

출력이 1이 되는 항에 대해서 입력되는 값을 AND 연산하고, 각각의 항을 OR 연산한다. 간략화가 가능한 경우 간략화한다.

$$B_0 = \bar{A} \cdot B$$

$$D = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$$



디코더 (Decoder)

- n Bit의 Code화된 정보를 그 Code의 각 Bit 조합에 따라 2^n 개의 출력으로 번역하는 회로이다.
- 명령어의 명령부나 번지를 해독할 때 사용하는 회로로 주로 AND 게이트로 구성되어 있다.

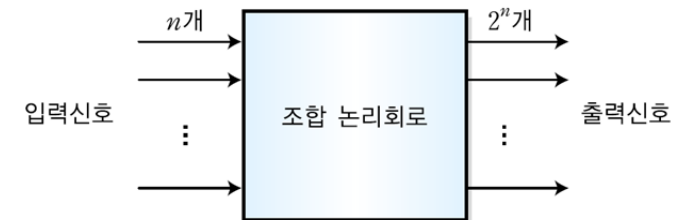
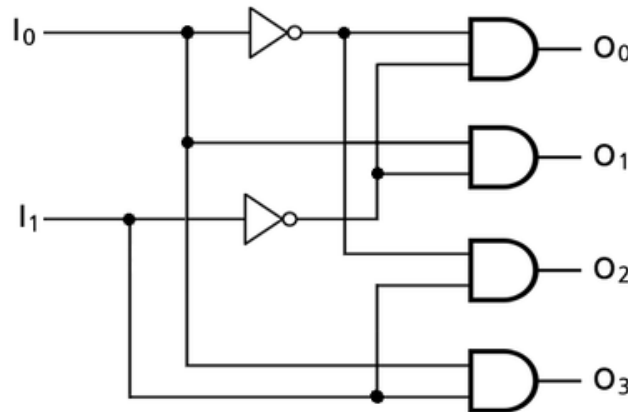
X	Y	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$D_0 = \bar{A} \cdot \bar{B}$$

$$D_1 = \bar{A} \cdot B$$

$$D_2 = A \cdot \bar{B}$$

$$D_3 = A \cdot B$$



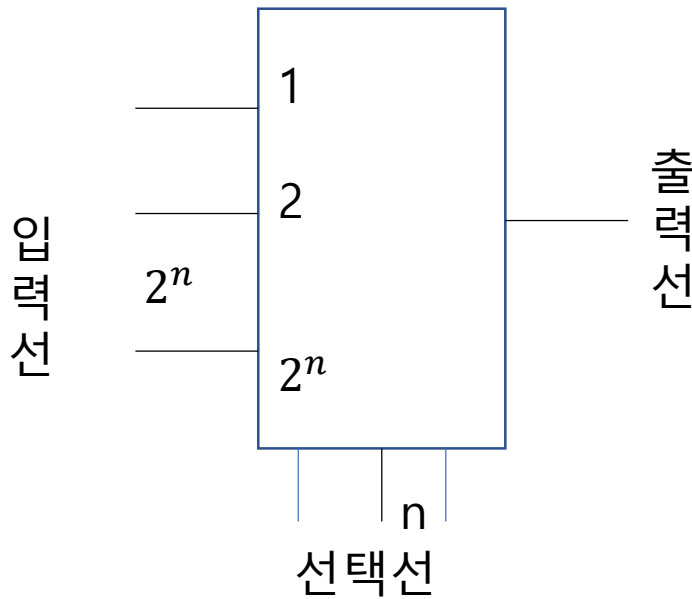
인코더 (Encoder)

- 인코더는 디코더의 반대 기능을 한다
- 2^n 개의 입력선으로 입력된 값을 n 개의 출력선으로 코드화해서 출력하는 회로이다.



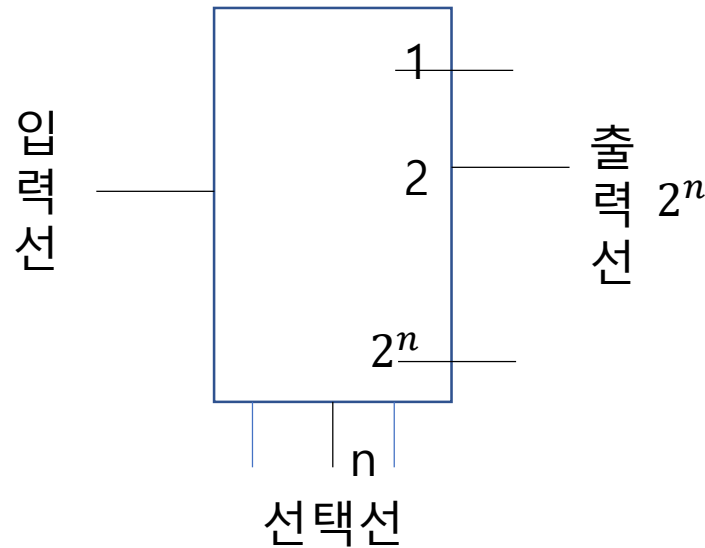
멀티플렉서(MUX, Multiplexer)

- 멀티플렉서는 2^n 의 입력선 중 1개를 선택하여 그 선으로부터 입력되는 값을 1개의 출력선으로 출력시키는 회로이다.
- 2^n 개의 입력선 중 1개의 선을 선택하기 위해 n 개의 선택선(Select Line)을 이용한다.
- 버스(bus) 구성에 사용



디멀티플렉서(DeMUX, DeMultiplexer)

- 디멀티플렉서는 1개의 입력선으로 들어오는 정보를 2^n 개의 출력선 중 1개를 선택하여 출력하는 회로이다.
- 2^n 개의 출력선 중 1개의 선을 선택하기 위해 n 개의 선택선을 이용한다.



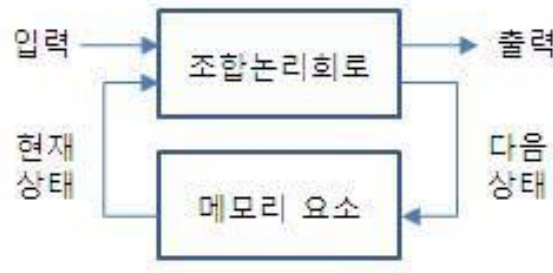
기출문제

SECTION 42

순서 논리 회로

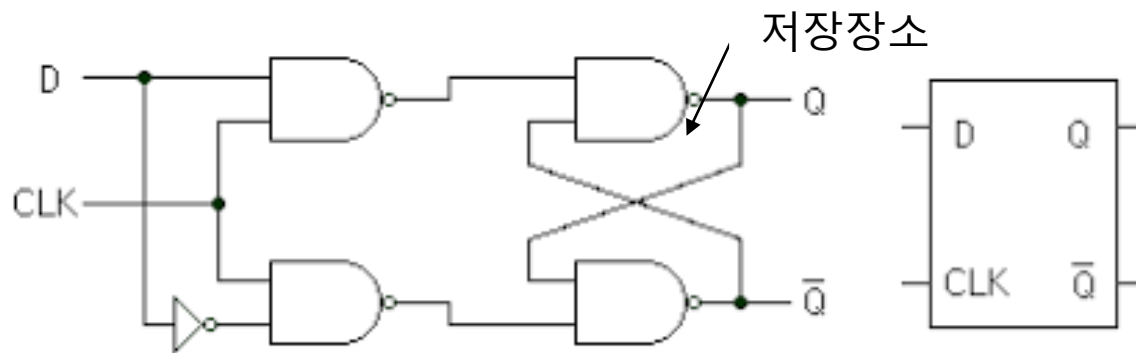
순서논리회로의 특징

- 순서논리회로는 외부로부터의 입력과 현재 상태에 따라 출력이 결정되는 회로이다. 논리 게이트 외에 메모리 요소와 귀환 기능을 포함하므로 기억기능이 있다.
- 순서논리회로의 출력은 입력 상태와 메모리 요소들의 상태에 따라 값이 결정되므로 언제나 일정한 값을 갖지 않는다
- 순서 논리 회로는 플립플롭과 논리 게이트로 구성된다.
- 순서논리회로는 신호의 타이밍에 따라 동기식과 비동기식으로 나뉘어진다.
- 대표적인 순서논리회로에는 플립플롭, 카운터, 레지스터, RAM, CPU등이 있다.



플립플롭(FF, Flip_Flop)의 특징 (1/2)

- 플립플롭은 전원이 공급되고 있는 한, 상태의 변화를 위한 외부 신호가 발생할 때 까지 현재의 상태를 그대로 유지하는 논리회로이다.
- 플립플롭 한 개가 1Bit를 구성하는 2진 셀 이다.
- 반도체 기억장치에서 2진수 1자리값을 기억하는 메모리 소자이다.
- 플립플롭은 레지스터, RAM, 카운터 등을 구성하는 기본 소자이다.
- 기본적인 플립플롭은 두개의 NAND 또는 NOR게이트를 이용하여 구성한다.
- 플립플롭의 종류 : RS-FF , D-FF, JK-FF, T-FF, RST-FF, 마스터-슬레이브 FF

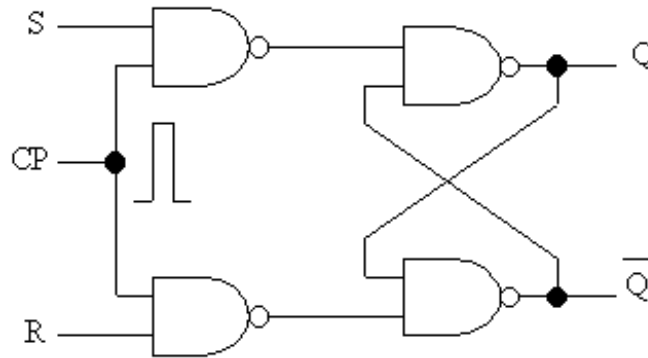


RS 플립플롭(Reset-Set FF)

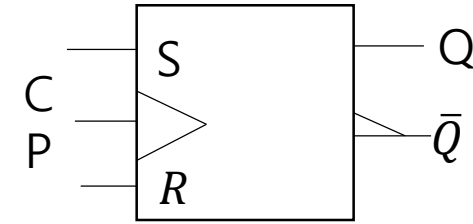
- RS플립플롭은 모든 플립플롭의 기본으로, S와 R선의 입력을 조절하여 임의의 Bit값을 그대로 유지시키거나, 무조건 0또는 1의 값을 기억시키기 위해서 사용되는 플립플롭이다.

SR 플립플롭

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	X



[RS 플립플롭]

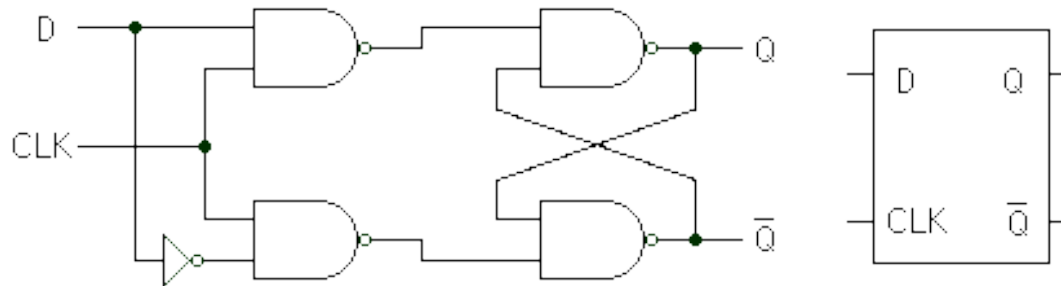


D 플립플롭

- D 플립플롭은 RS FF의 R선에 인버터를 추가하여 S선과 하나로 묶어서 입력 선을 하나만 구성한 플립플롭이다.
- D 플립플롭은 입력하는 값을 그대로 저장하는 기능을 수행한다.

D 플립플롭

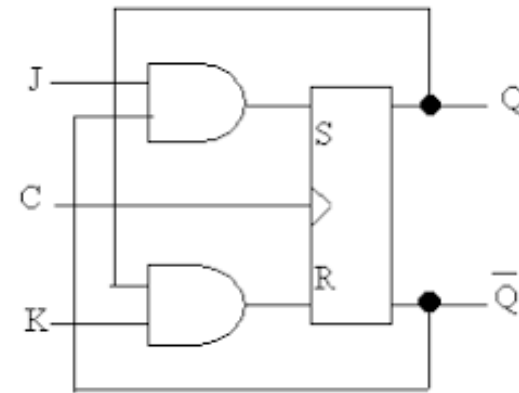
D	Q(t+1)
0	0
1	1



JK 플립플롭

- JK 플립플롭은 RS FF에서 $S=R=1$ 일때 동작되지 않는 결점을 보완한 플립플롭이다.
- RS FF의 입력선 S와 R에 AND 게이트 2개를 추가하여 JK FF의 입력선 J와 K로 사용한다.
- 모든 플립플롭의 기능을 포함시킨다.

J	K	$Q(t+1)$	설명
0	0	$Q(t)$	현 상태가 그대로 출력
0	1	0	0을 출력(Reset)
1	0	1	1을 출력(Set)
1	1	$Q'(t)$	Complement(보수)

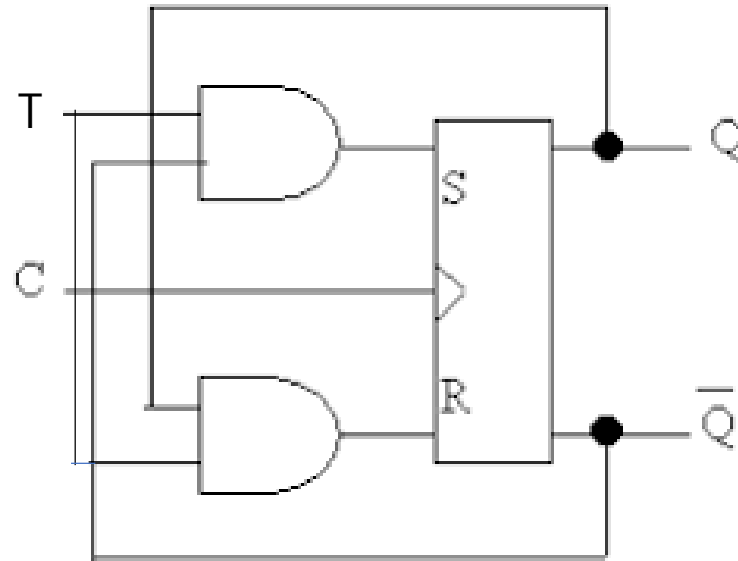


T 플립플롭

- T 플립플롭은 JK FF의 두 입력선을 묶어서 한 개의 입력선으로 구성한 플립플롭이다.
- $T=0$ 인 경우는 변화가 없고, $T=1$ 인 경우에 현재의 상태를 토글시킨다. 즉 원 상태와 보수 상태의 두 가지 상태로만 서로 전환된다.

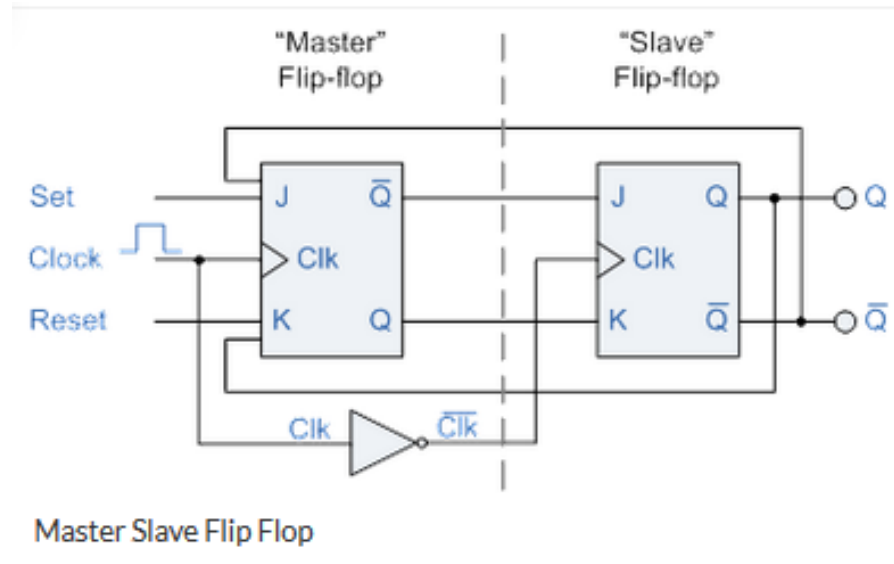
T 플립플롭

T	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$



마스터 – 슬레이브 플립플롭

- 마스터-슬레이브 플립플롭은 출력 측의 일부가 입력 측에 궤환(feedback) 되어 유발되는 레이스 현상을 없애기 위해 고안된 플립플롭이다.
- 마스터-슬레이브 플립플롭은 두 개의 플립플롭으로 구성되는 데 한쪽 회로가 마스터이고 다른 한쪽이 슬레이브의 위치에 있어 마스터-슬레이브 플립플롭이라 한다.



기출문제