tb_elec_engineer@naver.com

9장 멀티플렉서, 디코더, 프로그래머블 논리소자

→ 회로를 만들 때 자주 쓰이는 기능들을 미리 만들어 회로를 만들 때 쉽게 사용할 수 있도록 함

9.1 개요

소규모 집적회로 : NAND, NOR, AND, OR, 인버터

* 중규모 집적회로 : 덧셈기, 멀티플렉서, 디코더, 레지스터, 카운터

대규모 집적회로 : 메모리, 마이크로프로세서

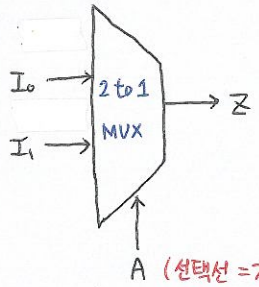
9.2 멀티플렉서 (MUX)

멀티플렉서 = 데이터 선택기 = MUX

→ 여러개의 입력 중 하나를 선택하는 기능을 함

① 2 to 1 MUX (= 2×1 MUX)

: 입력 2개, 출력 1 인 MUX



그림설명

⇒ 2개의 입력 I_0, I_1 중 하나를 선택하여 출력으로 보내는데

선택선 A가 0 이면 I_0을 출력

선택선 A가 1 이면 I_1을 출력한다.

∴ A가 0 이냐 1 이냐에 따라 출력 Z가 결정됨

진리표

A	Z
0	I_0
1	I_1

논리식 $Z = A'I_0 + AI_1$

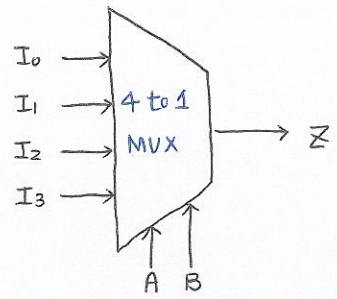
⊛ 논리식을 해석하면 $A=0 \rightarrow Z=I_0$

$A=1 \rightarrow Z=I_1$

tb_elec_engineer@naver.com

② 4 to 1 MUX = 4×1 MUX

: 입력 4개, 출력 1개인 MUX



⇒ I_0, I_1, I_2, I_3 의 입력 4개 중 A, B의 제어선에 따라 출력 Z가 결정된다.

A=0, B=0 이면 I_0이 출력
A=0, B=1 이면 I_1 출력
A=1, B=0 이면 I_2 출력
A=1, B=1 이면 I_3 출력

진리표

A	B	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

논리식

$$Z = A'B'I_0 + A'BI_1 + AB'I_2 + ABI_3$$

※ 논리식을 해석하면 위 내용과 같다

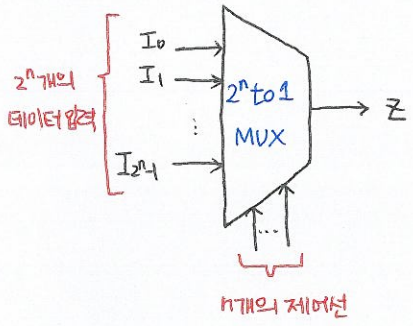
⋮

8 to 1 MUX 도 이와같이 적용하여 만들 수 있다

③ 위 내용을 일반화 하기

2^n to 1 MUX

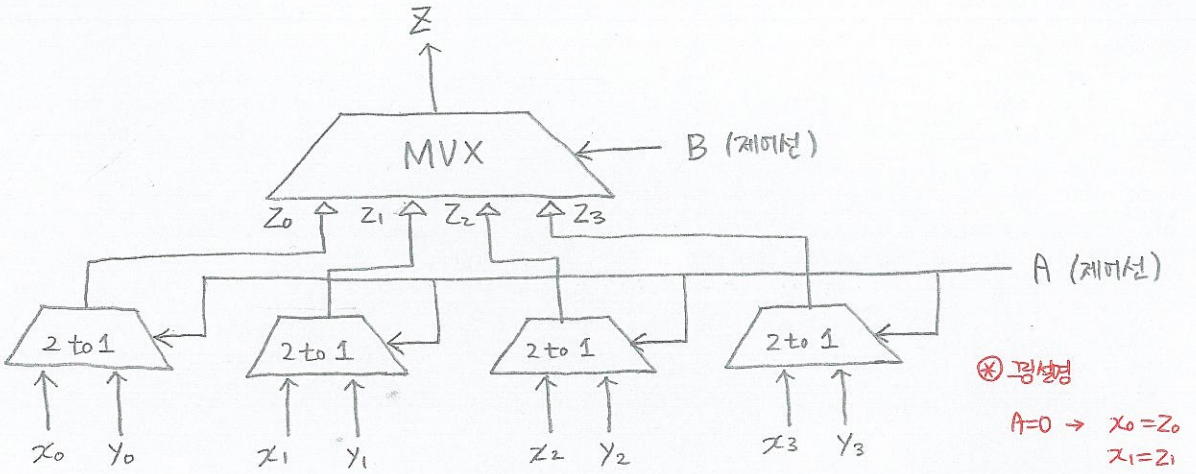
→ 2^n 개의 입력 과 n개의 제어선 (선택선) 을 가지는 멀티플렉서



⊕ X, Y 의 두 개의 입력을 갖는 MUX 설계하기

tb_elec_engineer@naver.com

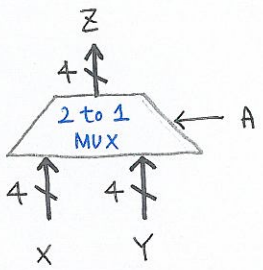
근데 X, Y 가 4비트



※ 그림설명

A=0 → $X_0 = Z_0$
$X_1 = Z_1$
$X_2 = Z_2$
$X_3 = Z_3$

⇩ 줄여서 표현하기



진한선은 "여러 개를 보낸다" 의 의미

여러 개의 신호가 있는데 그 신호들이 묶여 하나의 의미를 나타낼 때 간단히 나타내기 위해 \ (사선) 표시

4 ↕ 는 "4비트를 갖는다" 의 의미이다

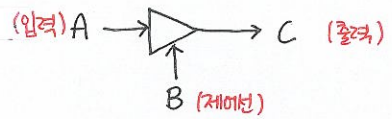
A=0 → Z=X

A=1 → Z=Y

9.3 3-상태 버퍼 (= 3-state Buffer)

tb_elec_engineer@naver.com

3-상태 버퍼는 들어오는 신호를 증폭시키기 위해 사용한다

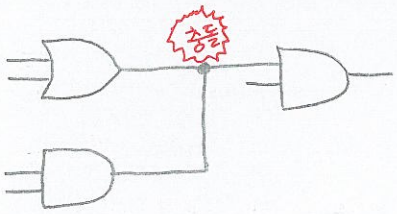


A 신호가 들어올 때 제어선 B가 0 이냐 1 이냐에 따라
출력 C가 결정된다.

B=1 이면 연결된 상태로 A 입력이 C로 전달
B=0 이면 개방회로 상태가 되어 A 입력이 전달되지 않는다. (High Impedance / 개방회로 상태라고 함)

문제) 2개 이상의 게이트 아 다른 소자들의 출력이 직접 연결되면 정상상태에서
논리회로가 동작하지 않고 게이트가 손상될 수 있음

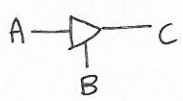
예)



문제해결) 3-상태 버퍼를 사용하여 2개 이상의 게이트 아 다른 소자들의 출력을 연결

3-상태 버퍼 종류

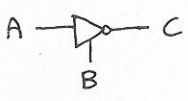
①



	B (제어선)	A (입력)	C (출력)
끊어짐	0	0	Z
	0	1	Z
연결	1	0	0
	1	1	1

⊛ Z = 높은 임피던스 상태
: 높은 저항과 임피던스로 전류가 흐르지 않을 때

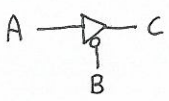
②



B	A	C
0	0	Z
0	1	Z
1	0	1
1	1	0

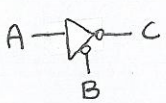
입력이 반전됨

③



B	A	C
0	0	0
0	1	1
1	0	Z
1	1	Z

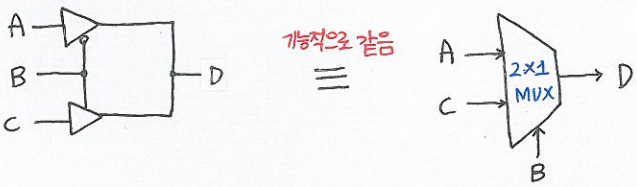
④



B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

tb_elec_engineer@naver.com

⊕ 3상 버퍼를 이용하여 MUX 구현할 수 있음



B=1 이면 아래 버퍼가 동작가능

∴ C=D

A=0 ⇒ 충돌이 일어나지 않음

B=0 이면 위 버퍼가 동작가능 (원래는 충돌이 일어날 수 있는 상황이지만

∴ A=D 3상 버퍼를 이용하여 충돌이 일어나지 않았다)

C=0

∴ 두개의 출력을 연결해줄 때 3상 버퍼를 이용

9.4 디코더와 인코더

디코더란?

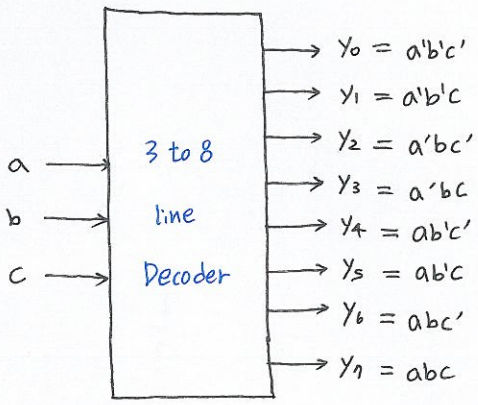
① 코드 변환기 : (예) 2진수 코드 입력에 해당하는 10진수를 출력

② 최소항 생성기 : n개의 입력 변수에 대해 2^n개의 최소항을 생성

↳ 진리표에서 한 행만 1이 되고 나머지는 0이 되는 곱 ⇒ 디코더 안에 AND 게이트가 있음을 알 수 있음

3 to 8 line Decoder

: 3개 입력과 2^3 개의 최소항을 갖는 디코더



a	b	c	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

⊛ 디코더 응용

Sum = $\sum m$ (1,2,4,7) → 뽑아서 더하면 됨

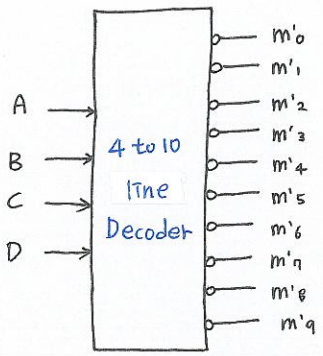
⊛ y_0은 000 일 때 출력됨

입력값의 각 조합에 대해 출력선 중 하나만 1이 된다

tb_elec_engineer@naver.com

반전 출력을 갖는 4 x 10 디코더

: 4개 의 입력과 2^4개의 최대항을 갖는 디코더



BCD 입력	10진수 출력
A B C D	0 1 2 3 4 5 6 7 8 9
0 0 0 0	0 1 1 1 1 1 1 1 1 1
0 0 0 1	1 0 1 1 1 1 1 1 1 1
0 0 1 0	1 1 0 1 1 1 1 1 1 1
⋮	⋮
1 0 0 1	1 1 1 1 1 1 1 1 1 0
1 0 1 0	1 1 1 1 1 1 1 1 1 1
S	//
1 1 1 1	1 1 1 1 1 1 1 1 1 1

4개의 입력을 가지는데 출력은 10개?
BCD 해독기로 디코더가 역할을 함
BCD 는 0~9 까지 가능

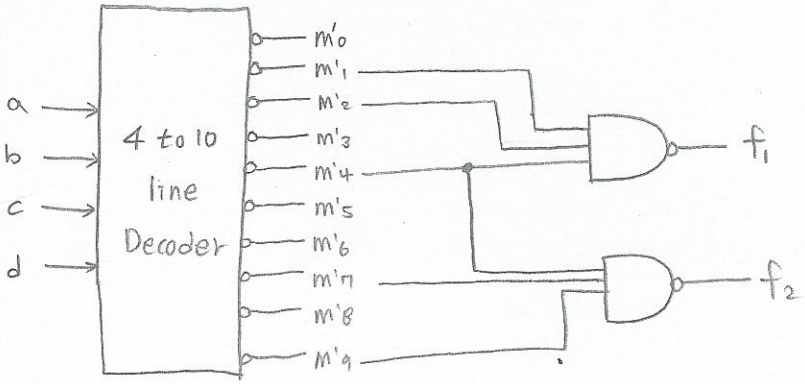
BCD 코드는 10~15까지는
생기지 않는다

위 디코더를 사용하여 원하는 함수를 구현할 수 있다.

ex) $f_1(a,b,c,d) = m_1 + m_2 + m_4$
$f_2(a,b,c,d) = m_4 + m_7 + m_9$
를 NAND 게이트를 사용하여 생성하기

⇒ $f_1 = m_1 + m_2 + m_4 = (m'_1\, m'_2\, m'_4)'$

$f_2 = m_4 + m_7 + m_9 = (m'_4\, m'_7\, m'_9)'$



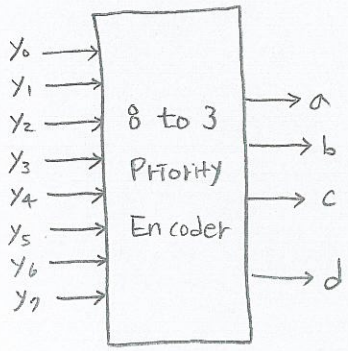
tb_elec_engineer@naver.com

인코더란? 디코더의 반대

하나만 1 인 입력에 대하여 2진수 값을 출력

→ 동시에 하나 이상의 입력이 1이면 제일 오른쪽에 있는 1에 대해 출력을 결정

8 to 3 우선 순위 인코더



y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	a	b	c	d
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
x	1	0	0	0	0	0	0	0	0	1	1
x	x	1	0	0	0	0	0	0	1	0	1
x	x	x	1	0	0	0	0	0	1	1	1
x	x	x	x	1	0	0	0	1	0	0	1
x	x	x	x	x	1	0	0	1	0	1	1
x	x	x	x	x	x	1	0	1	1	0	1
x	x	x	x	x	x	x	1	1	1	1	1

하나 이상의 입력이 1 일 경우 제일 오른쪽의 1에 대한 값을 출력하기 때문에 $y_0 \sim y_6$은 0 이든 1 이든 상관없어 무관항처리함

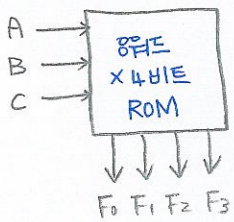
※ d는 모든입력이 0 일경우 0 아니면 1 값을 가짐

9.5 읽기전용 메모리 (ROM)

ROM = Read-only-Memories 이란?

저장된 데이터를 바꿀 수 없고 데이터를 저장해 그 데이터를 읽어내는 읽기기능만 가능한 회로

8워드 × 4비트 ROM



A	B	C	F_0	F_1	F_2	F_3
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	1	1
0	1	1	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	1	0	1	1	1	1
1	1	1	0	1	0	1

ROM에 저장되어 있는 출력들을 워드 (word) 라고 함

각 워드는 4비트 길이를 가져

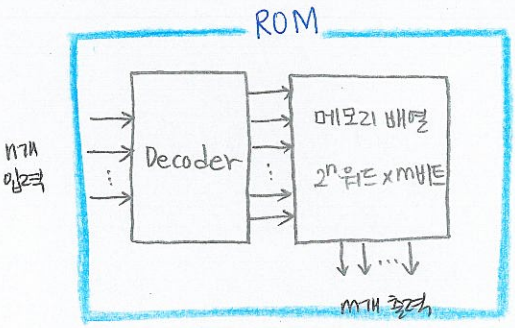
"8개 워드 × 4 비트 ROM" 이라고 함

n개의 입력들은 2^n개의 워드 중 하나를 선택하기 위한 주소 역할을 함

※ F_0만 끄집어내 사용하면 F_0의 진리표를 가지는 논리회로를 실현할 수 있음

tb_elec_engineer@naver.com

ROM 의 내부구조



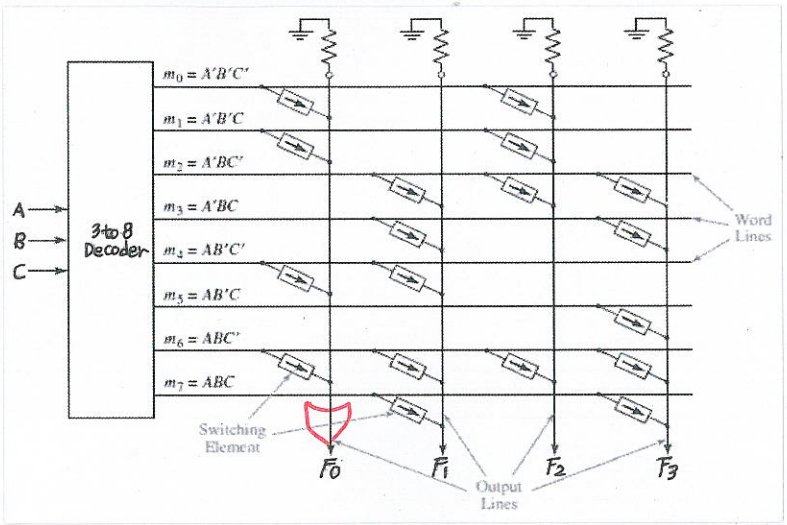
n개의 패턴이 디코더 입력에 인가

→ 2^n 디코더 출력 중 하나가 1 (최소항 생성)

→ 이 디코더 출력선이 메모리 배열의 한 워드 선택

→ 이 워드에 저장된 비트 패턴이 메모리 출력선으로 전송

ROM의 내부회로 구조



P.7의 진리표로 부터 ROM을 표현한 것이다.

진리표 출력이 0 이면 ROM의 메모리 배열에 스위칭 소자가 없고

진리표 출력이 1 이면 스위칭 소자가 연결되 해당 최소항이 출력된다.

디코더를 통해 최소항들이 생성되고 스위칭 소자가 연결되면

출력이 되는데 세로줄은 OR 게이트가 생략되어 있는것이라고 생각하면 된다.

$\therefore F_0 = \sum m\,(0,1,4,6) = A'B' + AC'$

$F_1 = \sum m\,(2,3,4,6,7) = B + AC'$

$F_2 = \sum m\,(0,1,2,6) = A'B' + BC'$

$F_3 = \sum m\,(2,3,5,6,7) = AC + B$

tb_elec_engineer@naver.com

9.6 프로그래머블 논리소자

→ 다양한 논리 함수를 제공하도록 프로그램이 가능한 디지털 집적회로

종류 ① 프로그래머블 로직어레이 (PLA)

② 프로그래머블 어레이로직 (PAL) : 특별한 형태의 PLA

① 프로그래머블 로직어레이 (PLA)

ROM과 동일한 기본기능 수행 : 저장된 데이터를 읽어내는 읽기 기능만 가능한 회로

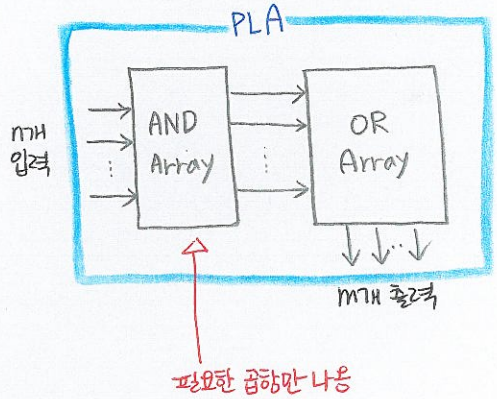
n개 입력과 m개의 출력을 가짐

2^n 워드 × m비트 ROM 이라고 표시

ROM의 내부조직과는 다름 → AND-OR 배열로 이루어져 있는데

입력변수 중 필요한 곱항만 실현시켜 OR 배열로 만듦

PLA의 내부조직



tb_elec_engineer@naver.com

$F_0 = A'B' + AC'$

$F_1 = B + AC'$

$F_2 = A'B' + BC'$

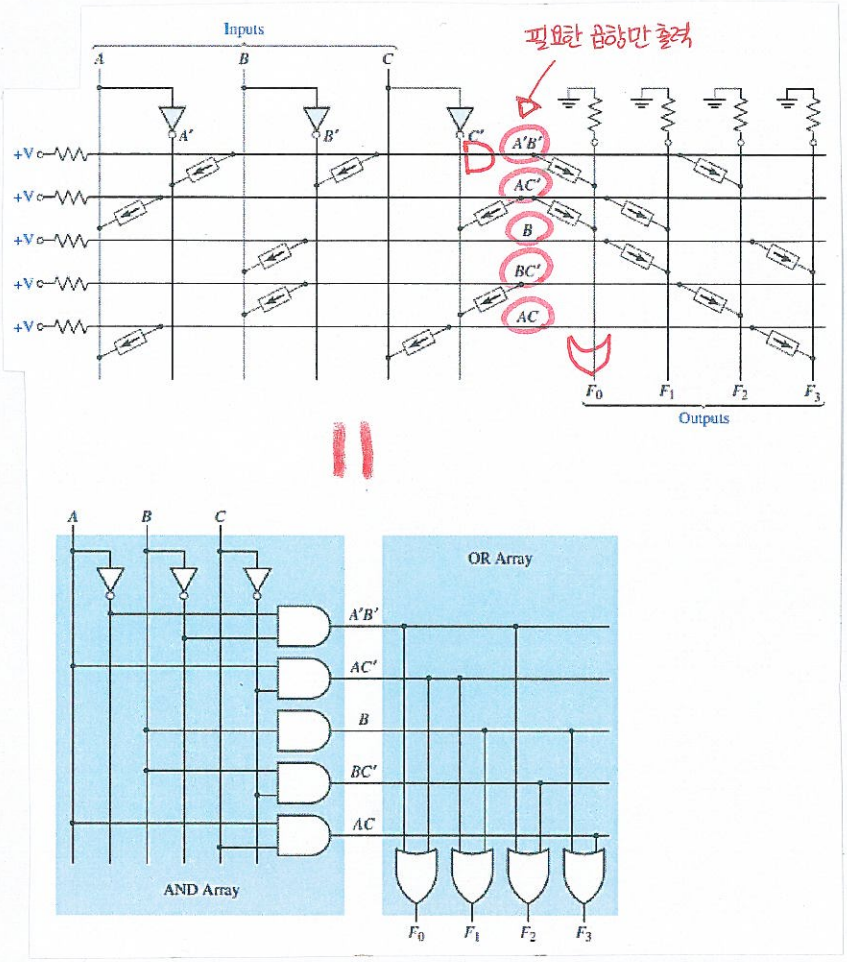
$F_3 = AC + B$

→ 밑줄 친 항들이 필요한 곱항들

밑줄이 없는 항들은 다른 함수에 이미 포함되어 있어 만들 때 뽑아서 사용하면 된다

⇓

PLA 실현



가로는 AND 게이트가

세로는 OR 게이트가 연결되어 있다고 생각!

⇓

PLA 표

필요한 곱항들

Product Term	Inputs A B C	Outputs F_0 F_1 F_2 F_3
A'B'	0 0 –	1 0 1 0
AC'	1 – 0	1 1 0 0
B	– 1 –	0 1 0 1
BC'	– 1 0	0 0 1 0
AC	1 – 1	0 0 0 1

$F_0 = A'B' + AC'$

$F_1 = AC' + B$

$F_2 = A'B' + BC'$

$F_3 = B + AC$

⇓ 변수의 보수: 0

변수: 1

–: 없음

⇓ 해당 곱항이 존재: 1

// 존재 X: 0

tb_elec_engineer@naver.com

예제) $f_1 = a'bd + abd + ab'c' + b'c$ → 필요한 곱항들 묶줄 표시

$f_2 = a'bd + c$

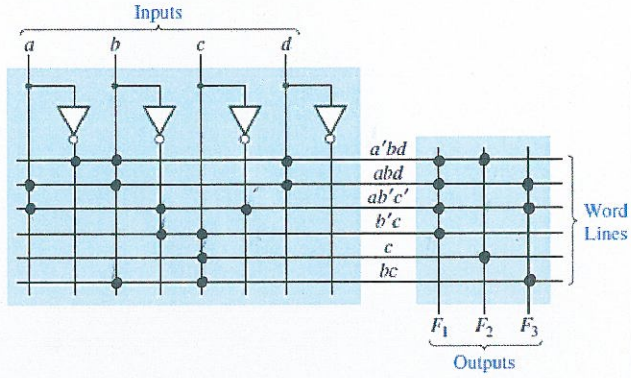
$f_3 = abd + ab'c' + bc$

PLA 표

	a b c d	f_1 f_2 f_3
a'bd	0 1 – 1	1 1 0
abd	1 1 – 1	1 0 1
ab'c'	1 0 0 –	1 0 1
b'c	– 0 1 –	1 0 0
c	– – 1 –	0 1 0
bc	– 1 1 –	0 0 1

(a) PLA table

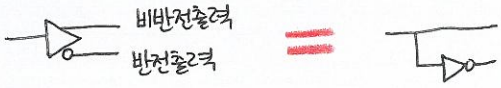
PLA 구조



(b) PLA structure

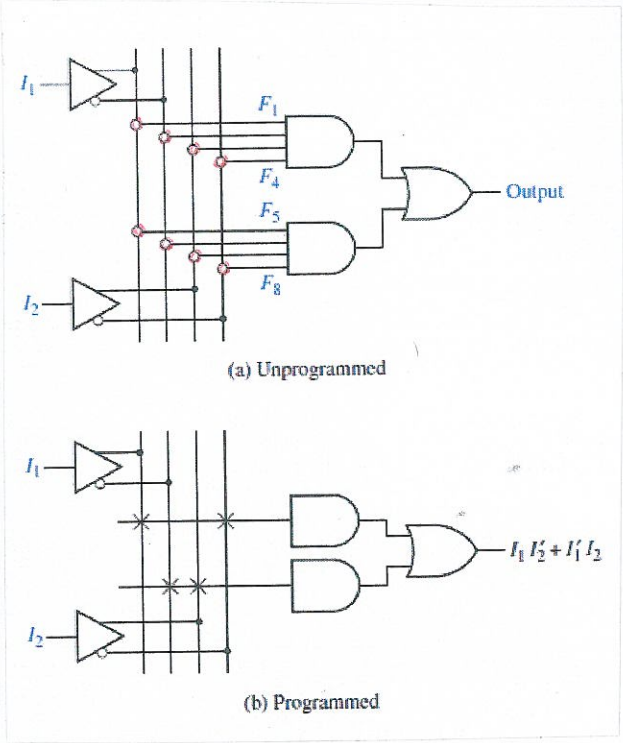
② 프로그래머블 어레이조직 (PAL)

AND 배열은 프로그램이 가능하지만 OR 배열은 고정되어 있는 특별한 형태의 PLA 구조는 PLA 와 같다



AND 배열의 연결은 X 로 표시

tb_elec_engineer@naver.com



○ : 연결할 수 있음을 나타내는 표시
즉, 프로그램을 하지 않은 상태이다

$I_1 I_2' + I_1' I_2$ 함수를 구현하기 위해 PAL 을 사용

예제) PAL 을 이용하여 전덧셈기 설계하기

전덧셈기 논리식

$$Sum = X'Y'C_{in} + X'YC_{in}' + XY'C_{in}' + XYC_{in}$$

$$Cout = XC_{in} + YC_{in} + XY$$