부울대수

부울 함수 간소화

❖ XOR와 XNOR의 카르노 맵

- 입력 변수에 나타나는1 의 개수에 따라 XOR와 XNOR를 구분할 수 있다.
- 카르노 맵에서 그룹으로 묶은 후 제거되는 변수를 제외한 다른 입력 변수의 1의 개수가 홀수이면 XOR, 짝수이면 XNOR이다.

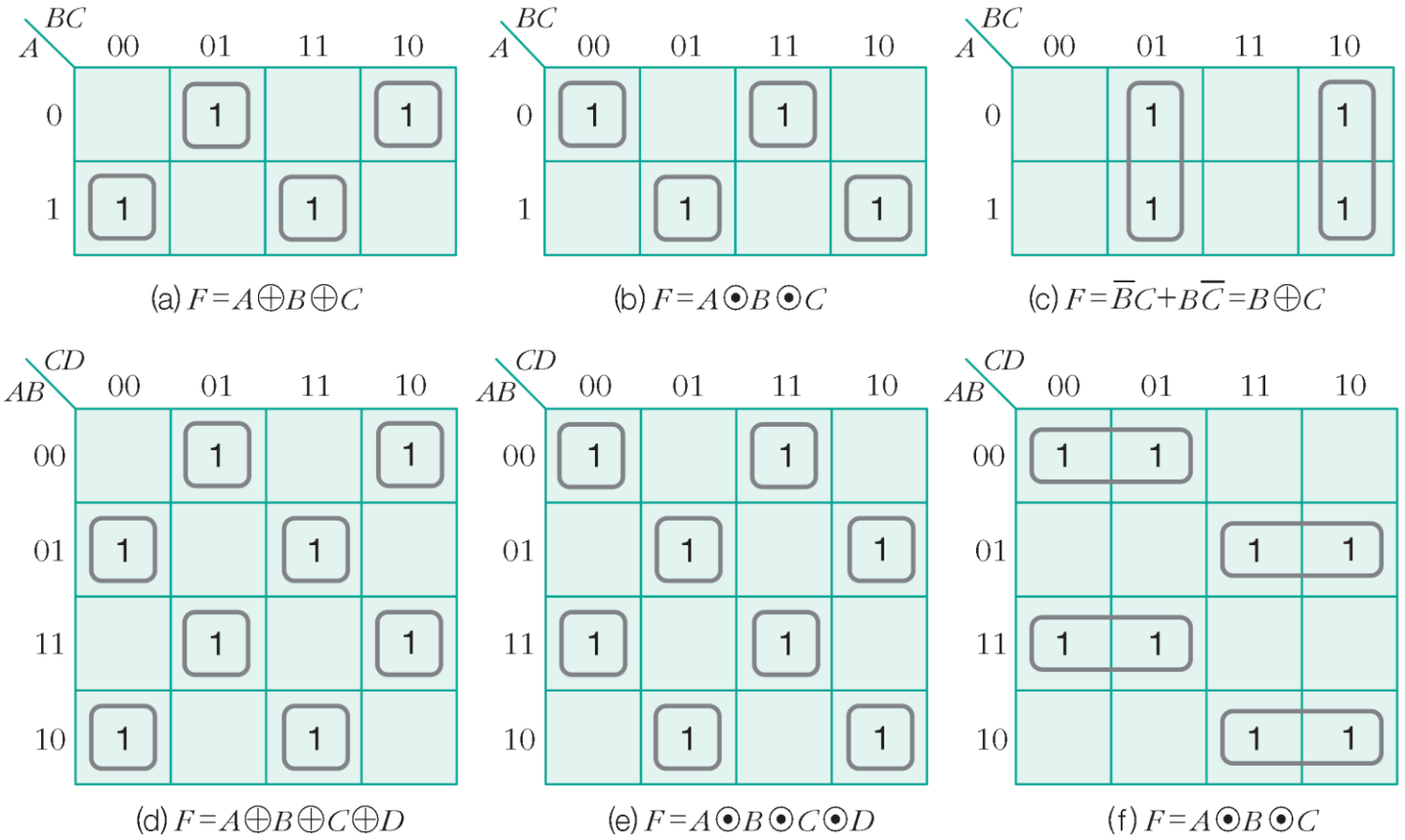


그림 3-31 XOR와 XNOR 카르노 맵 표현

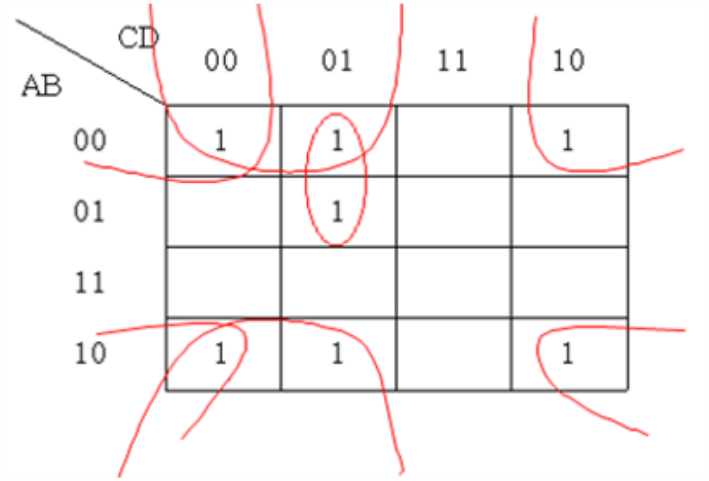
보충자료 : 합의 곱 형식으로 부울 함수의 간소화

[예제] 다음 부울 함수를 (a)논리곱의 논리합 형식으로 간소화하고 (b)논리합의 논리곱 형식으로 간소화하시오.

$$F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$$

풀이)

(a) 논리곱의 논리합 형식으로 간소화하기 위해 카르노 맵의 1인 항을 묶는다.

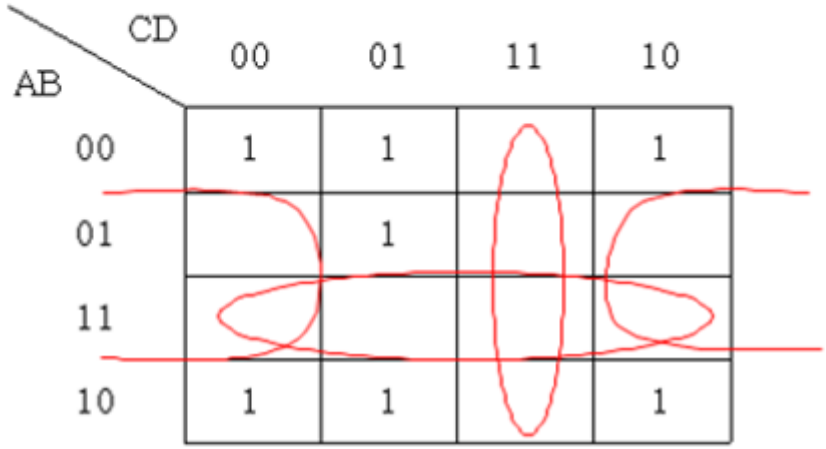


묶은 항의 곱항을 구하고 이를 논리합으로 정리한다.

$F(A,B,C,D) = B'D' + B'C' + A'C'D$ (논리곱의 논리합)

보충자료 : 합의 곱 형식으로 부울 함수의 간소화

(b) 논리합의 논리곱 형식으로 간소화하기 위해 카르노 맵의 0인 항을 묶는다. 이는 F 대신에 F' 을 구한 것이다.



$$F'(A,B,C,D) = AB + CD + BD'$$

$$\begin{aligned} F(A,B,C,D) &= (AB + CD + BD')' \\ &= (AB)' \cdot (CD)' \cdot (BD')' \\ &= (A' + B')(C' + D')(B' + D) \end{aligned}$$

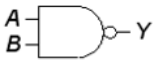
논리합의 논리곱을 얻기위해 0인 항을 논리곱의 논리합으로 묶은 식은 F' 을 구한 것이므로 F 를 구하기 위해 양변에 보수를 취한 결과 위 식을 얻는다.

보충자료 : 곱의 합 형식과 합의 곱 형식 표현의 구현

- **곱의 합 형식 구현 : 1단 AND, 2단 OR로 구현**
  - AND-OR로 구현된 논리회로는 NAND Gate 만으로 구현 가능
- **합의 곱 형식 구현 : 1단 OR, 2단 AND로 구현**
  - OR-AND로 구현된 논리회로는 NOR Gate 만으로 구현 가능

invert-OR(NAND), invert-AND(NOR)

- $Y = \overline{AB} = \overline{A} + \overline{B}$

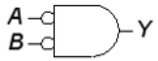


NAND

Invert-OR

- $Y = \overline{A + B} = \overline{A} \cdot \overline{B}$

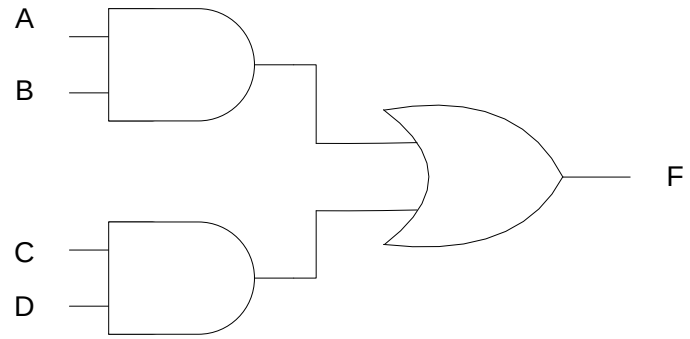
NOR



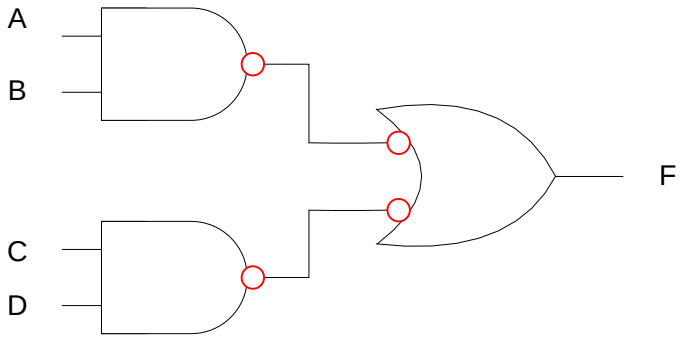
Invert-AND

보충자료 : 곱의 합 논리를 NAND Gate만으로 구현

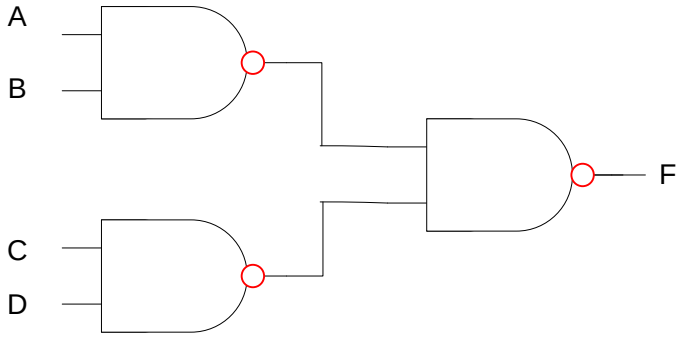
1. F=AB+CD



2. F=((AB)')'+((CD)')'

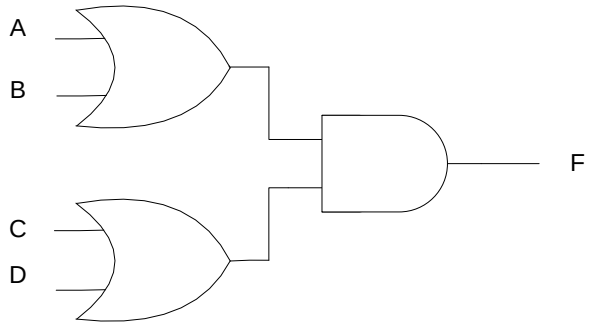


3. F=((AB)' (CD)')'

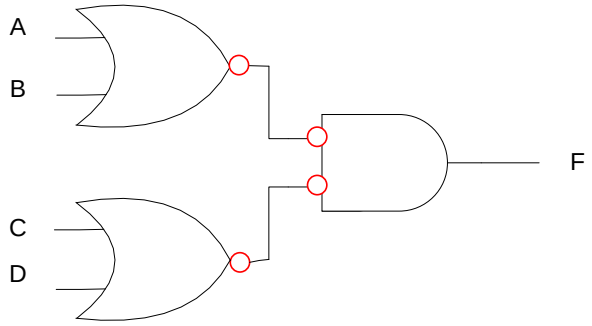


보충자료 : 합의 곱 논리를 NOR Gate만으로 구현

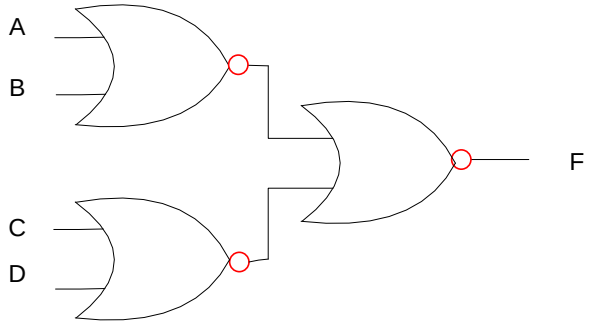
1. F=(A+B)(C+D)



2. F=((A+B)′)′((C+D)′)′



3. F=((A+B)′ +(C+D)′)′



조합논리회로
(Combinational Circuit)

1 데이터 형태에 따른 분류

❑ 조합 논리 회로의 개요

- 조합 논리 회로(combinational logic circuit)는 현재 입력 값으로 출력이 결정되는 회로

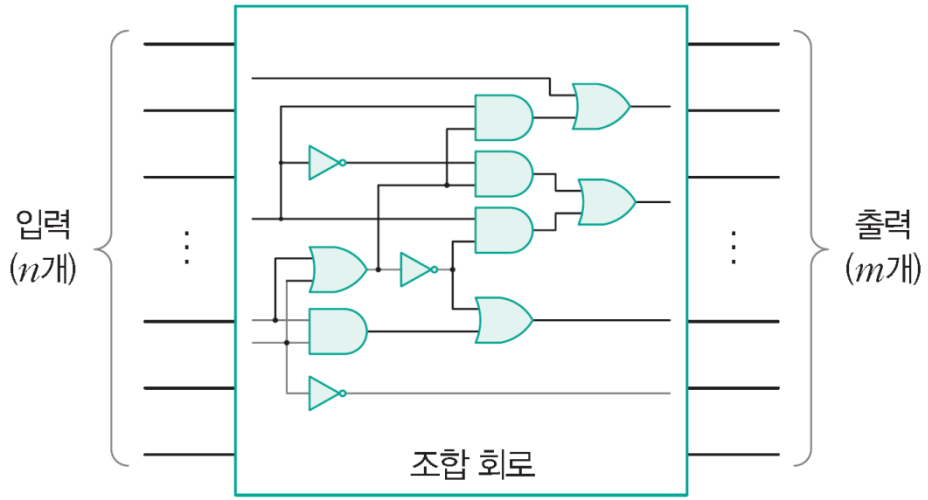


그림 3-32 조합 논리 회로의 개념도

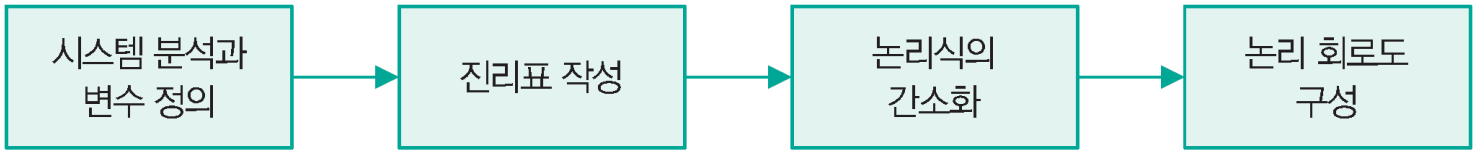


그림 3-33 조합 논리 회로의 설계 과정

2 조합 논리 회로의 종류

❑ 반가산기

- 반가산기(Half-Adder, HA)는 1자리 2진수 2개를 입력하여 합(*S*)과 캐리(Carry, *C*)를 출력하는 조합 논리 회로

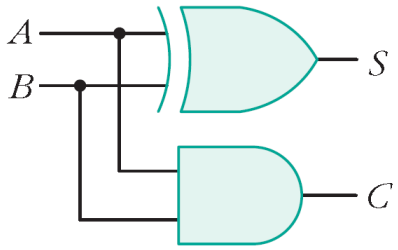
$$\begin{array}{r} A \\ +\ B \\ \hline C\ S \end{array} \qquad \begin{array}{r} 0 \\ +\ 0 \\ \hline 0\ 0 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 0\ 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 0\ 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 1\ 0 \end{array}$$

입력		출력	
A	*B*	*S*	*C*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

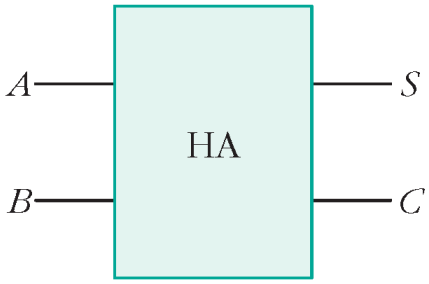
$S = \overline{A}B + A\overline{B} = A \oplus B$

$C = AB$

(a) 진리표와 논리식



(b) 논리 회로



(c) 논리 기호

그림 3-34 반가산기

❑ 전가산기

- 전가산기(Full-Adder, FA)는 2진수 입력 A, B와 아랫자리에서 올라온 캐리 C_i를 포함하여 1자리 2진수 3개를 더하는 조합 논리 회로

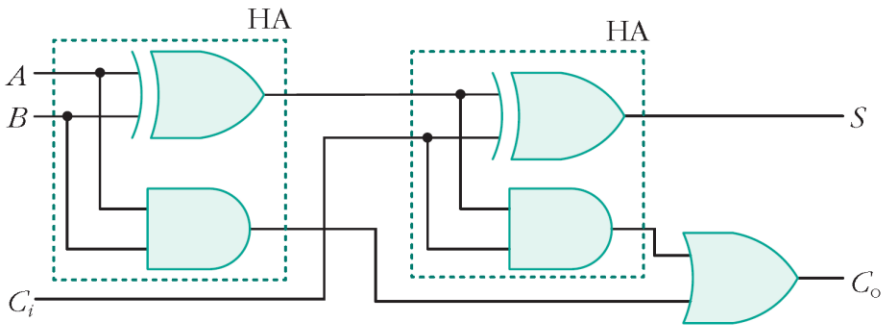
입력			출력	
A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a) 진리표

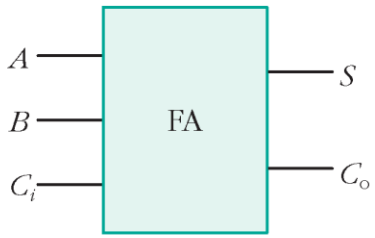
$$
\begin{aligned}
S &= \overline{A}\,\overline{B}C_i + \overline{A}B\overline{C_i} + A\overline{B}\,\overline{C_i} + ABC_i \\
&= \overline{A}(\overline{B}C_i + B\overline{C_i}) + A(\overline{B}\,\overline{C_i} + BC_i) \\
&= \overline{A}(B \oplus C_i) + A(\overline{B \oplus C_i}) \\
&= A \oplus (B \oplus C_i) = (A \oplus B) \oplus C_i
\end{aligned}
$$

$$
\begin{aligned}
C_o &= \overline{A}BC_i + A\overline{B}C_i + AB\overline{C_i} + ABC_i \\
&= C_i(\overline{A}B + A\overline{B}) + AB(\overline{C_i} + C_i) \\
&= C_i(A \oplus B) + AB
\end{aligned}
$$

(b) 논리식의 간소화



(c) 논리 회로



(d) 논리 기호

그림 3-35 전가산기

❑ 반감산기

- 반감산기(Half-Subtractor, HS)는 1비트 2진수 A에서 B를 빼 차(D)와 빌림 수(K)를 계산하는 뺄셈회로

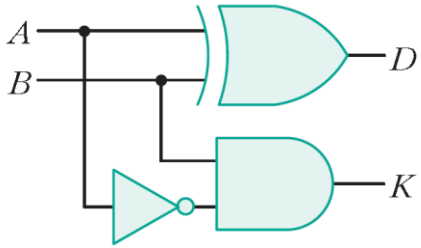
입력 A	입력 B	출력 D	출력 K	설명($D=A-B$) 뺄 수 없으면 윗자리에서 빌려 와 계산한다.		
0	0	0	0	0−0=0	빌림 수 없음 ($K=0$)	$\therefore D=0$
0	1	1	1	0−1=−1	빌림 수 2 ($K=1$)	$\therefore D=2-1=1$
1	0	1	0	1−0=1	빌림 수 없음 ($K=0$)	$\therefore D=1$
1	1	0	0	1−1=0	빌림 수 없음 ($K=0$)	$\therefore D=0$

(a) 진리표와 원리

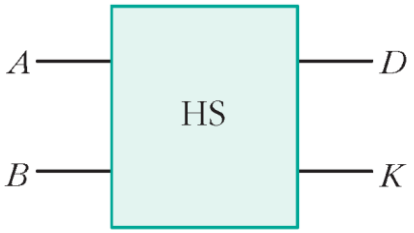
$$D = \overline{A}B + A\overline{B} = A \oplus B$$

$$K = \overline{A}B$$

(b) 논리식



(c) 논리 회로



(d) 논리 기호

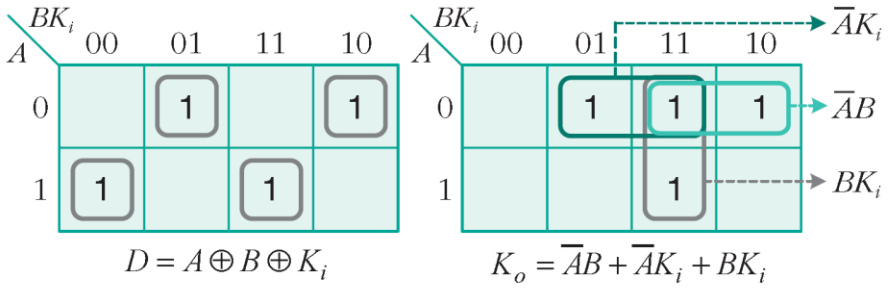
그림 3-36 반감산기

❑ 전감산기

- 전감산기(Full-Subtractor, FS)는 2진수 입력 A, B와 아랫자리로 빌려주는 수 K_i를 포함하여 A-B-K_i를 계산하는 조합 논리 회로

입력 A	B	K_i	출력 D	K_o	설명($D=A-B-K_i$) 뺄 수 없으면 윗자리에서 빌려 와 계산한다.		
0	0	0	0	0	$0-0-0=0$,	빌림 수 없음($K_o=0$)	$\therefore D=0$
0	0	1	1	1	$0-0-1=-1$,	빌림 수 2($K_o=1$)	$\therefore D=2-1=1$
0	1	0	1	1	$0-1-0=-1$,	빌림 수 2($K_o=1$)	$\therefore D=2-1=1$
0	1	1	0	1	$0-1-1=-2$,	빌림 수 2($K_o=1$)	$\therefore D=2-2=0$
1	0	0	1	0	$1-0-0=1$,	빌림 수 없음($K_o=0$)	$\therefore D=1$
1	0	1	0	0	$1-0-1=0$,	빌림 수 없음($K_o=0$)	$\therefore D=0$
1	1	0	0	0	$1-1-0=0$,	빌림 수 없음($K_o=0$)	$\therefore D=0$
1	1	1	1	1	$1-1-1=-1$,	빌림 수 2($K_o=1$)	$\therefore D=2-1=1$

(a) 진리표와 원리

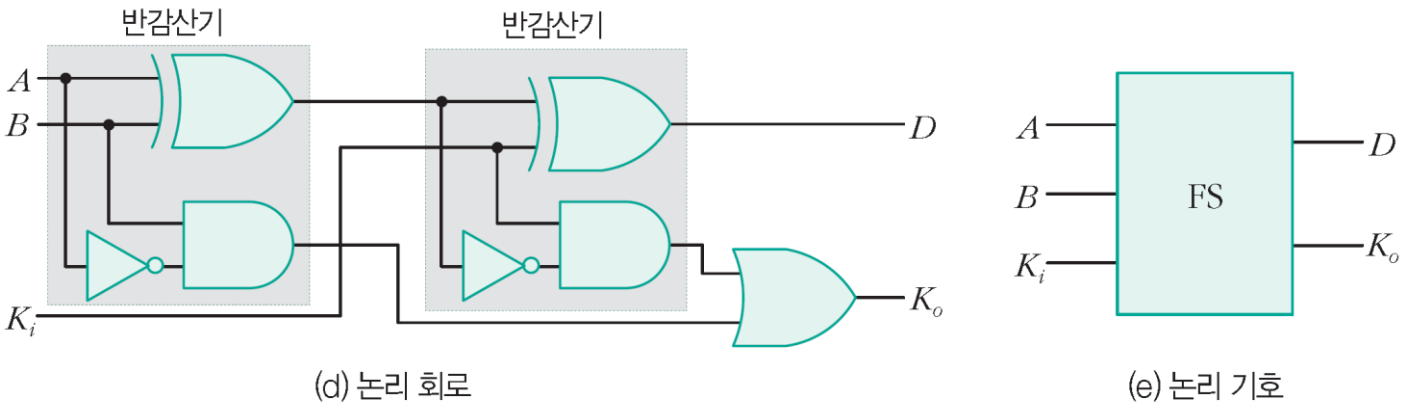


(b) 논리식의 유도

$$K_o = \overline{A}\overline{B}K_i + \overline{A}BK_i + A\overline{B}\overline{K}_i + ABK_i$$
$$= (\overline{A}\overline{B} + AB)K_i + \overline{A}B(K_i + \overline{K}_i)$$
$$= (\overline{A \oplus B})K_i + \overline{A}B$$

(c) 논리식의 간소화

❑ 전감산기(계속)



(d) 논리 회로 (e) 논리 기호

그림 3-37 전감산기

❑ 병렬 가감산기

- **병렬 가산기**(parallel-adder) : 전가산기 여러 개를 병렬로 연결하여 만든 2비트 이상의 가산기

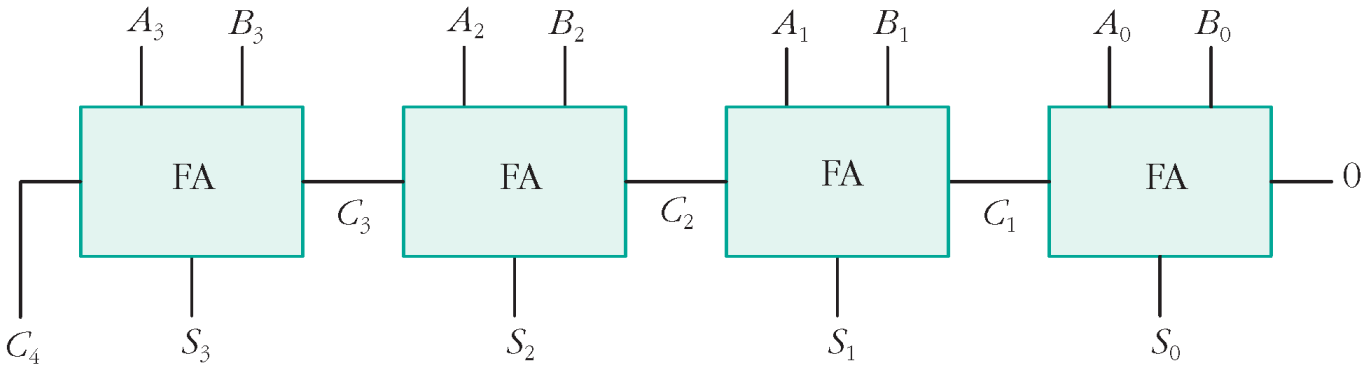


그림 3-38 전가산기를 이용한 병렬 가산기

캐리 예측 가산기(carry lookahead adder)

병렬 가산기는 속도가 매우 느리다. 그 원인은 아랫 단에서 윗단으로 전달되는 캐리 때문이다. 비트가 늘어날수록 지연이 더욱 심해진다. 이를 해결하기 위해 **캐리 예측 가산기**를 사용한다.

- **병렬 가감산기**(parallel-adder/subtracter) : 병렬 가산기의 B입력을 부호 S(sign)와 XOR하여 전가산기의 입력으로 사용함으로써 덧셈과 뺄셈이 모두 가능한 회로

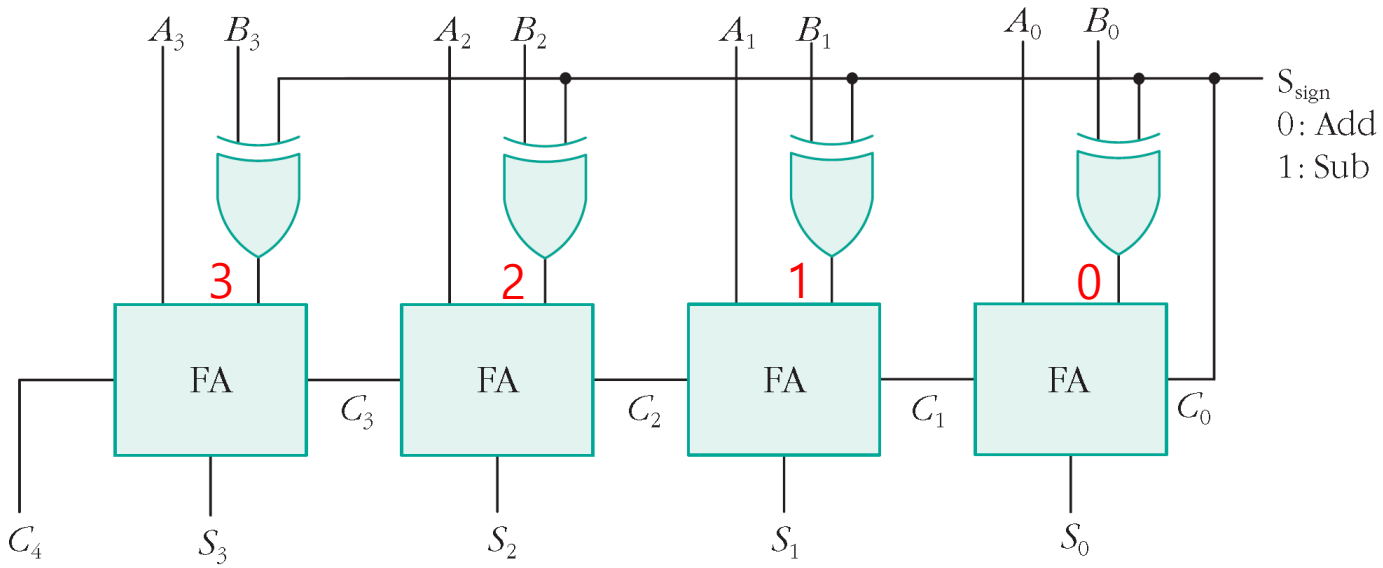


그림 3-39 병렬 가감산기

$S_{sign}=0$일 때 $S_0=A_0+B_0$, $S=A+B$

$S_{sign}=1$일 때 $S_0=A_0+\overline{B_0}$, $S=A+\overline{B}+1$

XOR Gate

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

A=0일 때 출력 F는 B와 같다.

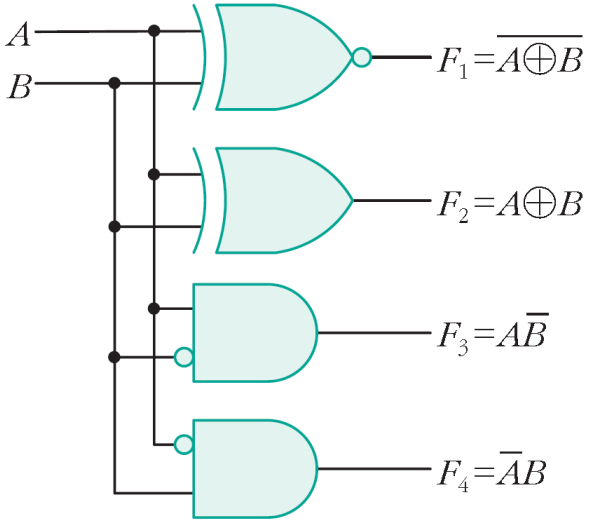
A=1일 때 출력 F는 $\overline{B}$와 같다.

❑ 비교기

- **2진 비교기**(comparator)는 두 2진수 값의 크기를 비교하는 회로다.

입력		출력			
A	B	$A=B$ F_1	$A\neq B$ F_2	$A>B$ F_3	$A<B$ F_4
0	0	1	0	0	0
0	1	0	1	0	1
1	0	0	1	1	0
1	1	1	0	0	0

(a) 진리표



(b) 논리 회로

그림 3-40 1비트 비교기

수고하셨습니다!