4-2.

BCD Adder (10진수 Adder)

binary adder의 결과를 BCD 코드로 변환.

Binary Sum						BCD Sum					
$K(C_{out})$	Z_8	Z_4	Z_2	Z_1		C_{out}	S_8	S_4	S_2	S_1	10진수
0	0	0	0	0		0	0	0	0	0	0
0	0	0	0	1		0	0	0	0	1	1
0	0	0	1	0		0	0	0	1	0	2
0	0	0	1	1		0	0	0	1	1	3
0	0	1	0	0		0	0	1	0	0	4
0	0	1	0	1		0	0	1	0	1	5
0	0	1	1	0		0	0	1	1	0	6
0	0	1	1	1		0	0	1	1	1	7
0	1	0	0	0		0	1	0	0	0	8
0	1	0	0	1		0	1	0	0	1	9
0	1	0	1	0	m_{10}	1	0	0	0	0	10
0	1	0	1	1	m_{11}	1	0	0	0	1	11
0	1	1	0	0	m_{12}	1	0	0	1	0	12
0	1	1	0	1	m_{13}	1	0	0	1	1	13
0	1	1	1	0	m_{14}	1	0	1	0	0	14
0	1	1	1	1	m_{15}	1	0	1	0	1	15
1	0	0	0	0		1	0	1	1	0	16
1	0	0	0	1		1	0	1	1	1	17
1	0	0	1	0		1	1	0	0	0	18
1	0	0	1	1		1	1	0	0	1	19

$+(0110)_2$

$z_8z_4 \backslash z_2z_1$	00	01	11	10
00				
01				
11	m_{12}	m_{13}	m_{15}	m_{14}
10			m_{11}	m_{10}

$= z_8z_4 + z_8z_2$

BCD Sum의 Carry Out $= K + z_8z_4 + z_8z_2$

↓

※ BCD Sum 에서 Carry Out이 있다면,
$S_8S_4S_2S_1$에 $+ (0110)_2$ 를 해서 변환하기.

Binary Multiplier

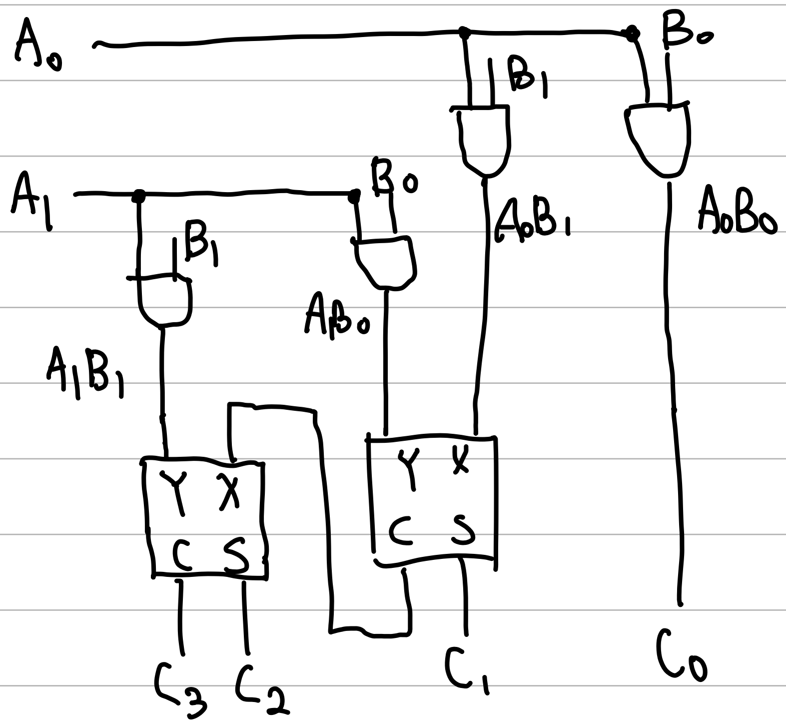
$$
\begin{array}{rrrrrrr}
 & & & 1 & 1 & 0 & 1 \\
\times & & & 0 & 1 & 1 & 0 \\
\hline
 & & & 0 & 0 & 0 & 0 \\
 & & 1 & 1 & 0 & 1 & \\
 & 1 & 1 & 0 & 1 & & \\
0 & 0 & 0 & 0 & & & \\
\hline
1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
$$

Adder 3개 필요.

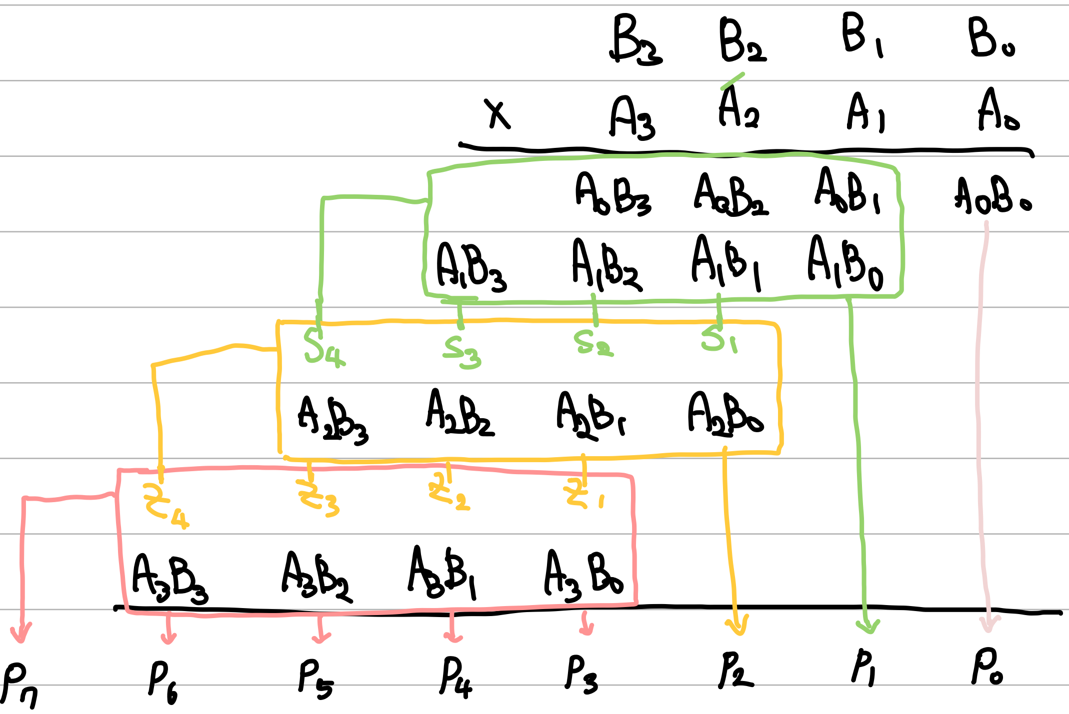
← N×N bit 하면 결과는 2N bit 필요.

2x2 binary multiplier

$$
\begin{array}{ccccc}
 & & & B_1 & B_0 \\
 & & \times & A_1 & A_0 \\
\hline
 & & C_{out} & A_0B_1 & A_0B_0 \\
 & & A_1B_1 & A_1B_0 & \\
\hline
 & C_{out} & C_2 & C_1 & C_0
\end{array}
$$



4×4 binary multiplier



Special Case

2의 제곱수를 곱하거나 나눌 때,

$(11)_2 \times (10)_2 = (110)_2$

$(11)_2 \times (100)_2 = (1100)_2$

$(110)_2 \div (10)_2 = (11)_2$

Comparator

$A = A_3A_2A_1A_0$
$B = B_3B_2B_1B_0$ $X_i = A_iB_i + A_i'B_i'$ 즉, XNOR

$A=B$ 라면, $X_3X_2X_2X_1 = 1$

$A>B$ 라면, $A_3B_3' + X_3A_2B_2' + X_3X_2A_1B_1' + X_3X_2X_1A_0B_0' = 1$

$A<B$ 라면, $A_3'B_3 + X_3A_2'B_2 + X_3X_2A_1'B_1 + X_3X_2X_1A_0'B_0 = 1$

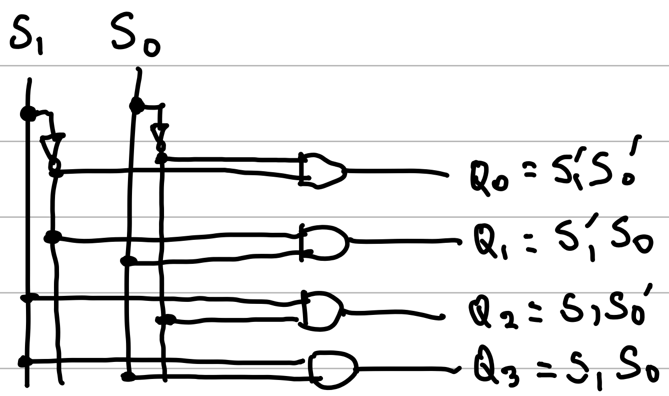
Decoder ☆

k-bit 입력을 받아 특정한 하나의 입력일 때만 1 출력.
↓
즉, 2^k 개의 입력 조합 중 하나일 때만 1출력.

n to 2^n decoder 는 2^n 개의 입력 조합 중 하나만 uniquely true.

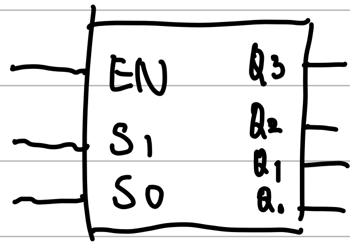
2 to 2^2 decoder

S_1	S_0	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



EN (enable)이 1이면 decoder 동작, 0이면 0만 출력.

EN	S_1	S_0	Q_0	Q_1	Q_2	Q_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



Decoder는 결국 minterm generator 이다.

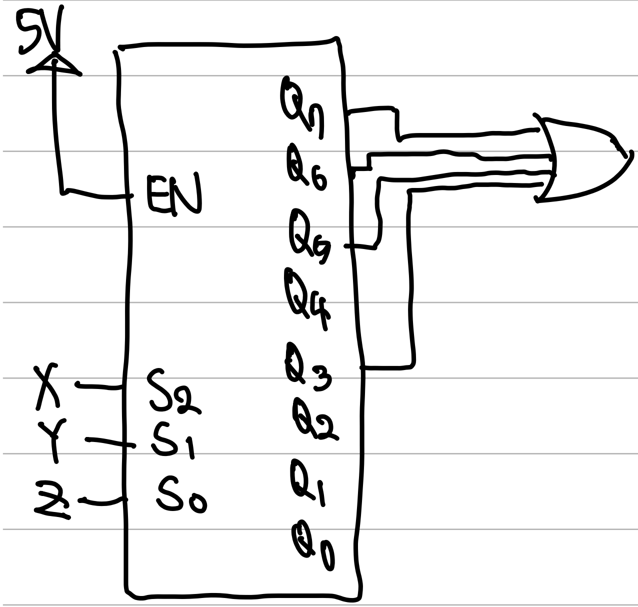
Adder with Decoder

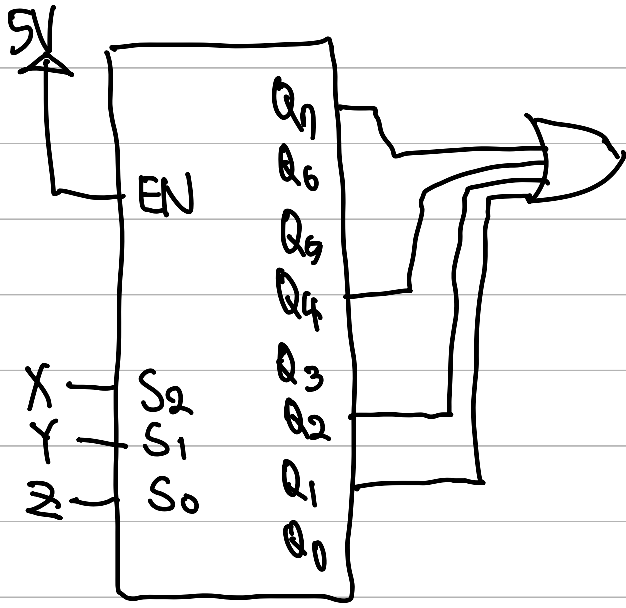
X	Y	Z	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$Carry = \sum_m (3, 5, 6, 7)$

$Sum = \sum_m (1, 2, 4, 7)$

$Carry = \sum_m (3, 5, 6, 7)$ $Sum = \sum_m (1, 2, 4, 7)$





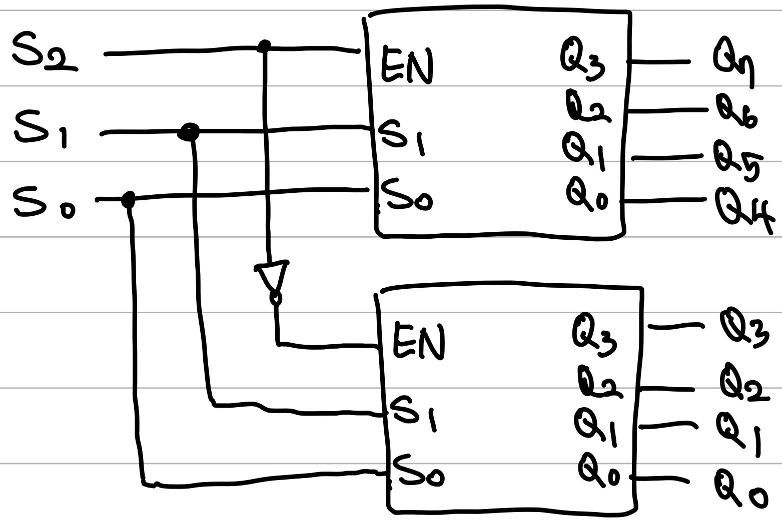
3 to 8 decoder

2개의 2 to 4 decoder로 구현 가능!

S_2	S_1	S_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_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

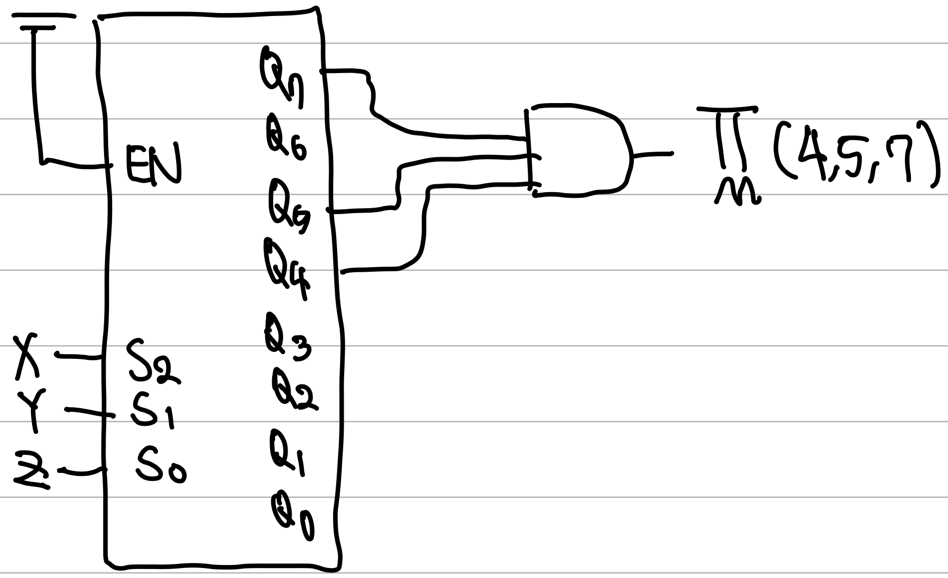
S_2가 0일때 EN=0 취급

S_2가 1일때 EN=0 취급



Active-Low Decoder

Ground 0



Encoder

(Decoder의 역)

2^n개의 input 중에 하나만 1일때 n개의 bit으로 출력.

↓

하나 이상이 1이면? 모든 input이 0이면?

하나 이상이 1이면? → input에 priority를 준다.

모든 input이 0이면? → Vaild 값이 0이면 무의미한 값, 1이면 유의미한 값

Priority Encoder

D_0	D_1	D_2	D_3	X	Y	Vaild = $D_0+D_1+D_2+D_3$
0	0	0	0	x	x	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

$X = D_3 + D_2$

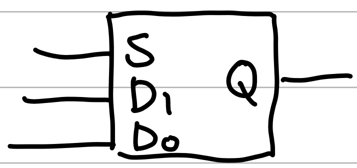
D_0D_1 \ D_2D_3	00	01	11	10
00	X	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

$Y = D_3 + D_1D_2'$

D_0D_1 \ D_2D_3	00	01	11	10
00	X	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

Multiplexer ☆

2^n to 1 Multiplexer

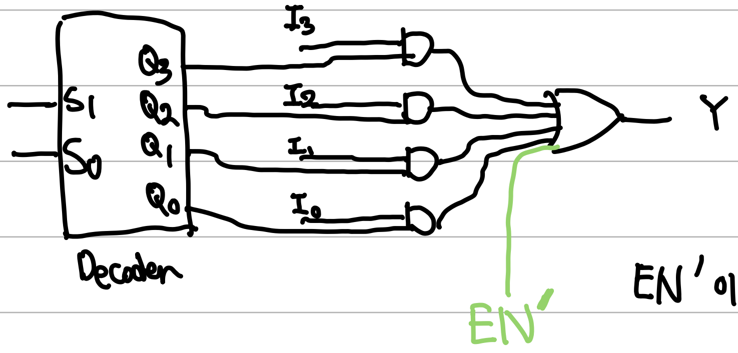


$Q = S'D_0 + SD_1$

Selection이 0이면 $Q = D_0$

Selection이 1이면 $Q = D_1$

4 to 1 Multiplexer



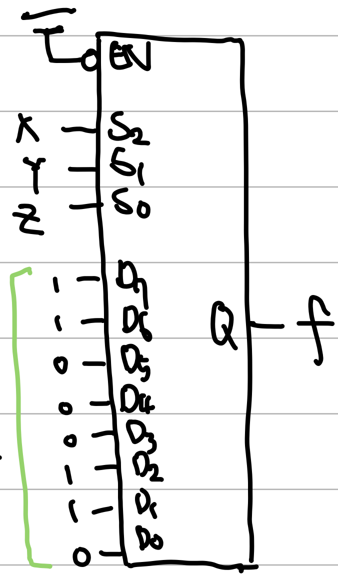
EN'이 0 (즉, 활성화 됨)면 Multiplexer 동작
1 (비활성화) 면 Y는 항상 1

Multiplexer 활용

$f = \sum_m (1,2,6,7)$

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	0
1	1	0	1
1	1	1	1

원하는 결과대로 설정



더 간단한 방법

$f = \sum_m (1,2,6,7)$

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	0
1	1	0	1
1	1	1	1

$XY = 00,\ f = Z$

$XY = 01,\ f = Z'$

$XY = 10,\ f = 0$

$XY = 11,\ f = 1$