

Xilinx Zynq FPGA, TI DSP, MCU
기반의 프로그래밍 및 회로 설계
전문가 과정

#75

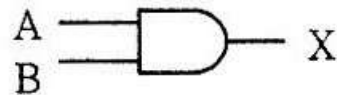
학생 : 김시윤
강사 : Innova Lee(이 상훈)

논리회로 카르노맵 ,진리표 , 여기표 개념정리

카르노맵을 이용하면 입력과 출력을 통해 논리식을 쉽게 도출할 수 있다.

먼저 쉽게 조합논리회로부터 카르노맵을 이용해보도록 한다.

입력신호		출력신호
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



COVER

우선 카르노맵은 입력에 대한 출력을 표로 표시해주는 것 이라고 설명할 수 있다. 여기서 입력은 A , B 이며 출력은 X 이다.

우리가 구하는 것은 A ,B 가 들어왔을때의 X 의 값이기 때문에 $X = ??$ 을 알고 싶은 것이다. 그러면 카르노맵을 X에 관한 카르노맵을 그려야한다.

간단히 표를 통해 보여주도록 한다.

X에 대한 카르노맵

A \ B	0	1
0		
1		1

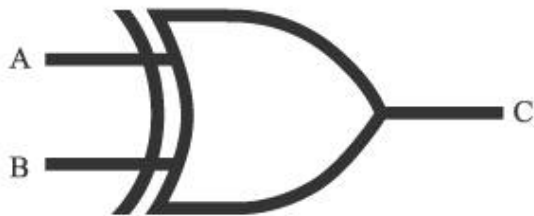
A,B 의 결과가 1인 경우만 카르노맵에 표시한다.

여기서 $A = 1$ 일때와 $B = 1$ 일때가 $X = 1$ 이므로 오른쪽 하단에 1이 표시된다.

카르노맵은 2^n 개를 묶어서 간략화 시킨다 여기서는 오른쪽 하단 한가지 밖에 없기 때문에 $X = A \& B$ 가 된다.

아직 이해가 안되는 사람이 있을거 같아 XOR로 한번 더 해보도록 한다.

XOR 게이트



표준 논리 기호

©doopedia.co.kr

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

진리표

Xor의 진리표는 위와 같다. 아까와 같이 진리표를 보고 카르노맵을 그려보도록 한다. 이번에는 C를 알고싶기 때문에 C에 대한 카르노맵을 그려보면 다음과 같다.

c에대한 카르노맵

A \ B	0	1
0		1
1	1	

C가 1인 순간은 A가 0이고 B가 1일 때, A가 1이고 B가 0인 순간이다.

카르노맵은 서로 근접한(1비트차이) 1을 2^n 개로 묶는다.
여기서 묶어보면 빨간네모와 같고 그것을 그대로 써보면

$A = 1$ 일때라 가정 $B = 1$ 일때라 가정.

윗줄에 네모는 $A = 0, B = 1$ 일 때 이므로 $\bar{A} \times B$ 가 되고,
밑에 줄의 네모는 $A = 1, B = 0$ 일 때 이므로 $A \times \bar{B}$ 가 된다.

카르노맵에 네모가 두 개 있으므로 위에 네모 상태 또는 밑에 네모 상태
가 C의 결과 값이 되므로 그것을 논리식으로 표현하면

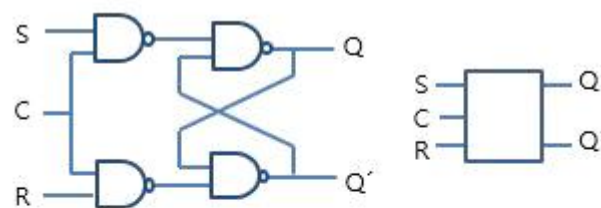
$C = A \times \bar{B} + \bar{A} \times B$ 가 된다.

이제 어느 정도 카르노맵에 대해 알 수 있을 것이다.

그럼 이제 카르노맵에 익숙해지기 위해 순차논리회로에 대한 카르노맵을
진행해보도록 한다.

순차논리회로 카르노맵

우선 간단한게 Gated SR Latch에 대한 카르노맵을 해보도록 한다.



C	S	R	Q(T+1)
0	X	X	No Change
1	0	0	No Change
1	0	1	0
1	1	0	1
1	1	1	부정 (undefind)

gated sr latch 의 진리표를 보면 위와 같다.

여기서 주의할 점은 Gated SR Latch 는 $Q(t+1)$ 즉 미래의 값을 알고싶기 때문에 현재의 $Q(t)$ 값과 SR 값에따른 $Q(t+1)$ 을 표현한 진리표가 필요하다.

위 진리표를 토대로 그려보면 아래와 같은 진리표가 재탄생한다.

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

위 진리표를 토대로 카르노맵을 그려보면 다음과 같다.

Q \ SR	00	01	11	10
0			X	1
1	1		X	1

여기서 SR 의 비트가 진리표에서는 00 , 01 , 10 , 11 인데 카르노맵에서는 00 , 01 , 11 , 10 이다. 이 이유는 카르노맵은 서로 인접한 비트들을 표로 나타낸 것이기 때문이다. 01 과 10 은 첫 번째 비트와 두 번째 비트가 서로 달라 2비트가 반전되어야한다, 즉 2비트 차이내기 때문에

인접한 비트라고 볼 수 없음으로 01 11 10 으로 1비트 차이나도록 순서를 바꿔준다. (그레이코드 개념을 숙지)

카르노맵은 2^n 개로 묶을수 있으며, 최대한 크게 묶는 것이 좋다.

S, R = 1 일때는 부정 이므로 실제로 대입해보면 0 1 어떤 것이 올지 모르기 때문에 이럴때는 X를 사용한다. X는 카르노맵에서 1로 가정한 다.

크게 묶고 서로 인접한 2^n 개 가 존재하면 작게도 한번 묶으준다.

큰네모와 작은네모 두 개를 논리식으로 표현하면 $Q(t+1)$ 의 논리식이 탄생한다.

큰네모의 식은 도출하면 S 이다.

작은네모의 식을 도출하면 $Q \times \bar{R}$ 이다.

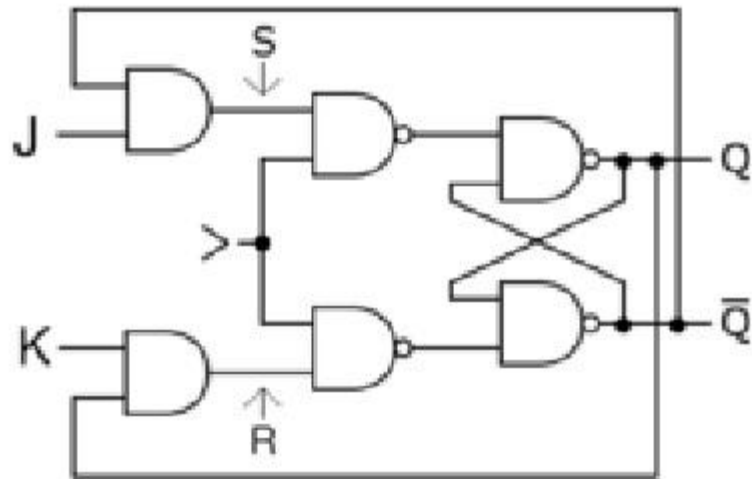
즉 $Q(t+1)$ 은 큰네모 또는 작은네모 이므로 둘이 or 연산을 해주면

$Q(t+1) = S + Q \times \bar{R}$ 와 같이 표현 가능하다.

JK Flip Flop 진료표 여기표 카르노맵

JK Flip Flop 진료표 여기표 카르노맵을 하기위해 여기까지왔다.

우선 JK Flip Flop 의 진리표와 카르노맵을 먼저 진행 한 후 여기표를 설명하도록 하겠다.



JK flip flop operation

J	K	Q(t)	Q (t+1)
0	0		이전 값 유지
0	1	0	1
1	0	1	0
1	1		Toggle(반전)

기존에 알고있던 JK 플립플롭의 진리표를 그리면 다음과 같다.
 하지만 카르노맵을 그려주기 위해서는 Q(t)에 따른 변화를 표에 넣어주
 어야 한다.
 위 진리표를 토대로 다시그리면 다음과 같다.

Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

위를 토대로 카르노맵을 그려보면 다음과 같다.

(Q(t+1) 에 대한 카르노맵)

Q(t) \ JK	00	01	11	10
0			1	1
1	1			1

빨간색 네모와 파란색 네모 두 개가 있다.

빨간색 네모를 도출하면 $\overline{Q} \times J$ 이다.

파란색 네모를 도출하면 $Q \times \overline{K}$ 이다.

결국 Q(t+1)의 최종식은

$Q(t+1) = J \times \overline{Q} + \overline{K} \times Q$ 가 된다.

여기표

여기표를 들어가기 전에 진리표의 개념을 설명하면

어떤 입력이 들어왔을 때의 출력 값을 표로 나타낸 결과라고 할 수 있다.

여기표는 진리표와 반대의 개념이다.

즉 어떤 출력을 얻기 위한 입력 값을 나타낸 표가 여기표 이다.

개념 설명이 끝났으므로 JK 플립플롭의 여기표를 그리며 설명하도록 한다.

Q(t)	Q(t+1)	J	K
0	0	0	0
		0	1
0	1	1	1
		1	0
1	0	0	1
		1	1
1	1	0	0
		1	0

JK 플립플롭의 여기표를 그리면 위처럼 나타낼 수 있다.

현재 출력 Q(t) 와 그 다음 출력 Q(t+1)이 있을 때,

Q(t) = 0 이고 다음출력인 Q(t+1)=0 이 올 경우는 J = 0 , K = 0 일때
와 J = 0 , K = 1 일 때 두가지 경우가 있다. (위 진리표를 참고)

이와 같이 밑에도 여러 가지 경우의 JK입력에 따라 출력값이 정해진다.

여기서 J나 K 가 0 또는 1 두 개의 상태가 와도 되는 경우를 X라고 가정하고 여기표를 간략화 시켜보도록 하면,

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

이와 같이 표현 할 수있다.

이제 여기표를 이용한 예제를 하나 풀도록 한다.

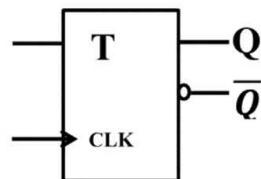
여기표 이용 예제

JK Flip Flop을 이용하여 T Flip Flop을 설계하시오

여기표를 이용하기 딱 좋은 예제이다.

T 플립플롭은 Toggle 플립플롭입니다.

입력이 0일때는 이전값을 유지하고 입력이 1이면 이전값을 반전시킨다.



The truth table of a T-flip-flop

T	Q _{t+1}
0	Q _t
1	$\overline{Q_t}$

T플립플롭의 진리표를 나타내면 위와 같다.

우선 T에 대한 진리표를 Q값을 넣어 써보도록 한다.

Q(t)	T	Q(t+1)
0	0	0 (hold)
0	1	1 (toggle)
1	0	1 (hold)
1	1	0 (toggle)

이와 같다. 이 회로를 JK로 설계하기 위해서는 여기표가 사용이 된다.
JK의 Q(t) , Q(t+1) 여기표 항목을 추가하여 다시 표를 도출하면 아래와 같다.

Q(t)	T	J	K	Q(t+1)
0	0	0	X	0
0	1	1	X	1
1	0	X	0	1
1	1	X	1	0

여기서 우리는 T를 입력했을 때 T의 출력이 J K 에 들어가서 위와 같은 결과가 나오게 만들어 주어야 하기 때문에 J, K를 출력 T Q를 입력으로 두고 카르노맵을 실행하여야한다. 즉 카르노맵은 2개가 나오게된다.

J에 대한 카르노맵

Q \ T	0	1
0		1
1	X	X

파란색 네모는 T

빨간색 네모는 Q이다.

따라서 J에 입력되는 논리식은 다음과 같다.

$$J = T + Q$$

K에 대한 카르노맵

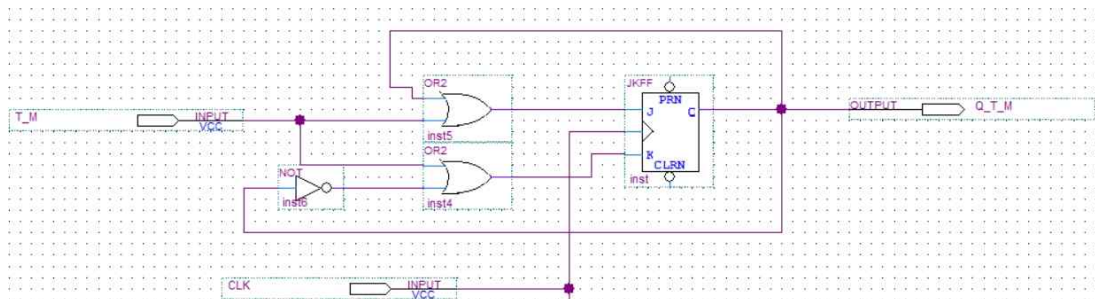
Q \ T	0	1
0	X	X
1		1

빨간색 네모는 \overline{Q}

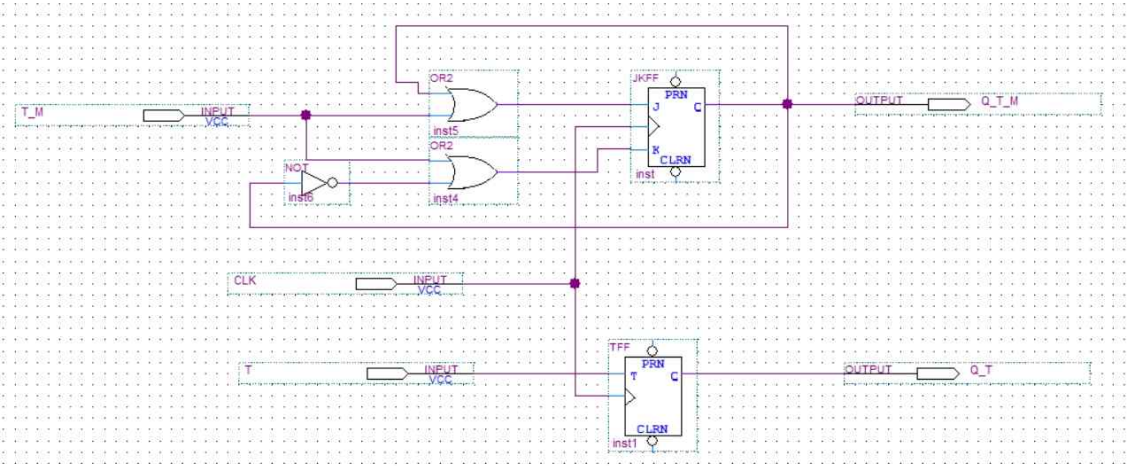
파란색 네모는 T

따라서 $K = \overline{Q} + T$ 가 된다.

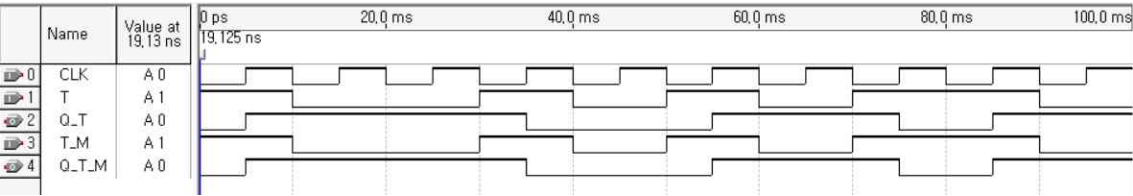
위 토대로 JK를 이용하여 T 플립플롭을 설계하면 다음과 같다.



T는 JK의 Modulation을 줄여 T_M이라 지정하고 $Q = Q_T_M$ 으로 지정하였다. 위 설계한 회로와 T 플립플롭을 비교해보도록한다.



이와같이 설계를 완료하여 출력결과를 관찰해보도록 한다.



관찰 결과 원래의 T의 입력과 원래의 T의 출력인 Q_T랑 비교했을 때 똑같은 결과가 나오는 것을 확인 할 수 있었다.