12주차 예비보고서

전공: 물리학과 학년: 3학년 학번: 20191286 이름: 김나현

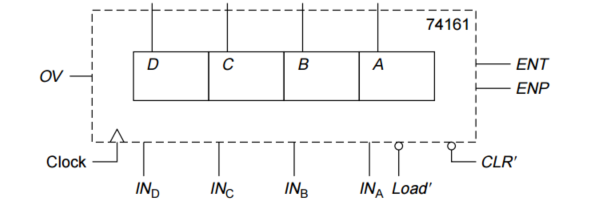
1. Counter에 대해서 조사하시오(예시 포함).

2개 이상의 플립플롭으로 구성되어 매 입력 펄스마다 미리 정해진 순서대로 상태가 변하는 순서논리회로를 의미합니다. 본 실험에서 다룰 카운터는 매 입력 펄스마다 이진수 출력이 나타내는 십진수 값이 1씩 커지거나 작아지도록 하는 논리 회로입니다. 카운터는 발생 횟수를 세거나 동작 순서를 제어하기 위한 논리 회로로, 타이밍 신호 생성에 적합하다는 특징이 있습니다.

카운터는 크게 비동기식 카운터와 동기식 카운터로 구분됩니다. 아래에서 더 자세히 다룰 것이지만 간략하게 설명을 해보면 비동기식 카운터는 ripple carry adder와 비슷하게 카운터를 구성하는 모든 플립플롭이 동일한 클럭 펄스에 의해 활성화되는 것이 아니라 앞쪽에 있는 플립플롭의 출력에 의해 뒤쪽에 있는 플립플롭의 출력이 변화하여 리플 카운터라고도 불립니다. 비동기식 카운터는 ripple carry adder에서도 확인할 수 있었던 전달 지연이 존재하여 글리치(glich)가 발생하여 원치 않은 결과를 얻을 수 있다는 단점이 있기 때문에 이러한 문제점을 해결하기 위해 동기식 카운터가 도입되었습니다. 동기식 카운터는 카운터를 구성하는 모든 플립플롭이 동일한 클럭 펄스에 의해 활성화되므로 비동기식 카운터에서 나타났던 전달 지연의 문제가 발생하지 않습니다.

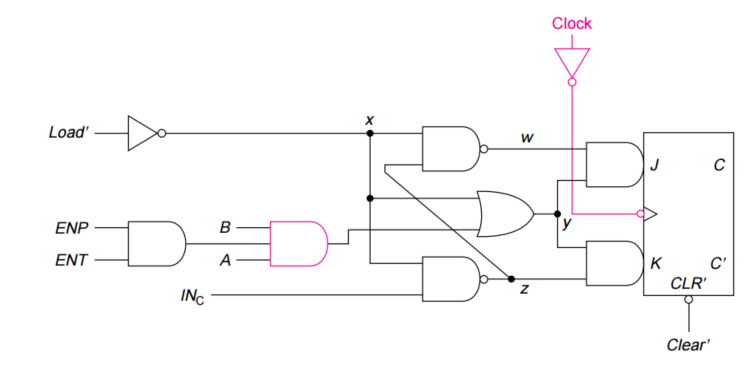
대부분의 동기식 카운터는 parallel load를 가지고 있는데 load 신호와 각 입력 비트 단자를 이용하여 값을 preset합니다. 또 대부분의 카운터는 일반적으로 active-low인 동기식 또는 비동기식 clear 신호를 가지고 있으며 대부분의 비동기식 카운터에서는 static한 clear를 가지고 있습니다. 일부 동기식 카운터는 숫자를 증가시키는 것과 감소시키는 것을 동시에 수행할 수도 있습니다. 대부분의 카운터는 carry와 overflow 출력을 가지고 있는데 이는 카운터가 최대에 도달했는지를 알려주고 다시 0으로 돌아갑니다.

우선, 74161이라는 카운터에 대해 알아볼 것인데 이는 동기식 counting과 loading을 하며 비동기식 active-low clear를 가지고 있는 것입니다. 74161은 두 개의 count enable을 가지고 있는데 하나는 ENT이고 하나는 ENP’입니다. 가장 높은 비트부터 D, C, B, A라는 이름을 붙여 카운터의 모습을 나타내면 다음과 같습니다.



<Figure a> 74161 카운터의 block representation

이 중 다음 그림과 같이 C비트에 대한 논리 회로를 예시로 74161 카운터의 동작을 구체적으로 설명해보겠습니다. Clock 신호가 하강 에지 트리거 플립플롭에 들어가기 전에 NOT 게이트를 거쳐 반전되므로 이 카운터는 상승 에지 트리거 플립플롭으로 동작합니다. C비트 논리 회로와 다른 비트 논리 회로의 유일한 차이점은 붉은 색으로 나타낸 AND 게이트입니다. 이것은 J와 K의 입력을 위한 값입니다. Load’가 0이면 x 지점은 1이 되고 y 지점 역시 1이 되어 z가 INc’과 같고 w가 INc와 같으므로 플립플롭은 Inc를 로드하게 됩니다. 반대로 Load’가 1이면 x 지점은 0이 되고, w와 z는 모두 1이 되고 y는 붉은 색으로 나타낸 AND 게이트의 출력과 같게 되므로 붉은 색 AND 게이트의 출력이 1이면 J와 K 모두 1이 되어 토글될 수 있습니다.



<Figure b> 74161 카운터에서 C 비트에 대한 논리 회로

1. 동기식 Counter 및 비동기식 Counter에 대해서 조사하시오.

비동기식 카운터는 하나의 클록 펄스에 모든 플립플롭이 동기화되지 않고 앞에 있는 플립플롭의 출력이 뒤에 있는 플립플롭의 클럭으로 사용되어 동작하는 카운터를 의미합니다. 비동기식 카운터는 리플(ripple) 카운터라고도 불리며 각 플립플롭을 통과할 때마다 지연시간이 누적되므로 전달 지연(propagation delay)이 발생한다는 단점이 있지만 동작 및 논리 회로 구성이 단순하고, 구현이 용이하다는 장점이 있습니다. 비동기식 카운터는 주로 JK 플립플롭이나 T 플립플롭을 일렬로 연결한 형태로 구현됩니다. 비동기식 카운터는 상향 카운터와 하향 카운터로 구분되는데 상향 카운터는 하강 에지에서 트리거되어 하강 에지가 발생될 때마다 각 플립플롭에서의 출력이 0000부터 1111까지 십진수로 1씩 커지는 결과를 얻을 수 있습니다. 반대로, 하향 카운터는 상승 에지에서 트리거되어 상승 에지가 발생될 때마다 각 플립플롭에서의 출력이 1111부터 0000까지 십진수로 1씩 줄어드는 결과를 얻을 수 있습니다. 다음은 JK 플립플롭으로 구현한 상향, 하향 카운터와 timing diagram입니다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

<Figure 1> JK 플립플롭으로 구현한 상향 카운터

텍스트, 전자기기, 키보드이(가) 표시된 사진

자동 생성된 설명

<Figure 2> JK 플립플롭으로 구현한 상향 카운터의 timing diagram

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

<Figure 3> JK 플립플롭으로 구현한 하향 카운터

텍스트, 전자기기, 키보드이(가) 표시된 사진

자동 생성된 설명

<Figure 4> JK 플립플롭으로 구현한 하향 카운터의 timing diagram

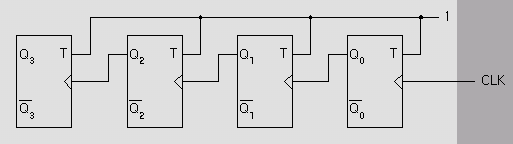
<Figure 2>와 <Figure 4>를 보면 첫 번째 플립플롭은 클럭 펄스에 의해 trigger되지만 두 번째 플립플롭은 첫 번째 플립플롭의 출력에 의해, 세 번째 플립플롭은 두 번째 플립플롭의 출력에 의해, 네 번째 플립플롭은 세 번째 플립플롭에 의해 trigger된다는 것을 확인할 수 있습니다. 또한 QA는 입력 클럭 주파수의 1/2, QB는 입력 클럭 주파수의 1/4, QC는 입력 클럭 주파수의 1/8, QD는 입력 클럭 주파수의 1/16가 된다는 것을 확인할 수 있고, 상향 카운터는 하강 에지에 의해, 하향 카운터는 상승 에지에 의해 trigger되는 것을 확인할 수 있습니다. 이때, 하향 카운터는 클럭 펄스가 상승 에지일 때 첫 번째 플립플롭의 출력이 1이 되고, 이에 2, 3, 4번째 플립플롭 모두 trigger되어 1로 바뀌므로 1111부터 0000으로 하나씩 감소하는 출력이 나타난다는 것을 알 수 있습니다. JK 플립플롭의 입력 J와 K가 모두 1일 때는 클럭 펄스에 의해 상태 변화가 촉발될 때마다 이전의 출력에 보수를 취하여 반전시킨다는 사실을 통해 매 클럭 펄스가 입력될 때마다 첫 번째 플립플롭은 토글된다는 것을 알 수 있습니다.

다음은 T 플립플롭으로 구현한 상향, 하향 카운터입니다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

<Figure 5> T 플립플롭으로 구현한 상향 카운터



<Figure 6> T 플립플롭으로 구현한 하향 카운터

T 플립플롭은 T가 1이면 JK 플립플롭에서 J와 K가 모두 1일 때와 동일하게 플립플롭의 이전 출력을 반전시킵니다. 따라서, JK 플립플롭 대신 T 플립플롭을 사용하여 위의 <Figure 5>와 <Figure 6>과 같이 구현할 수 있습니다.

동기식 카운터는 비동기식 카운터에서 발생할 수 있는 전달 지연의 문제를 없애기 위해 모든 플립플롭이 하나의 공통적인 클럭에 연결되어 동시에 동기화되어 동작하는 카운터를 의미합니다. 우선, 2비트 동기식 카운터에 대해 알아보면 2비트 동기식 카운터는 0부터 3까지를 차례대로 출력합니다. 따라서, 다음과 같은 상태도와 상태 여기표로 표현할 수 있습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 7> 2비트 동기식 이진 카운터의 상태도와 상태 여기표

2비트 동기식 카운터를 어떻게 구현해야 할지 알아보기 위해 <Figure 7>의 상태 여기표를 통해 카르노 맵을 그려보면 다음과 같습니다.

텍스트, 시계, 손목시계이(가) 표시된 사진

자동 생성된 설명

<Figure 8> 2비트 동기식 이진 카운터를 구현하기 위한 카르노 맵

따라서, JA와 KA로는 항상 1이 입력되어야 하고, JB와 KB로는 QA가 입력되면 되므로 2비트 동기식 이진 카운터는 다음 <Figure 9>와 같이 논리 회로를 설계하여 구현할 수 있습니다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

<Figure 9> 2비트 동기식 이진 카운터와 timing diagram

<Figure 9>의 timing diagram을 보면 JK 플립플롭을 이용하여 구현한 2비트 동기식 카운터가 00주터 시작하여 01, 10, 11까지 증가했다가 다시 00으로 돌아오는 것을 확인할 수 있습니다.

텍스트, 쇼지, 낱말맞추기게임, 시계이(가) 표시된 사진

자동 생성된 설명이제 3비트 동기식 카운터에 대해 알아보면 3비트 동기식 카운터는 000부터 111까지를 출력하므로 다음과 같은 상태 여기표로 나타낼 수 있습니다.

<Figure 10> 3비트 동기식 이진 카운터의 상태 여기표

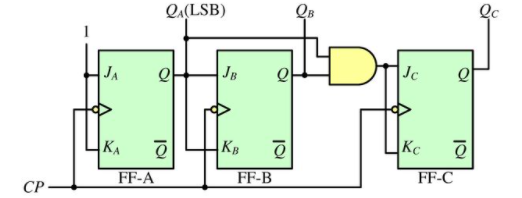
마찬가지로, 3비트 동기식 카운터를 어떻게 구현해야 할지 알아보기 위해 <Figure 9>의 상태 여기표를 통해 카르노 맵을 그려보면 다음과 같습니다.

텍스트, 시계이(가) 표시된 사진

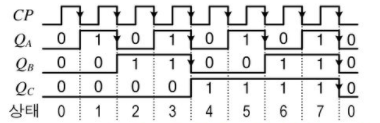
자동 생성된 설명텍스트, 시계이(가) 표시된 사진

자동 생성된 설명<Figure 11> 3비트 동기식 이진 카운터를 구현하기 위한 카르노 맵

따라서, JC와 KC로는 항상 QA와 QB의 논리곱이 입력되어야 하고, JB와 KB로는 QA가 입력되어야 하고, JA와 KA로는 항상 1이 입력되면 되므로 3비트 동기식 이진 카운터는 다음 <Figure 12>와 같이 논리 회로를 설계하여 구현할 수 있습니다.



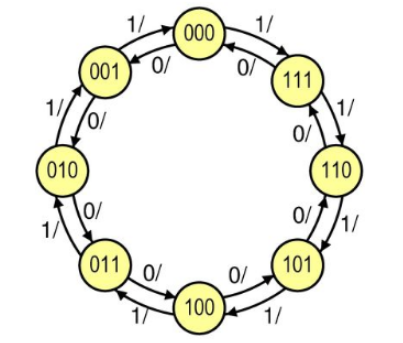
<Figure 12> 3비트 동기식 이진 카운터

3비트 동기식 카운터의 timing diagram을 확인해보면 다음과 같고 이는 000부터 111까지를 출력한 후 다시 000로 돌아가는 3비트 동기식 카운터의 특성을 잘 보여줍니다.

<Figure 13> 3비트 동기식 이진 카운터의 timing diagram

이때, <Figure 9>와 <Figure 13>을 보면 2비트, 3비트 동기식 카운터의 첫 번째 플립플롭 모두, 비동기식 카운터의 첫 번째 플립플롭와 동일하게 매 클럭 에지마다 토글이 되는 것을 확인할 수 있습니다.

JK 플립플롭과 T 플립플롭을 이용하여 비동기식 상향, 하향 카운터를 만들었던 것과 같이, JK 플립플롭을 이용하여 외부 입력 X에 따라 동기식 상향 카운터와 하향 카운터 둘 중 하나로 동작하는 동기식 상향, 하향 카운터를 만들 수 있습니다. 3비트 동기식 상향, 하향 카운터를 예시로 확인해 보면 3비트 동기식 상향, 하향 카운터는 다음과 같은 상태도로 표현할 수 있습니다.



<Figure 14> 3비트 동기식 상향, 하향 카운터의 상태도

이때, 3비트 동기식 상향, 하향 카운터는 외부 입력 X가 1일 때는 하향 카운터로, X가 0일 때는 상향 카운터로 동작한다고 가정하였습니다. 이러한 상태도를 이용하여 3비트 동기식 상향, 하향 카운터의 상태 여기표를 작성해 보면 다음과 같습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

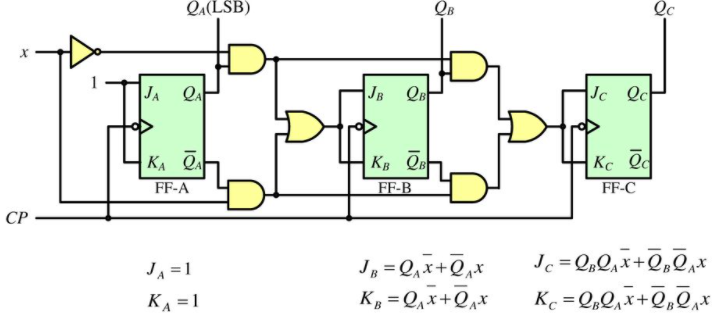
<Figure 15> 3비트 동기식 상향, 하향 카운터의 상태 여기표

텍스트, 낱말맞추기게임이(가) 표시된 사진

자동 생성된 설명3비트 동기식 상향, 하향 카운터의 논리 회로를 설계하기 위해 위의 <Figure 15>를 보고 카르노 맵을 작성하면 다음과 같습니다.

<Figure 16> 3비트 동기식 상향, 하향 카운터를 구현하기 위한 카르노 맵

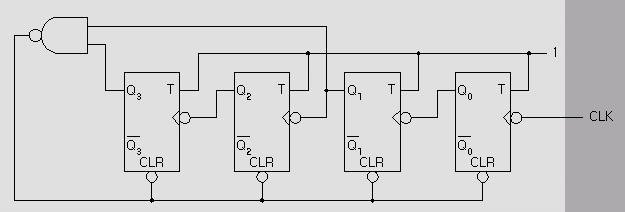
따라서, JC와 KC로는 (QB)(QA)(X’)+(QB’)(QA’)(X)가 입력되어야 하고, JB와 KB로는 (QA)(X’)+(QA’)(X)가 입력되어야 하고, JA와 KA로는 항상 1이 입력되면 되므로 3비트 동기식 상향, 하향 카운터는 다음 <Figure 17>과 같이 논리 회로를 설계하여 구현할 수 있습니다.



<Figure 17> 3비트 동기식 상향, 하향 카운터

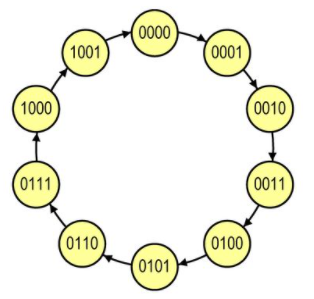
1. Decade Counter에 대해서 조사하시오.

비동기식 카운터를 이용하여 BCD 카운터를 만들 수 있는데, BCD 카운터란 위에서 설명한 0부터 15까지의 값을 반환하는 상향, 하향 카운터와 달리 0부터 9까지의 BCD를 반환하는 카운터를 의미합니다. 그렇다면 0000부터 1111까지를 출력한 후 다시 0000으로 돌아가는 상향 카운터와 달리, BCD 카운터는 0000부터 1001까지를 출력하면 다시 0000으로 돌아가야 합니다. 따라서, 상향카운터를 이용해 BCD 카운터를 설계하면 다음과 같습니다.



<Figure 18> T 플립플롭으로 구현한 비동기식 BCD 카운터

이때, <Figure 5>와 <Figure 18>을 비교해 보면 BCD 카운터는 상향 카운터에 NAND 게이트를 이용하여 수정한 것이라는 것을 알 수 있습니다. <Figure 18>을 보면 NAND 게이트의 출력은 각 플립플롭의 비동기식 CLR 단자에 연결되어 있으며 이 비동기식 CLR 단자는 active low 신호에 의해 동작하는 것을 알 수 있습니다. 따라서, NAND 게이트의 출력이 0이 되면 모든 플립플롭의 출력은 클럭과 무관하게 0이 되어버리고 이 NAND 게이트는 카운터가 1010을 출력하려고 할 때 Q1과 Q3가 모두 1이므로 1을 출력하여 BCD 카운터가 1010 이후의 값들을 출력하지 않고 다시 0000으로 돌아갈 수 있게 하는 역할을 합니다.

동기식 카운터를 이용해서도 BCD 카운터를 만들 수 있는데 우선 동기식 카운터의 상태도를 그려보면 다음과 같습니다.

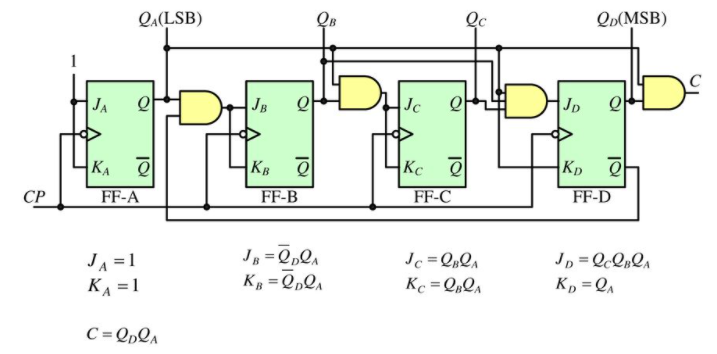
<Figure 19> 동기식 BCD 카운터의 상태도

이를 이용하여 동기식 BCD 카운터의 상태 여기표를 그려보면 다음과 같습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 20> 동기식 BCD 카운터의 상태 여기표

이때, 자리 올림수 출력 C는 BCD 카운터의 출력이 1001일 때 논리 1가 되도록 합니다. <Figure 20>의 동기식 BCD 카운터의 상태 여기표를 보고 카르노 맵을 그려 동기식 BCD 카운터의 회로를 설계해보면 다음 <Figure 21>과 같습니다.

<Figure 21> 동기식 BCD 카운터

위의 <Figure 21>를 보면 동기식 BCD 카운터는 JK 플립플롭 4개와 4개의 AND 게이트를 이용하여 구현이 가능합니다.

1. FSM(finite-state Machine)에 대해서 조사하시오.

FSM은 finite-state Machine의 약자로, 유한 상태 기계 또는 finite automaton(유한 오토마톤)이라고 불립니다. FSM은 컴퓨터 프로그램과 전자 논리 회로를 설계하는 데에 유용하게 사용되는 수학적 모델이며 흔히 상태 기계라고 부릅니다. FSM은 유한한 개수의 상태를 가질 수 있으며 임의의 시간에 오직 하나의 상태만을 가지고 그때의 FSM의 상태를 현재 상태(Current State)라고 합니다. FSM는 특정 조건을 만족하게 하는 어떠한 사건에 의해 한 상태에서 다른 상태로 변화하는데 이를 전이(Transition)라고 합니다.

FSM를 이해하기 위해서는 state diagram에 대해 알아보아야 하는데 state diagram은 FSM에서 상태와 상태천이를 나타내기 위하여 도식화해서 표현하는 그림을 의미합니다. 이러한 상태도를 이용하면 논리 회로에서 순차회로의 카운터와 같은 응용 회로에서 각각의 상태를 구분하고 입력에 따라 상태 변화를 도식화할 수 있습니다.

1. 기타 이론

동기식 카운터의 예로는 링 카운터, 존슨 카운터 등이 있습니다. 링 카운터란 한 번에 단 한 개의 플립플롭의 출력만이 1이 되고 다른 나머지 플립플롭의 출력은 모두 0이 되는 카운터를 의미합니다. 링 카운터는 1000, 0100, 0010, 0001을 반복하여 출력하므로 D 플립플롭으로 구현한 링 카운터의 상태 여기표를 작성해 보면 다음과 같습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

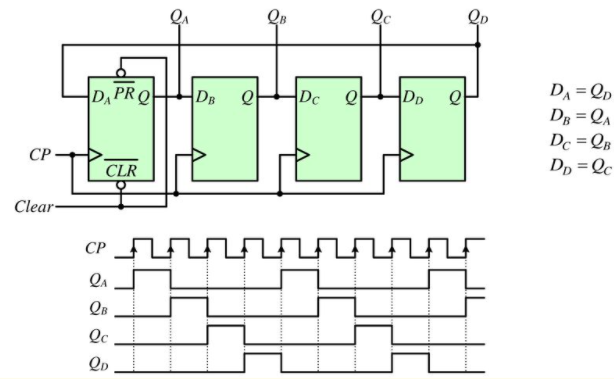
<Figure 22> 링 카운터의 상태 여기표

텍스트, 낱말맞추기게임이(가) 표시된 사진

자동 생성된 설명따라서, 이를 토대로 링 카운터를 설계하기 위해 카르노 맵을 작성해 보면 다음과 같습니다.

<Figure 23> 링 카운터를 구현하기 위한 카르노 맵

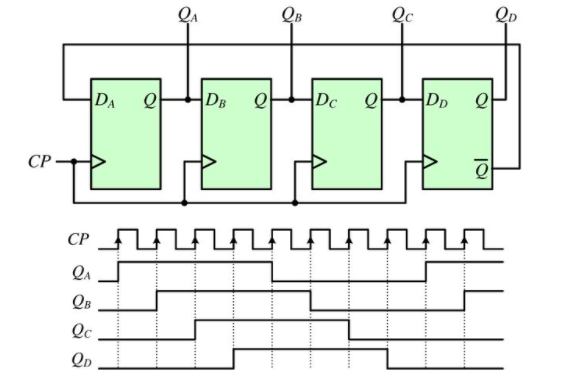
그러므로 다음과 같이 논리 회로를 설계할 수 있고 초기에는 비동기식 CLR 단자를 0으로 하여 모든 플립플롭의 출력을 0으로 세팅한 다음 첫 번째 플립플롭부터 차례대로 1을 출력할 수 있게 회로를 설계할 수 있습니다.



<Figure 24> 링 카운터와 timing diagram

이때, QD가 1이 되면 QD는 다시 첫 번째 플립플롭의 DA로 입력되어 첫 번째 플립플롭의 출력이 1이 될 수 있게 합니다.

n개의 플립플롭으로 구성된 링 카운터는 n개의 서로 다른 상태를 나타낼 수 있지만 존슨 카운터를 구현하면 n개의 플립플롭을 이용하여 2n개의 서로 다른 상태를 나타낼 수 있습니다. 다음은 존슨 카운터의 회로도와 timing diagram입니다.



<Figure 25> 존슨 카운터와 timing diagram

QD가 DA로 들어가는 링 카운터와 달리 존슨 카운터는 QD’이 DA로 들어간다는 차이가 있습니다. 4비트 존슨 카운터의 상태표를 작성해 보면 다음과 같습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 26> 존슨 카운터의 상태표