

2018년 2학기 컴퓨터공학실험Ⅱ
CSE3016-05반 12주차 예비 보고서

학번: 20171665

이름: 이 선 호

2018. 12. 7

목 차

I	Counter		
1.	Counter와 Counter의 예시	3
2.	Decade Counter	5
3.	비동기식과 동기식 Counter	6
II	Finite State Machine		
1.	Finite State Machine	7
III	기타 이론		
1.	이진 병렬 승산기	7

I Counter

1. Counter와 Counter의 예시

Counter는 연속적인 clock 입력 또는 데이터 입력에 따라 고정된 sequence에 따른 움직임을 보이는 장치이다. 출력은 주로 플립플롭에 저장된 내용인 시스템의 상태로 표현되기 때문에 state table에서 출력 열을 작성하지 않는다.

플립플롭을 사용하여 Counter의 순차 회로를 제작할 수 있다. 예를 들어, 다음과 같은 sequence를 반복하는 Counter를 제작한다고 가정한다.

0, 3, 2, 4, 1, 5, 7, 0, ...

Counter의 상태 값이 반드시 위의 sequence처럼 1씩 증가하거나 1씩 감소하는 방향으로만 변해야 하는 것은 아니며, 또 반드시 0부터 세기 시작해야만 하는 것은 아니다. 넓은 의미로 Counter를 말할 때는 정해진 개수의 상태 값을 순환하도록 구성만 되면 Counter로 취급한다. Counter는 일반적으로 데이터 입력을 필요로 하지 않지만, 데이터 입력을 사용해도 무관하다. 예를 들어, 위의 sequence에 따라 입력 x에 1을 입력할 때 다음 상태로 이동하고 0을 입력하면 현재 상태를 유지하는 Counter를 제작하는 것도 가능하다. 그러나 일반적인 동기식 Counter의 예를 들기 위해 여기서는 다른 데이터 입력 없이 오직 clock이 활성화될 때 다음 상태로 이동한다고 가정한다. 위의 sequence에 따라 회로의 state diagram을 그리면 state는 6을 제외하여 0부터 7까지 총 7개가 생긴다. 그러나 실제 회로에서는 십진수를 다룰 수 없기 때문에 이진수로 나타내야 한다. 따라서 7개의 상태를 나타내기 위해 3개의 비트(2^3 이 8이어서 3개의 비트로 최대 8개의 상태를 나타낼 수 있음)를 사용한다. 각 7개의 상태를 이진수로 나타냈을 때 현재 상태부터 다음 상태로 이동할 수 있는 State Table을 그리면 다음과 같다.

q_1	q_2	q_3	q_1^*	q_2^*	q_3^*
0	0	0	0	1	1
0	0	1	1	0	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	1	1	1
1	1	0	X	X	X
1	1	1	0	0	0

위의 표처럼 상태 0(000)의 다음 상태는 3(011)이고, 상태 1(001)의 다음 상태는

5(101)이다. 이처럼 각 상태에 관하여 다음 상태의 이진수를 같은 행에 적는다. 상태가 6(110)일 때 다음 상태가 'XXX'인 이유는 가정했던 sequence에 6이 존재하지 않기 때문에 다음 상태가 정해지지 않은 것이다.

State Table을 바탕으로 이진수의 각 자릿수마다 현재 상태에서 다음 상태로 어떻게 변하는지를 참고하여 11주차 결과 보고서 추가이론에서 조사했던 Excitation Table을 사용해서 특정 플립플롭의 입력이 각 경우마다 어떻게 되는지를 찾는다. 찾은 플립플롭 입력과 현재 상태에 관하여 카르노 맵을 그려서 각 플립플롭 입력의 SOP 식을 구하고, 이를 바탕으로 적절한 논리 게이트를 사용하여 회로를 구성한다.

위의 예시에서 이진수의 모든 각 자리마다 D 플립플롭을 사용한다고 가정한다. 그리고 q_1 , q_2 , q_3 의 상태를 변화시키는 D 플립플롭의 입력을 각각 D_1 , D_2 , D_3 이라고 한다. 입력 값에 따른 D 플립플롭 입력 값을 State Table과 같이 작성하면 아래와 같다.

q_1	q_2	q_3	q_1^*	q_2^*	q_3^*	D_1	D_2	D_3
0	0	0	0	1	1	0	1	1
0	0	1	1	0	1	1	0	1
0	1	0	1	0	0	1	0	0
0	1	1	0	1	0	0	1	0
1	0	0	0	0	1	0	0	1
1	0	1	1	1	1	1	1	1
1	1	0	X	X	X	X	X	X
1	1	1	0	0	0	0	0	0

각 플립플롭 입력에 관해 카르노 맵을 그리면 아래와 같다.

$q_2q_3 \backslash q_1$	0	1
00	0	0
01	1	1
11	0	0
10	1	X

$$D_1 = q_2'q_3 + q_2q_3'$$

$q_2q_3 \backslash q_1$	0	1
00	1	0
01	0	1
11	1	0
10	0	X

$$D_2 = q_1'q_2'q_3' + q_1'q_2q_3 + q_1q_2'q_3$$

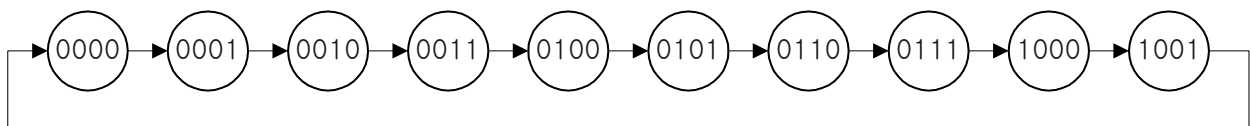
$q_2q_3 \backslash q_1$	0	1
00	1	0
01	0	1
11	1	0
10	0	X

$$D_3 = q_2'$$

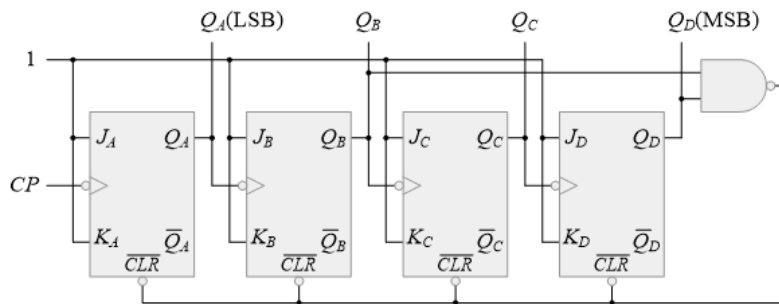
위의 D 플립플롭 입력의 SOP식을 바탕으로 논리 게이트를 사용하여 회로도를 구성하면 Counter 제작이 완성된다.

2. Decade Counter

일상생활에서 인간은 10진수를 주로 사용하기 때문에 Counter에서도 0에서 9까지의 state를 갖고, 9에서 다시 0으로 가는 Decade Counter(BCD Counter)가 고안되었다. 위의 설명과 예시에서와 마찬가지로 유사한 방법으로 회로를 구현하지만, 여기서 한 가지 유의할 점은 9에서 0으로 갈 때 이진수의 모든 자릿수를 다시 0으로 초기화 하는 reset과 관련된 wire를 사용한다는 것이다. 여러 개의 Decade Counter를 사용하여 두 자리 이상의 십진수에 관한 Counter를 제작한다고 하면, 이 reset 신호를 가지고 다음 자릿수에 1을 증가시키라는 신호를 보낼 수도 있다. 10개의 상태를 사용하기 때문에 4개의 비트를 사용해야 하며, 이를 State Diagram과 Truth Table로 나타내면 아래와 같다.



Count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



앞에서와 같은 방법으로 Counter 회로를 제작하면 위와 같은 논리 회로도가 나온다.

3. 비동기식과 동기식 Counter

Counter 예시에서 살펴본 것처럼 연속적인 Clock 입력에 따라 상태가 고정된 sequence에 따른 움직임을 보이는 장치가 동기식 Counter이면, 비동기식 Counter는 동기식 Counter에서 사용된 것과 동일하게 Clock에 의해 동작하는 플립플롭을 이용하지만, 실제로 모든 플립플롭에 관해 Clock을 사용하는 것이 아니라 각 플립플롭이 이전 플립플롭의 천이의 의해 활성화(Trigger) 되는 장치를 말한다. 그래서 이를 Ripple Counter라고도 부른다.

예를 들어, 첫 번째 플립플롭의 입력에만 Clock Pulse가 입력이 되고, 다른 플립플롭은 각 플립플롭의 출력을 다음 플립플롭의 Clock 입력으로 사용하는 것이 비동기식 Counter이다. 앞에서 설명한 Decade Counter에서 낮은 자리의 Counter가 9에서 0으로 되면서 reset될 때, 그 신호를 다음 자리의 Counter에 전하여 그 자리의 Count를 1만큼 증가시키는 것도 비동기식의 예시가 된다.

비동기식 Counter는 동기식 Counter에 비해 회로가 간단하지만 전달 지연이 커진다는 단점이 있다.

II Finite State Machine

1. Finite State Machine

State Machine은 각각의 상태들을 어떤 조건에 따라 연결해 놓은 것을 말한다. 이 중에서 Finite State Machine(FSM)은 유한한 개수의 상태를 가질 수 있는 추상 기계를 말하며, 한 번에 반드시 하나의 상태만 가지게 된다. FSM의 동작은 일반적으로 하나의 입력을 받고 그에 의거하여 현재 상태에서부터 다음 상태로 전이되고 하나의 출력을 가진다. 따로 출력 값을 가지지 않을 경우 Counter에서처럼 플립플롭에 저장된 값이 출력이다. 상태들을 하나의 vertex로 두고 이들 간의 이동 경로를 edge로 표현하면 이는 그래프가 되고 state diagram이라고 부른다. FSM을 사용하는 이유는 가능한 상태와 전이를 명확히 규정할 수 있고 상태 중복을 피할 수 있어서 기계의 동작을 분명하게 규정할 수 있기 때문이다.

FSM은 순차 논리로 이루어진 기계여서 조합 논리와는 다르다. 조합 논리는 현재의 출력이 오직 현재의 입력과만 영향을 주고받는다면, 순차 논리는 이전 상태 또는 입력이 계속 현재 상태 또는 출력에도 영향을 미친다.

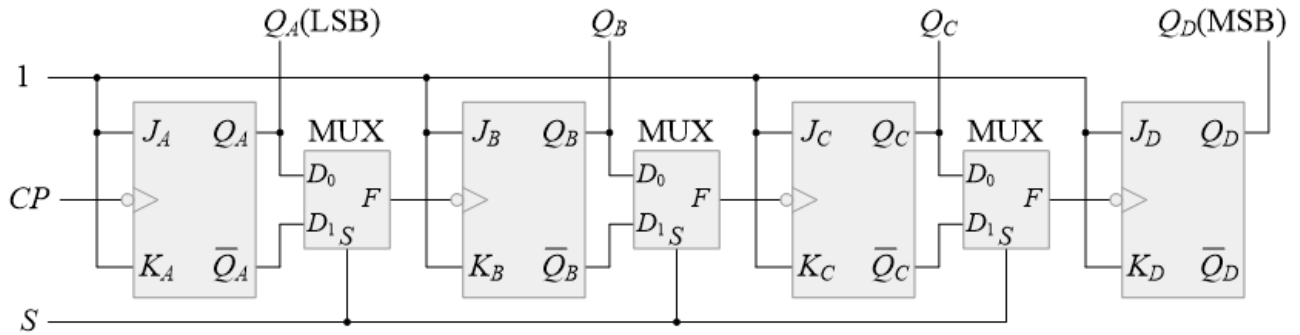
FSM의 동작은 형태에 따라서 Moore State Machine과 Mealy State Machine으로 나뉜다. Moore State Machine은 출력이 오직 현재 상태에 의존한다. Moore Machine의 장점은 출력이 따로 input에 의존하지 않고 오직 현재 상태에 달려있기 때문에 단순화시키기 좋다. 처음 입력에 관해서는 출력이 결정되지 않는데, 이는 오직 다음 출력이 현재 상태에만 의존하기 때문에 처음 입력되었을 때는 현재 상태를 알 수가 없다. 반면에 Mealy Machine은 출력이 현재 상태에도 의존하지만 현재 입력에도 같이 영향을 받는다. 그래서 Moore보다는 상대적으로 회로 상태 전이를 이해하기 어렵다는 단점이 있지만, 일반적으로 Mealy는 Moore보다 더 적은 state를 갖게 되는 장점이 존재한다.

III 기타 이론

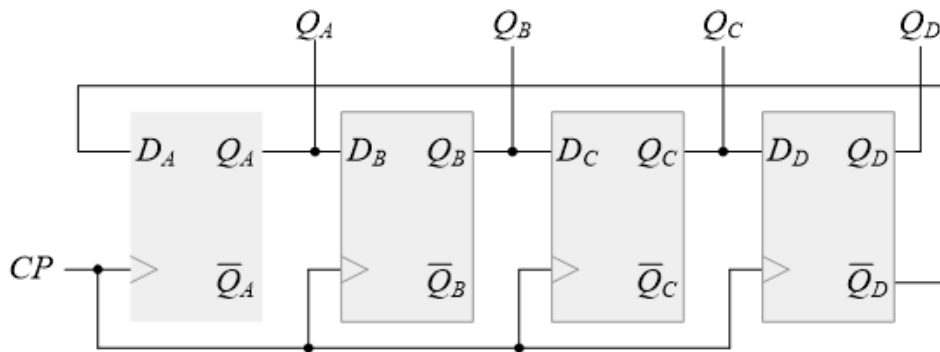
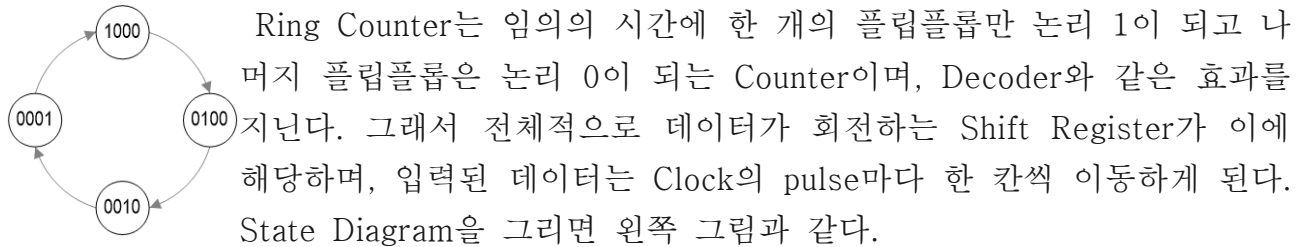
1. 다양한 Counter

위에서 살펴본 Counter 외에도 다양한 종류의 Counter가 존재한다. 상향/하향 Counter는 어떤 특정 입력이 0으로 들어오면 현재 상태에서 다음 상태로 이동하고, 입력이 1로 들어오면 현재 상태에서 이전 상태로 이동하는 Counter이다. 주로 MUX를

사용하여 선택 입력 값에 따라 순차적으로 연결된 플립플롭의 Clock 입력으로 이전 플립플롭에서 출력된 Q와 Q' 중 하나의 출력을 연결시켜서 구현한다. JK 플립플롭과 MUX를 사용하여 구현한 상향/하향 Counter의 논리 회로도를 그리면 아래와 같다.



Preset Counter는 0보다 큰 수의 상태부터 시작 가능하며, preset 입력을 받아서 값이 1이면 다시 원래 시작한 상태로 초기화되는 Counter이다.



Johnson Counter는 동일한 수의 플립플롭을 가지고 Ring Counter의 2배의 출력을 나타내기 위해 사용되며 Ring Counter의 마지막 플립플롭의 Q' 출력을 첫 번째 플립플롭의 입력으로 엇갈리게 연결한 Counter이다. 단점은 사용되지 않는 초기 상태가 주어지면 사용되지 않는 순서만이 계속 반복되어 무한 loop에 빠질 수가 있다.