13주차 예비보고서

전공: 경영학과 학년: 3학년 학번: 20190963 이름: 한다현

**1.**

레지스터는 플립플롭의 집합으로, 플립플롭은 한 비트의 정보를 저장하는 기능이 있기 때문에 n비트 레지스터의 경우 n개의 플립플롭으로 구성되어 있고, n비트의 이진 정보를 저장할 수 있다. 시프트 레지스터란 저장되어 있는 비트 데이터를 한 방향이나 양방향으로 측면 이동시킬 수 있는 기능을 가진 레지스터를 의미한다. 시프트 레지스터는 레지스터의 한 종류이며 여러 개의 플립플롭을 순차적으로 연결하여 한 플립플롭의 출력이 그 옆에 있는 플립플롭의 입력이 되도록 설계한다. 시프트 레지스터는 입력과 출력의 방법에 따라 직렬입력-직렬출력, 직렬입력-병렬출력, 병렬입력-직렬출력, 병렬입력-병렬출력 4가지 형태로 분류된다. 먼저 직렬입력-직렬출력 시프트 레지스터(SISO)는 1개의 입력 단자와 1개의 출력 단자를 가지며, 첫 번째 플립플롭에 데이터를 입력하면 그 데이터가 clock pulse에 따라 마지막 플립플롭에서 출력된다. 즉, 데이터를 한 번에 1비트씩 입력시켜 차례로 1비트씩 출력하는 레지스터이다. 4비트 SISO의 초기값이 0000인 상태에서 1을 입력하면 각 플립플롭에는 1000이 저장되고 출력은 0이 된다. 다음 clock pulse에 1이 입력되면 1100이 저장되며 출력은 0이 된다. 이러한 방식으로 SISO는 1비트의 데이터를 순차적으로 입력 받고 1비트의 데이터를 순차적으로 출력한다. 직렬입력-병렬출력 시프트 레지스터(SIPO)는 직렬로 입력되어 각 플립플롭에 저장된 데이터를 1비트씩 직렬로 꺼내는 것이 아니라 모든 데이터를 동시에 출력하는 레지스터이다. 4비트 SIPO에 차례로 1001이 입력되면 각 플립플롭은 1001을 저장한다. 이후 출력을 하게 되면 4개의 플립플롭이 저장하고 있던 데이터를 동시에 출력하여 1001이 출력된다. 병렬입력-직렬출력 시프트 레지스터(PISO)는 n비트의 입력 데이터를 레지스터에 병렬로 입력하고, 이 데이터를 하나의 출력 단자를 통해 순차적으로 출력하는 레지스터이다. 4비트 데이터 1001이 4비트 PISO에 동시에 입력되면 각 플립플롭은 1001을 저장하고, 이후 마지막 플립플롭의 출력단자에서 1001이 차례로 출력된다. 마지막으로 병렬입력-병렬출력 시프트 레지스터(PIPO)는 입력 데이터가 동시에 레지스터에 입력되면 clock pulse 입력에 따라 각 플립플롭에 저장되어 있던 데이터가 병렬 출력되는 레지스터이다. PIPO는 이전 플립플롭의 출력이 다음 플립플롭의 입력이 되지 않으며, 4비트 PIPO에 1001이 병렬로 입력된 후 CLK가 입력되면 1001이 동시에 출력된다.

**2.**

링 카운터는 마지막 플립플롭의 출력 Q를 처음 플립플롭에 입력하는 시프트 레지스터 회로를 말하며 순환 시프트 레지스터라고도 한다. 앞서 조사한 시프트 레지스터는 처음 플립플롭에 새로운 데이터가 입력되었다. 하지만 링 카운터는 첫 번째 플립플롭에 새로운 데이터가 입력되는 것이 아니라, 마지막 플립플롭에서 출력되는 데이터가 첫 번째 플립플롭의 입력으로 돌아온다. 이러한 방식이 순환 구조를 가지기 때문에 순환 시프트 레지스터라고 하는 것이다. 상승에지에서 트리거 되는 4비트 링 카운터를 예로 들어보면 4비트 링 카운터는 4개의 플립플롭으로 구성되어 있고 이전 플립플롭의 출력이 다음 플립플롭의 입력이 되며 네 번째 플립플롭의 출력은 첫 번째 플립플롭의 입력이 된다. 이 링 카운터에 1000이라는 4비트 데이터가 저장되어 있다면 clock pulse의 상승에지에서 데이터가 오른쪽으로 시프트하여 0100이 저장된다. 가장 왼쪽 비트가 0이 되는 이유는 1000에서 가장 오른쪽에 있는 0이 처음 플립플롭에 입력되었기 때문이다. 다시 clock pulse가 상승하면 링 카운터에는 0010이 저장되고, 다음 clock pulse의 상승에지에서는 0001이 링 카운터에 저장된다. 예시와 같이 4개의 플립플롭으로 구성된 4비트 링 카운터를 사용하면 1000, 0100, 0010, 0001 4가지 상태를 가질 수 있기 때문에 4까지 카운트가 가능하고, 0001까지 도달한 뒤에는 자동으로 1000부터 다시 카운트할 수 있다. 따라서 n비트 링카운터는 n개의 플립플롭으로 구성되어 있으며 n의 수까지 셀 수 있다.

**3.**

업 카운터는 수가 증가하는 카운터이고, 다운 카운터는 수가 감소하는 카운터이다. 4비트 카운터를 예로 들면, 4비트 업 카운터는 0부터 15까지의 수를 셀 수 있으며, 0부터 시작하여 1, 2, 3, … 15의 순서로 동작한다. 15까지 상태가 변화한 뒤에는 다시 0으로 돌아와 같은 동작을 반복한다. 반면 4비트 다운 카운터도 0부터 15까지의 수를 셀 수 있지만 0부터 시작하여 수가 증가하는 것이 아니라 15부터 시작하여 수가 감소하는 방식으로 동작한다. 4비트 다운 카운터는 15부터 시작해서 14, 13, 12 … 0의 순서로 동작하며 0까지 상태가 변화한 뒤에는 다시 15로 돌아와 같은 동작을 반복한다. 업 카운터와 다운 카운터는 별개의 회로로 각각 설계될 수도 있지만 제어 입력을 추가하면 하나의 회로로 설계가 가능하다. 3비트 업 카운터의 상태는 000, 001, 010, 011, 100, 101, 110, 111으로 총 8개이다. 3비트 다운 카운터의 상태는 111, 110, 101, 100, 011, 010, 001, 000으로 총 8개이다. 두 카운터의 상태를 보면 다운 카운터의 상태는 업 카운터의 보수임을 알 수 있다. 111의 보수는 000이고, 110의 보수는 001이기 때문이다. 따라서 제어 입력에 따라 각 플립플롭의 출력 Q를 사용해 상태를 결정할 것인지 Q’을 사용해 상태를 결정할 것인지를 결정하면 업 카운터와 다운 카운터를 한 개의 회로로 설계할 수 있다. 3비트 2진 카운터에 제어 입력 UP과 DOWN을 추가한다고 가정하면, UP에 1이 입력되면 각 플립플롭의 Q를 통해 카운터의 상태를 결정하고, DOWN에 1이 입력되면 각 플립플롭의 Q’을 통해 카운터의 상태를 결정하는 방식이다. 예를 들어, 카운터의 각 플립플롭의 Q가 000일 때, UP에 1이 입력되면 000이, 즉 0이 출력되어 업 카운터의 역할을 하는 것이고, DOWN에 1이 입력되면 Q’인 111이, 즉 15가 출력되어 다운 카운터의 역할을 하는 것이다.

**4.**

리플 카운터는 비동기 카운터에 속하며, 앞 단의 출력이 다음 단의 clock pulse로 입력된다. 모든 플립플롭에 clock pulse를 입력하는 동기식 카운터와 다르게, 리플 카운터는 앞 단의 플립플롭의 출력이 다음 단의 clock pulse로 입력되어 다음 플립플롭을 트리거하기 때문에 동기식 플립플롭에 비해 더 많은 지연 시간이 발생하며 동작속도에 제한을 받게 된다. 리플 카운터는 주로 보수 기능이 있는 JK 플립플롭이나 T 플립플롭을 이용한다. 이 때 이전 플립플롭의 값이 1에서 0으로 반전되는 경우, 즉 Q’의 값이 1이 되는 경우에 다음 플립플롭이 작동된다. 4비트 리플 카운터를 예로 들면, 0000의 상태에서 첫 번째 플립플롭에 clock pulse 1이 입력되면 첫 번째 비트가 반전되기 때문에 0001의 상태가 된다. 이 경우에는 첫 번째 플립플롭이 1에서 0으로 반전된 것이 아니라, 0에서 1로 반전되었기 때문에 Q’의 값은 0이 되어 두 번째 플립플롭이 동작하지 않았다. 이 후 다시 clock pulse 1이 입력되면 첫 번째 비트는 1에서 0으로 변화하며, 첫 번째 플립플롭의 Q’의 값이 1이 되었기 때문에 두 번째 플립플롭이 트리거 되고 두 번째 비트의 값이 0에서 1로 반전되기 때문에 0010의 상태가 된다. 이 때 두 번째 플립플롭이 0에서 1로 반전되었기 때문에 세 번째 플립플롭은 동작하지 않는다. 다시 clock pulse 1이 입력되면 첫 번째 플립플롭의 값은 0에서 1로 반전되며 Q’의 값이 0이기 때문에 다음 플립플롭은 동작하지 않고 상태는 0011이 된다. Clock pulse에 1이 다시 입력되면 첫 번째 플립플롭에서 1에서 0으로 반전되기 때문에 두 번째 플립플롭이 동작하고, 두 번째 플립플롭 역시 1에서 0으로 반전되기 때문에 세 번째 플립플롭이 동작한다. 세 번째 플립플롭에서는 0에서 1로 반전되기 때문에 네 번째 플립플롭은 동작하지 않으며 리플 카운터의 상태는 0100이 된다. 이와 같은 방식으로 4비트 리플 카운터는 0000에서 1111까지 15개의 수를 셀 수 있지만, 앞서 조사했던 바와 같이 clock pulse가 모든 플립플롭에 동시에 입력되지 않고, 앞 단의 플립플롭의 출력에 영향을 받기 때문에 많은 지연 시간이 발생한다는 단점이 있다.

**5.**

시프트 레지스터를 활용한 카운터에는 링 카운터 외에도 존슨 카운터가 있다. 앞서 조사한 링 카운터는 마지막 단의 플립플롭의 Q가 처음 플립플롭의 입력이 되었다. 하지만 존슨 카운터는 마지막 단의 플립플롭의 Q’이 처음 플립플롭의 입력이 된다. 즉, 링 카운터는 정상출력을 피드백 받지만 존슨 카운터는 반전출력을 피드백 받는다. 예를 들어 3비트 존슨 카운터가 있다면, 이 카운터가 갖는 상태는 000, 100, 110, 111, 011, 001로 총 6개이다. 이는 플립플롭의 수의 2배이기 때문에 존슨 카운터의 MOD는 2\*n이라고 할 수 있다. 처음 상태 000에서 clock pulse 1이 가해지면, 가장 오른쪽 비트의 값인 0의 반전 출력인 1이 처음 플립플롭에 입력되기 때문에 카운터의 다음 상태는 100이 된다. 100인 상태에서 다시 clock pulse가 가해지면, 가장 오른쪽 플립플롭의 반전 출력인 1이 처음 플립플롭에 입력되고, 다른 값들은 오른쪽으로 이동하게 되어 카운터의 상태는 110이 되는 형식이다.

참조:

디지털 논리설계(황의철)

디지털 논리와 컴퓨터 설계(Mano, Morris)

디지털 공학(조현철)

디지털 회로 설계(박인규)