14주차 예비보고서

전공: 경영학과 학년: 3학년 학번: 20190963 이름: 한다현

**1.**

Finite state machine이란 유한 상태 기계이며, 시스템이 가질 수 있는 유한개의 상태로 구성된 상태 기계이다. 이 시스템은 특정 상태가 되면 시스템 입력이 바뀌더라도 일정한 시간 동안 그 상태로 남아 있을 수 있어야 하기 때문에 상태를 기억할 수 있는 능력이 필요하다. 그렇기 때문에 상태를 저장하기 위해 주로 플립플롭을 사용하여 설계한다. 또한 한 상태에서 다른 상태로 변화할 때, 다음 상태는 입력과 현재 상태에 의해 결정되며, 상태 기계의 출력 역시 현재 상태와 입력에 의해 결정된다. 즉, 현재 상태에서 다음 상태로 변화하기 위해서는 현재 상태가 무엇인지, 상태 기계에서 입력 받는 데이터가 무엇인지에 따라 결정된다. Finite state machine은 크게 moore machine과 mealy machine으로 구분되는데, mealy machine은 다음 상태와 출력을 결정할 때 현재 상태와 입력을 모두 고려한다. 반면 moore machine은 다음 상태를 결정할 때는 현재 상태와 입력을 모두 고려하지만, 출력을 결정할 때는 현재 상태만을 고려한다. Mealy machine과 moore machine에 대해서는 다음 문항들에서 다루도록 하겠다. 상태 기계를 설계하는 과정에서 필요한 것이 상태도(state diagram)이다. 상태도는 상태 기계의 상태가 어떻게 변화하는지를 diagram으로 표현한 것이다. 이를 사용하면 상태 기계의 동작을 신속하게 분석할 수 있으며 상태 기계를 설계할 때 유용하게 사용된다. 예를 들어 00의 상태에서 1이 입력되었을 때 상태가 01로 변화하고 0이 입력되었을 때 11로 변화한다면, 이를 도형과 화살표를 이용해 간단하고 보기 쉽게 표현할 수 있다. 따라서 상태 기계를 설계하기 위해서는 먼저 적절한 상태도를 생성한다. 이 상태도를 통해 상태의 변화를 분석하고, 분석한 것을 바탕으로 카르노 맵, 플립플롭, 논리 게이트 등을 사용해 상태 기계를 설계할 수 있다. Finite state machine의 예로는 sequential counter(순차 계수기)가 있다. 순차 계수기는 0부터 1씩 순차적으로 증가하는 경우도 있지만, 정해진 상태의 순서에 따라 계수를 진행하기도 한다. 예를 들어, 1, 3, 0, 2, 1, 3, 0, 2와 같은 순서로 상태가 변화하면서 계수가 진행된다. 3비트 크기의 순차 계수기가 있다고 가정하면, 이 계수기는 000부터 111까지 8개의 유한한 상태를 가질 수 있다. 이 순차 계수기의 상태 변화의 순서가 000, 011, 101, 111, 100, 000, 011, 101, 111 … 의 순서라면, 000부터 011, 101, 111로 이동하는 동안 계수가 진행될 것이고, 111 이후에 다시 초기 상태인 000으로 돌아갈 것이다. 이러한 방식으로 계수기를 설계한다면, 사용하는 상태의 개수를 통해 몇개의 수를 카운트할지 결정할 수 있다. 예를 들면 위의 예시에서는 4개의 상태를 사용하였기 때문에 4가지 수를 카운트할 수 있고, 5개의 상태를 사용하여 설계한다면, 5가지의 수를 계수할 수 있을 것이다.

**2.**

Mealy machine은 finite state machine의 한 종류이다. Mealy machine은 다음 상태를 결정할 때, 현재 상태와 입력 값을 고려한다. Mealy machine의 상태가 00, 01, 10 세 개가 있고 현재 상태가 00이라면, 다음 상태는 입력이 0인지, 1인지에 의해 결정된다. 현재 상태가 00이고, 입력이 1일 때 01로 상태가 변화하고, 입력이 0일 때는 10으로 상태가 변화하는 mealy machine이 있다면, 현재 상태 00일 때 1이 입력되면 이 상태 기계의 다음 상태는 01이 될 것이고, 현재 상태가 00일 때 0이 입력되면 이 상태 기계의 다음 상태는 10이 될 것이다. 현재 상태가 01일 때도 마찬가지로, 상태 기계의 입력이 0인지, 1인지에 따라 다음 상태가 결정될 것이면, 현재 상태가 10일 때에도 상태 기계의 입력이 0인지, 1인지에 따라 다음 상태가 결정될 것이다. Mealy machine의 출력 또한 현재 상태와 상태 기계의 입력으로 결정된다. 앞의 예시를 다시 사용해보면, 어떤 mealy machine의 현재 상태가 00일 때 1이 입력되면 01로 상태가 변화하고 0이 입력되면 10으로 상태가 변화한다. 이 상태 기계의 출력이 00에서 01로 상태가 변화할 때 1이 된다고 하자. 이 경우 출력이 결정되기 위해서는 현재 상태가 00이어야 하고, 1이라는 데이터가 입력되어야 한다. 즉, 현재 상태가 00이더라도 상태 기계의 입력이 0이라면 상태는 10으로 변화할 것이고, 1이 출력되지 않을 것이다. 따라서 mealy machine에서는 현재 상태와 입력 모두가 상태 기계의 출력을 결정한다. Mealy machine은 입력이 변하거나 현재 상태가 변할 때 출력도 함께 변하며, 이것이 출력 신호의 타이밍 제어를 어렵게 만든다는 단점이 있지만, 같은 기능을 하는 시스템을 설계할 때 moore machine을 사용할 때보다 더 적은 상태의 수가 필요하다는 장점이 있다.

**3.**

Moore machine도 finite state machine의 한 종류이며 mealy machine을 변형시켜 설계할 수 있다. Moore machine도 mealy machine과 마찬가지로 현재 상태와 입력을 통해 다음 상태를 결정한다. 00, 01, 10, 11을 상태로 갖는 moore machine에서 현재 상태가 00이고, 다음 상태가 01 또는 10으로 변화한다면, 현재 상태 00에서 상태 기계의 입력이 0인지 1인지에 따라 다음 상태가 결정된다. 하지만 출력을 결정하는 방식에서 mealy machine과 차이가 있다. Mealy machine이 현재 상태와 입력을 모두 고려해 출력은 결정한다면 moore machine은 현재 상태만으로 출력을 결정하고, 입력은 상태 기계의 출력에 영향을 주지 않는다. 00, 01, 10, 11을 상태로 갖는 moore machine에서 상태가 11인 경우에 1을 출력한다고 가정하면, moore machine이 11이라는 상태를 유지하는 동안에는 입력의 값과 상관없이 항상 1을 출력한다. 현재 상태가 01일 때 1이 입력되면 11로 상태가 변화한다고 할 때, 01에서 1이 입력되어 11로 상태가 변화하면 1이 출력된다. 이 후에도 현재 상태가 11일 때 1이 입력되면 상태가 변화하지 않고 0일 때 상태가 00으로 변화하는 moore machine에서는 현재 상태가 11일 때 1이 입력되면 상태가 변하지 않은채로 1을 출력하고, 0이 입력되면 상태가 00으로 변화하지만 1을 출력한다. Moore machine은 출력이 변하는 시간과 현재 상태가 변하는 시간이 같아 출력 신호의 타이밍 제어에 용이하다는 장점이 있다. 따라서 moore machine과 mealy machine을 구분하는 방법은 출력을 통해 결정할 수 있다. 출력이 현재 상태에만 영향을 받으면 moore machine이라는 것을 유추할 수 있고, 출력이 현재 상태와 입력 모두에 영향을 받으면 mealy machine이라는 것을 알 수 있다.

**4.**

Sequence detector는 순서 검출기이다. 순서 검출기는 입력신호가 특정한 순서로 들어오는 것을 검출하여 출력을 생성하는 회로이다. 예를 들어, 입력신호가 1001의 순서로 들어오는 것을 검출하거나, 1이 세 번 연속으로 들어오는 것을 검출하여 출력을 생성한다. Sequence detector도 상태 기계이기 때문에 mealy machine이나 moore machine을 사용해 설계할 수 있다. 1이 세 번 연속으로 들어오는 것을 검출하는 sequence detector를 예로 들어보면, 이 검출기는 111이라는 패턴을 찾고, 1을 출력한다. 먼저 moore machine을 사용하여 설계할 수 있는데, 이 때는 4개의 상태가 필요하다. 초기 상태를 A라고 할 때 1이 입력되면 다음 상태인 B로 이동하고, 0이 입력되면 A인 상태를 유지한다. B 상태에서는 1이 한 번 입력된 상태이며, 이 때 1이 입력되면 다음 상태인 C로 이동하고, 0이 입력되면 다시 처음 상태인 A로 돌아간다. C 상태에서는 1이 두 번 입력된 상태이며, 이 때 1이 입력되면 D로 이동하고, 0이 입력되면 A로 돌아간다. D는 1이 세 번 입력되었다는 것을 의미하기 때문에 1이 출력된다. D 상태에서 1이 입력되면 다시 111이 입력된 것을 의미하기 때문에 D인 상태를 유지하고 1을 출력하며 0이 입력되면 처음 상태인 A로 돌아간다. Mealy machine으로 설계하면 세 개의 상태가 필요하다. 초기 상태인 A에서 1이 입력되면 B로 상태가 변하고, 0이 입력되면 A 상태를 유지한다. B에서 1이 입력되면 C로 상태가 변하고, 0이 입력되면 A로 돌아간다. C에서 1이 입력되면 1이 연속으로 세 번 입력되었다는 것을 의미하기 때문에 1이 출력되고, C인 상태를 유지하며 0이 입력되면 다시 처음 상태인 A로 돌아간다. Moore machine으로 111을 검출하는 sequence detector를 설계한 경우에는 4개의 상태가 필요하고, mealy machine으로 sequence detector를 설계한 경우에는 3개의 상태가 필요했다. 4개의 상태는 00, 01, 10, 11이기 때문에 2개의 플립플롭이 사용된다. 하지만 만약 5개의 상태가 필요한 경우에는 두 개의 플립플롭만으로는 5개의 상태를 만들 수 없기 때문에 3개의 플립플롭이 필요하다. 세 개의 플립플롭을 사용하면 최대 8개의 상태를 갖는 sequence detector를 설계할 수 있다.

**5.**

Sequence detector를 활용하면 parity checker를 설계할 수 있다. Parity bit는 이전에 조사했던 바와 같이 데이터를 전송하는 과정에서 오류가 발생하였는지 확인하기 위해 데이터의 마지막에 추가하는 bit이다. parity bit를 포함한 8비트의 데이터가 전송된다고 가정한다. 이 때 데이터에서 1의 개수가 홀수가 되도록 parity bit를 설정한다. 예를 들어, 0010001이라는 7비트 데이터가 있으면, parity bit는 1이 되어 데이터의 개수가 홀수가 된다. 이 데이터를 수신하여 parity checker를 통해 오류가 있는지 판단하는 과정에서 sequence detector가 사용될 수 있다. Moore machine을 사용한다면 상태는 A, B 두 상태가 있다. A는 1의 개수가 짝수임을 의미하는 상태이고, B는 1의 개수가 홀수임을 의미하는 상태이다. 1의 개수가 짝수일 때는 오류가 발생한 것을 의미하기 때문에 상태가 A일 때 1이 출력된다. 상태가 A일 때 1이 입력되면 B로 변화하고, 0이 입력되면 그대로 A를 유지한다. 상태가 B일 때 1이 입력되면 A로 변화하고 0이 입력되면 그대로 B를 유지한다. 00100011이라는 8비트 데이터가 Parity checker로 입력되면 출력 z는 11000010이 되는데, 마지막 z의 값이 0이므로 데이터 전달 과정에서 오류가 발생하지 않았음을 알 수 있다. 반면 00100111이 입력되면 출력 z는 11000101이 되는데, 마지막 z의 값이 1이므로 데이터 전달 과정에서 오류가 발생하였다는 것을 확인할 수 있다.

참조:

디지털 논리설계(comer)

논리회로와 컴퓨터 구조(전중남)