

FSM은 finite-state machine의 약자로 유한한 수의 state를 가지는 컴퓨터의 이론적 모형입니다. FSM은 동시에 여러 state를 지닐 수 없고 하나의 state를 가질 수 있습니다. 즉 A와 B라는 state가 FSM에 존재하면 A와 B의 상태를 동시에 가질 수 없고, 각 시점당 A나 B state를 하나씩만 가질 수 있습니다. 현재 state에서 input(event)이 들어오게 되면 다른 state로 변화할 수 있는 데, 이러한 과정을 transition이라고 합니다.

이제 FSM의 구성과 작동을 살펴보겠습니다. FSM이 가지고 있는 bit의 개수에 따라 가질 수 있는 state의 개수가 정해지게 됩니다. 만약 n개의 bit가 존재하면 최대 2^n 개의 state를 가질 수 있게 됩니다. 예를 들자면 3개의 bit를 가지면 000, 001, 010, 011, 100, 101, 110, 111 인 state 8개를 가질 수 있습니다. 이러한 state가 될 수 있는 조건 또한 FSM의 구성이 됩니다. 이러한 FSM은 clock이 발생할 때 state transition이 일어나는데, 이때 state transition은 입력을 포함한 현재 회로의 state 값에 따라 다음 state를 정하게 됩니다.

이 때문에 FSM은 sequential circuits를 표현하는 abstract model이라고도 할 수 있는데, 여기서 sequential circuit는 현재의 input뿐만 아니라 과거의 history에 의해 output이 결정되는 회로입니다. 여기서 history는 memory에 저장된 state값이라고 할 수 있습니다. 이와 반대로 combinational circuit은 input만을 고려하여 output을 내보냅니다.

다시 돌아와서 대표적인 sequential circuit들을 소개해보겠습니다. mealy machine과 moore machine인데요, mealy machine은 현재의 input과 state에 모두 영향을 받아 output을 도출하는 장치입니다. moore 머신은 오직 현재 state만을 고려하여 output을 도출하는 장치입니다. 이러한 moore machine은 input을 받으면 mealy machine처럼 바로 output을 발생시키는 것이 아니라, input을 통해 memory의 state를 transition시키고 이후 이를 통해 output을 결정합니다. moore machine과 mealy machine의 차이를 예를 통해 알아보도록 하겠습니다. 아래 두 개의 표 모두 연속적으로 세 번 1이 input으로 들어왔을 때 output이 1이 나오는 장치를 나타낸 것입니다. 빨간색 사각형 안을 보면 무어머신은 input이 들어오고 난 다음 turn에 output이 1이 되고, mealy machine은 마지막 1이 input이 들어간 turn에 바로 output이 1이 됨을 확인할 수 있습니다.

reference는 화면과 같습니다.

여기까지 2조 발표였습니다. 감사합니다