

실험 Verilog-13: 예비보고서

전공: 컴퓨터공학

학년: 2

학번: 20171645

이름 박찬우

1. 목적

FSM과 Sequence Detector에 대해 이해한다.

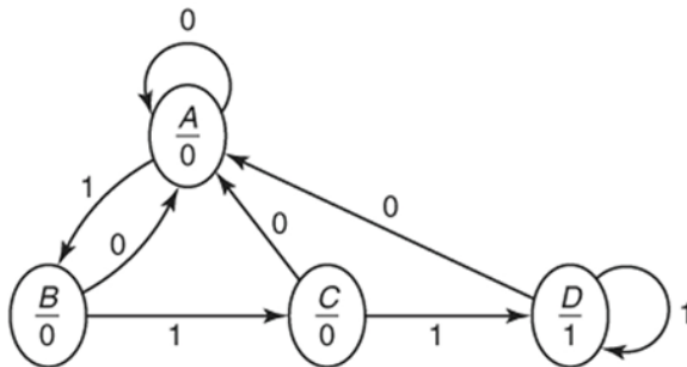
2. 요구 사항

FSM에 대해 설명하시오.

FSM(Finite State Machine)이란 유한상태기계, 즉 유한한 상태를 가지고 입력에 따라 현재 상태에서 다음 상태로 넘어감과 동시에 액션이 나타나는 기계 및 회로를 일컫는다. FSM의 종류는 크게 두 가지로 나눌 수 있는데, 각각 무어 모델(Moore model)과 밀리 모델(Mealy model)이다.

무어 모델의 경우 유한 상태 기계의 출력값은 오직 현재 상태에 따라서만 결정된다. 따라서, 무어 기계는 행위를 단순화 시킬 수 있다는 장점이 있다.

다음은 무어 모델 FSM의 간단한 예이다.



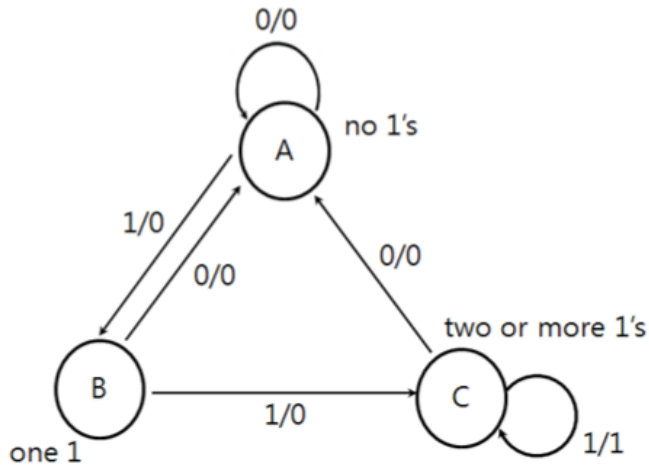
A 상태에서 0을 입력받으면 다시 A로 돌아가고, 1을 입력받으면 B상태로 넘어간다. B에서 1을 입력받으면 C로 넘어가고, 0을 입력받으면 다시 A로 돌아간다. C, D 역시 그림처럼 작동하게 된다. 이때 중요한 점은, 각 상태 아래에 결과가 정해져있다는 것이다. 즉, C에서 1을 입력받아 D로 이동할 경우 반드시 출력이 1이라는 뜻이다.

아래는 위 무어머신의 실행 예시이다.

x	0	1	1	0	1	1	1	0	0	1	0	1	1	1	1	1	0	0			
q	?	A	B	C	A	B	C	D	A	A	B	A	B	C	D	D	D	A	A	?	
z	?	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0

q의 상태가 D일때 output z가 1을 출력하는 모습을 확인할 수 있다.

밀리 모델의 경우 무어 모델과 달리 출력값이 입력 값과 현재 상태 모두에 의존한다는 차이점이 존재한다. 대신 무어모델보다 상태의 수가 줄어드는 장점이 있다.
다음은 밀리 모델의 간단한 예이다.



무어 모델과 달리 상태에 따른 결과가 정해져있지 않다. 현재 상태가 C여도 1을 입력받으면 1을 출력하지만, 0을 입력받으면 0을 출력한다. 이와 같은 결과는 아래 실행 예에서 확인할 수 있다.

x	0	1	1	0	1	1	1	0	0	1	0	1	0	1	1	1	1	0	0			
q	?	A	B	C	A	B	C	C	A	A	B	A	B	A	B	C	C	C	A	A		
z	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

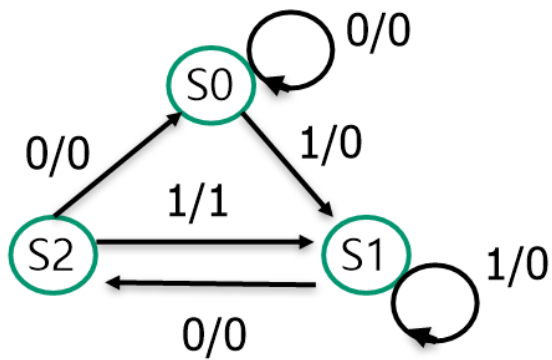
q가 C일때 입력 x의 값이 1이면 결과 z도 1이지만, x가 0이면 결과도 0인 걸 확인할 수 있다.

Sequence Detector에 대해 조사하시오.

Sequence Detector란 일련의 입력 속에서 특정 Sequence를 찾을 수 있다면 해당 Sequence에 대해 1을 출력하는 Flip-Flop으로 구성된 회로를 말한다. 이를테면, 101 이라는 sequence를 찾는 sequence detector가 있다면, 입력값 $x = 0011011001010100$ 일때 $001\underline{101}1001\underline{0101}00$ 이므로 출력 $z = 0000010000010100$ 을 출력할 것이다.

즉, 앞서 설명한 두개의 예시 모두 Sequence Detector의 예시가 될 수 있다. 무어 머신의 예시인 첫번째 예시의 경우 Sequence 111을 찾는 Sequence Detector이고, 밀리 머신의 예시인 두번째 예시의 경우 역시 111을 찾는 Sequence Detector이다. 이러한 Sequence detector는 state graph 및 state table을 구한 뒤, 플립플롭의 truth table과 비교하며 Karnaugh map을 통해 논리식을 구해 구현할 수 있다.

예를들면, 아래와 같은 State graph로부터 state table을 구할 수 있다.

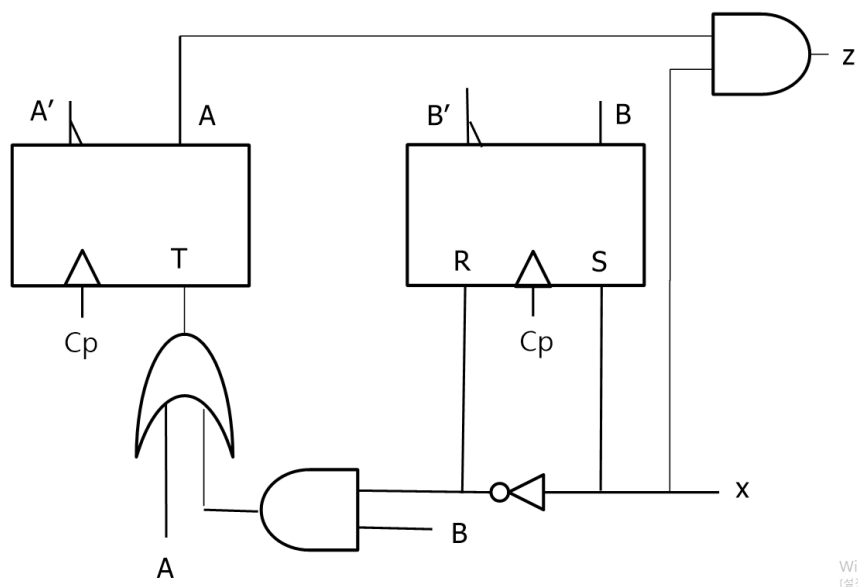


Current state	next state		Output Z	
	X=0	X=1	X=0	X=1
S0	S0	S1	0	0
S1	S2	S1	0	0
S2	S0	S1	0	0

S0, S1, S2 상태를 각각 00, 01, 10 (AB)으로 상태를 할당한다. A는 T flip flop, B는 SR flip flop으로 제작한다고 생각하면, 각 플립플롭의 truth table에 의해 다음과 같은 truth table이 성립한다.

A	B	x	A*	B*	z	TA	SB	RB
0	0	0	0	0	0	0	0	x
0	0	1	0	1	0	0	1	0
1	0	0	0	0	0	1	0	x
1	0	1	0	1	1	1	1	0
0	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	x	0

위 truth table을 바탕으로 z, TA, SB, RB의 카르노 맵을 구해 아래와 같은 회로를 구할 수 있다.



Wii
1883