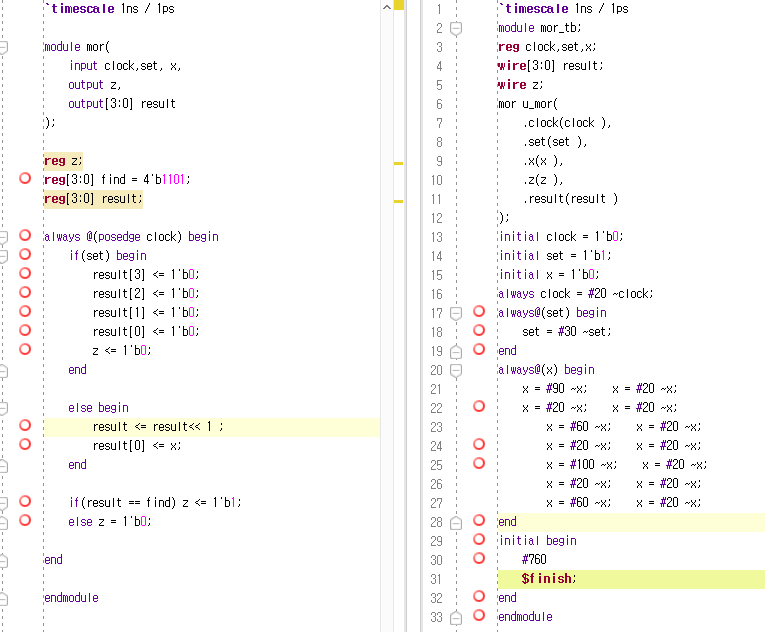
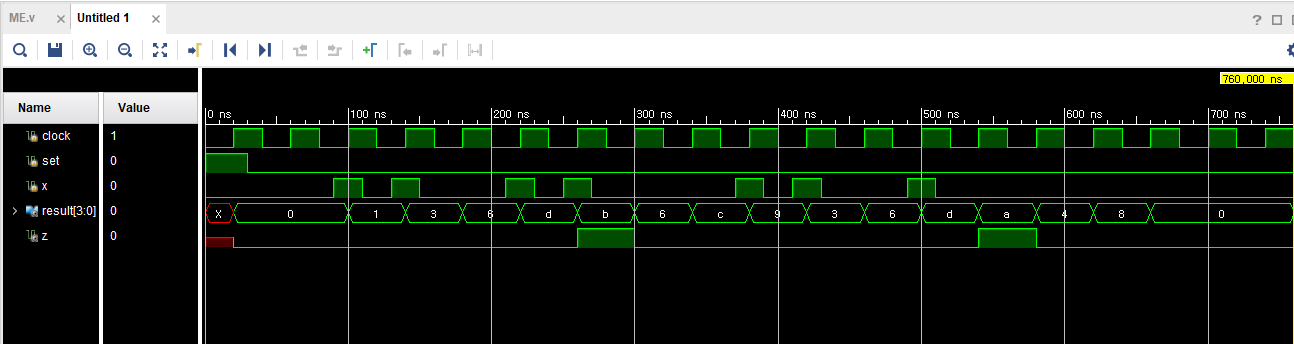
14주차 결과보고서

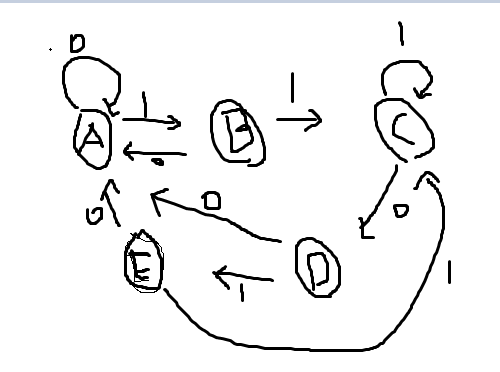
전공: 컴퓨터공학과 학년: 3학년 학번: 20191612 이름: 윤기웅

1.

무어 머신은 밀리머신과 달리 현재의 상태에서만을 고려하여 다음 상태를 정하는 것이다. 그래서 감지하고자 하는 상태를 발견 후에 결과값이 변하게 된다. 1101을 감지 해야 하는데 이는 십진수로 바꾸면 13임으로 d의 값을 갖는다. 시뮬레이션을 보게 되면 d 다음의 상태에서 결과값이 1로 변하게 된다. set은 초기화 하는 리셋 기능을 하며 상태가 진행되고 있더라도 모든 자리의 값을 0으로 바꾼다. 클락이 바뀔 때마다 입력값 x를 받아들여서 결과값 배열 result를 바꾼다. 만약 이것이 찾고자 하는 수인 1101과 같다면 결과를 바꾼다. testbench에서 클락은 주기를 20으로 항상 설정하고 x의 입력이 들어오도록 만들어준다. 이 때 1101이 들어와야 하니 시간대를 잘 설정해주면 된다.





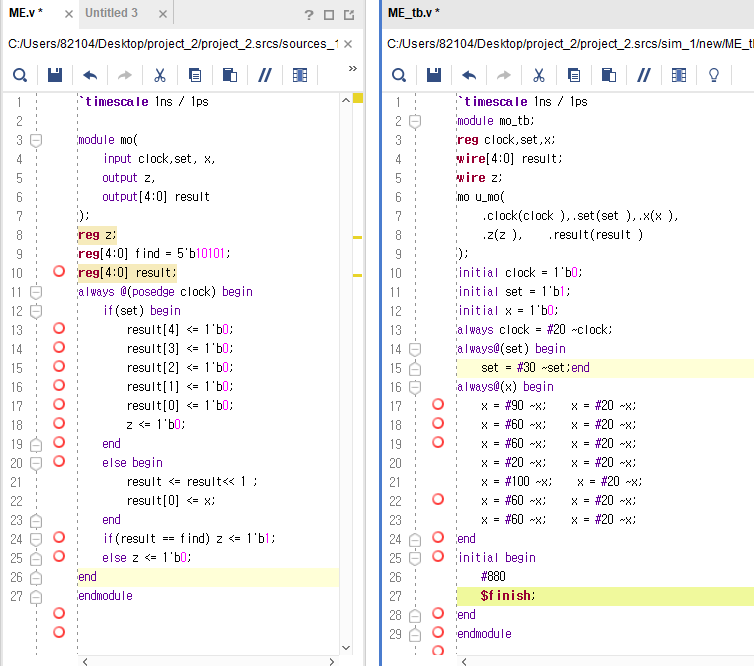


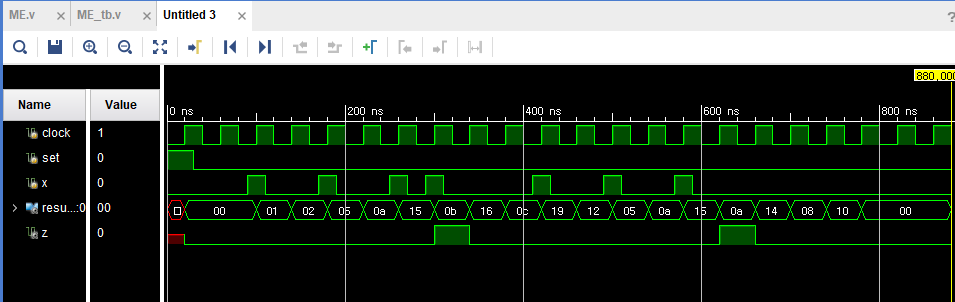
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Present State | Next State | | Output | | X=0 | X=1 | | A | A | B | 0 | | B | A | C | 0 | | C | D | C | 0 | | D | A | E | 0 | | E | A | C | 1 | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | In | 0 | 1 | 1 | 0 | 1 | 1 | | State | x | A | B | C | D | E | | Out | x | 0 | 0 | 0 | 0 | 1 | |

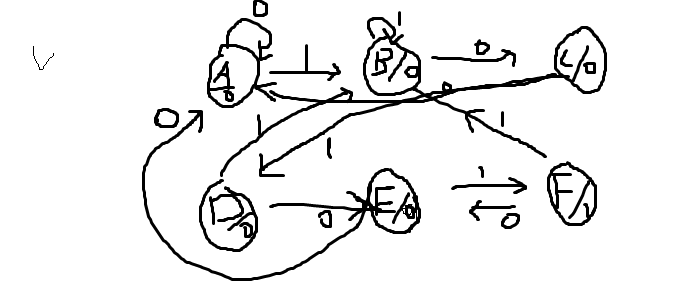
2.

1) Moore machine

무어 머신은 결과값을 z로 설정하여 찾고자 하는 10101을 발견하면 1이 되고 아닌 경우는 0의 상태를 유지한다. 10101을 find배열에 미리 입력하고 계속 입력을 받으면서 저장된 배열하고 같아지면 결과값을 1로 바꾼다. 진행되는 상태를 모두 초기화 시키기 위해서는 리셋을 만들어야 하는데 이를 set으로 만들었다. 시뮬레이션에서 여러 가지 경우가 나올 수 있도록 클락의 주기랑 맞게 입력값을 변화시켜준다. 무어 머신이기 때문에 10101을 발견 후 그 다음 클럭에서 결과값이 1로 변한다. 클럭마다 입력인 x를 배열의 첫 자리로 넣고 그 다음 자리부터는 하나씩 쉬프트 연산을 해서 값을 변경해준다. 5자리의 이진수를 감지해야 하기 때문에 상태가 6개가 있어야 하고 최종 상태인 F인 경우에만 결과값을 1로 만들어준다. 1010101의 숫자들이 차례대로 들어오게 된다면 앞의 다섯 자리의 10101에 대해서 무어 머신이 감지하고 다시 0이 들어오면 이전의 101에 합쳐서 1010을 감지한 것으로 간주해서 처음 상태인 A로 돌아가는 것이 아니라 4개의 수를 감지한 E로 돌아간다. 만약 10101101의 수가 들어오게 되면 처음 다섯 자리의 수에 대해서 무어 머신이 감지하고 결과값을 1로 바꿔준다. 그 뒤에 0이 들어오면 1010이 되는데 1이라서 1011이 된다. 그러면 앞의 101을 쓸 수 없고 그냥 제일 마지막의 1만 감지한 것으로 되어서 첫 상태의 다음 상태인 B가 된다. 다른 상태들에서도 찾고자 하는 값이 아닌 경우에 첫 상태인 A로 가는 것이 아니라 이전의 입력까지 고려한 다음에 맞는 상태로 돌아가야 한다.

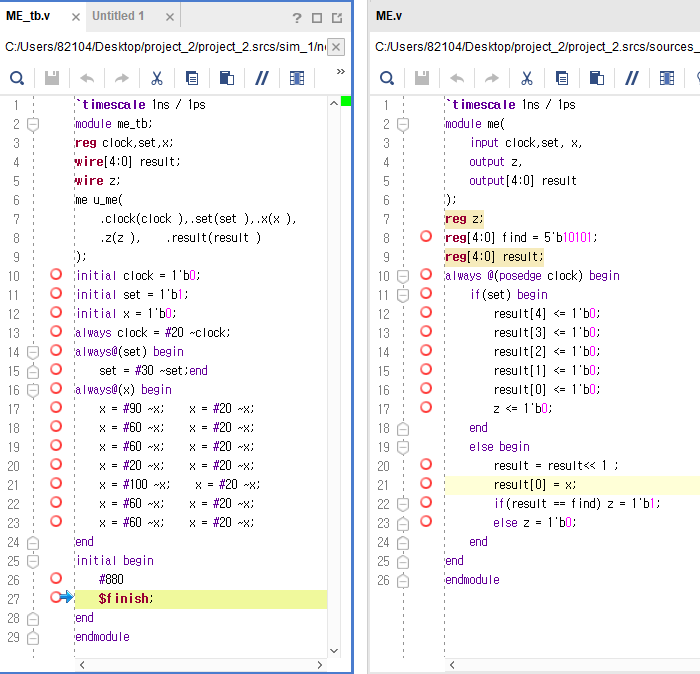


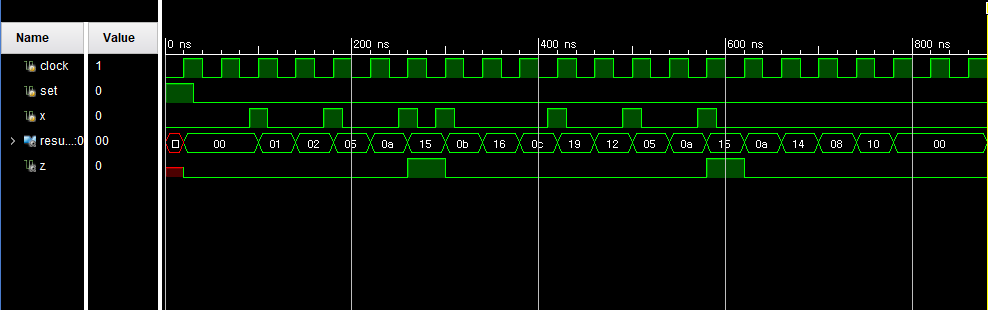


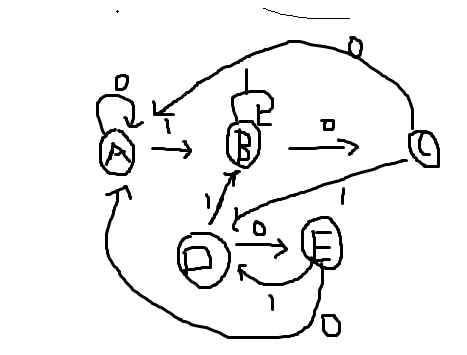


2) mealy machine

10101을 감지하는 밀리머신이기 때문에 상태는 5개가 되어야 하고 만약 찾고자하는 배열을 찾으면 바로 결과값 z가 0에서 1로 변한다. 마지막 상태인 E이전에는 각자 감지하고자 하는 1혹은 0을 찾으면 다음 상태로 간다. 하지만 마지막 E에서 1을 감지하면 A 첫 상태로 돌아가는 것이 아니라 바로 이전 상태인 D로 넘어간다. 왜냐하면 계속 들어오는 입력에 대해서만 다시 배열을 감지하는 것이 아니라 이전의 배열을 고려해야 하기 때문이다. 1010101이라는 입력이 차례대로 들어온다면 앞의 다섯 자리 수에 대해서 밀리머신이 감지하고 결과값을 1로 변경한다. 그리고 중간의 101부터 다시 고려할 때 3개의 수는 이미 감지 했음으로 D의 상태에서 시작해서 앞으로 나올 0과 1을 감지한다.







|  |
| --- |
| Mealy Machine |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | now | Next | | Out | | | X=0 | X=1 | X=0 | X=1 | | A | A | B | 0 | 0 | | B | C | B | 0 | 0 | | C | A | D | 0 | 0 | | D | E | B | 0 | 0 | | E | A | D | 0 | 1 | |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | In | 0 | 1 | 0 | 1 | 0 | 1 | 1 | | State | x | A | B | C | D | E | D | | Out | x | 0 | 0 | 0 | 0 | 1 | 0 | |
| Moore Machine |
| |  |  |  |  | | --- | --- | --- | --- | | now | Next | | Out | | X=0 | X=1 | | A | A | B | 0 | | B | C | B | 0 | | C | A | D | 0 | | D | E | B | 0 | | E | A | F | 0 | | F | E | B | 1 | |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | In | 0 | 1 | 0 | 1 | 0 | 1 | 0 | | State | x | A | B | C | D | E | F | | Out | x | 0 | 0 | 0 | 0 | 0 | 1 | |

­