14주차 결과보고서

전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1. Overlapping 방식의 Sequence Detector 1101 Moore machine 구현

우선 sequence detector는 배열을 감지하는 기능으로, 미리 정의된 일련의 입력을 인식하는 장치이다. 이때 overlapping 방식이라는 것을 설명하자면 sequence의 마지막 bit가 또 다른 새로운 sequence의 start로 사용된다는 의미이다. 또한 Moore Machine은 input과 관계없이 output이 오로지 현재 상태에 의해서만 결정이 된다. 1101을 moore machine으로 구현하기 전에 몇 가지 정하고 넘어가겠다. 예를 들어 입력으로 들어가는 sequence는

11001010110110101이라고 가정을 해보자.

또한 state를 A는 아무것도 입력되지 않은 상태, B는 1이 입력된 상태, C는 11이 입력이 된 상태, D는 110이 입력된 상태이다. E는 1101이 입력된 상태이다. 이렇게 정리한 state와 sequence를 기반으로 state table을 작성해보겠다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| state | ? | B | C | C | D | E | A | B | A | B | C | D | E | C | D | E | A |
| output | ? | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

이를 바탕으로 state table을 작성해보자면,

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

위를 바탕으로 state diagram을 구현하였다.

이 state table을 기반으로 transition table을 작성하자면,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Present State | y2y1y0 | x | Next  state | y2\*y1\*y0 |
| A | 000 | 0 | A | 000 |
| A | 000 | 1 | B | 001 |
| B | 001 | 0 | A | 000 |
| B | 001 | 1 | C | 010 |
| C | 010 | 0 | D | 011 |
| C | 010 | 1 | C | 010 |
| D | 011 | 0 | A | 000 |
| D | 011 | 1 | E | 100 |
| E | 100 | 0 | A | 000 |
| E | 100 | 1 | C | 010 |

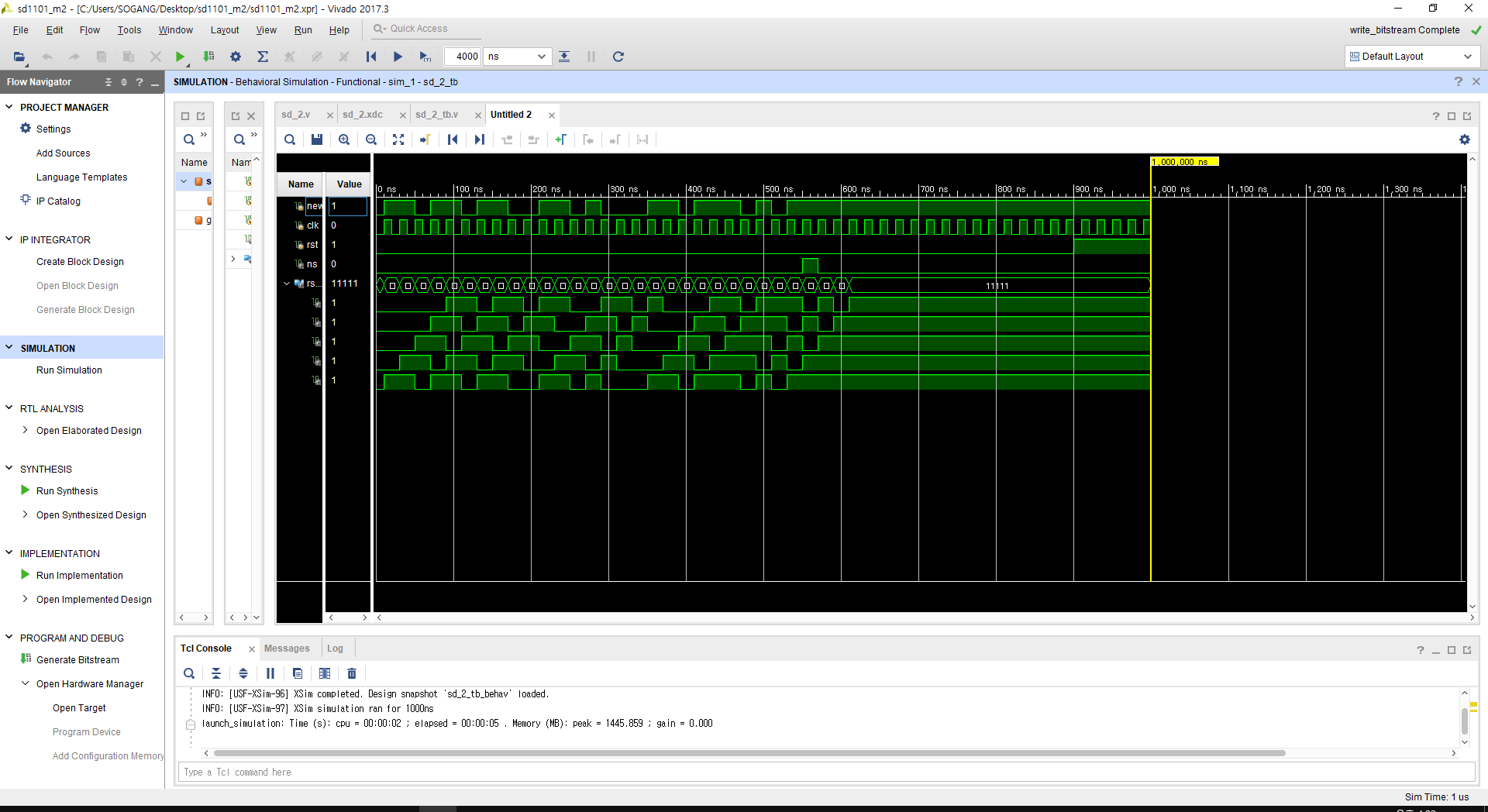
따라서, Verilog 코드를 작성해보면,

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* Simulation 결과



1101이 나올 때 ns 즉, output이 1이 되는 부분을 확인할 수 있다. 1101이 나오고 그 후 E라는 state가 update가 되기 때문에 mealy machine인 경우에는 바로 업데이트가 되지만, moore machine의 경우는 한 단계 이후에 output이 1로 업데이트 된다.

2. Overlapping 방식의 Sequence Detector 10101을 구현(mealy, moore machine 모두)

우선 mealy machine에서 sequence detector 10101을 구현하자면,

state A를 아무것도 입력되지 않은 상태, B는 1이 입력된 상태, C는 10이 입력된 상태, D는 101이 입력된 상태, E는 1010이 입력된 상태이라고 한다면,

state table을 그린다면,

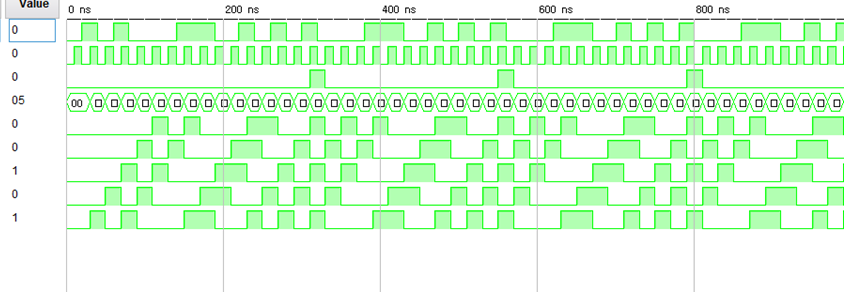
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Present State | Next State | | Output | |
|  | X= 0 | X = 1 | X= 0 | X= 1 |
| A | A | B | 0 | 0 |
| B | C | B | 0 | 0 |
| C | A | D | 0 | 0 |
| D | E | B | 0 | 0 |
| E | A | D | 0 | 1 |

위를 바탕으로 state diagram을 그린다면,

이를 바탕으로 Verilog로 10101 sequence detector를 구현해보자.

|  |  |
| --- | --- |
|  |  |

앞서 말했듯이 상태 E(1010이 입력된 상태에서 )에서 input이 1이 들어왔을 때 output인 ns를 1로 설하였다. 만약 input이 0이 들어왔을 때, if(ps == 5’b10101) 구문을 만족하지 않고, 0을 출력한다. 실습에서 진행했던 mealy machine과 동일하기 shift를 시키고, 없어진 첫 비트를 new로 채운다. 따라서, 1010을 만들기 위해서는 new를 키고, 끄고 shift하는 동작을 반복하여 구현하고, 다음 new를 킨다면, ns의 값이 1이 된다.



위에서부터 차례대로 new, clk, ns, ps[3:0]이다. simulation 결과를 잘 살펴보면, @(posedge clk)로 clk이 positive 되는 부분에 바뀌는 것을 확인할 수 있다. ns가 1인 부분을 잘 살펴보면, clk이 posedge가 되기 직전에, 01010임을 알 수 있다. 그 후 posedge가 되었을 때, shift가 되면서, new의 1이 삽입 면서 10101이 되면서 바로 ns가 1이 된다. 이것이 현재의 상태와, input의 값에 모두 영향을 받는 meal machine이다. 800ns 이후 01010 부분을 살펴보면, 이 때 new의 값은 0이므로, 10100이 된다. 이에 따라, output이 0이 된다. 같은 state였지만, ns의 다른 출력 결과를 확인할 수 있다.

다음은, 같은 sequence인 10101을 확인하는 Moore machine을 설계하겠다. 이는 위 moore machine 과 마찬가지로, 오로지 현재 상태만 output에 영향을 주는 머신이다. state를 정리하자면, 아무것도 입력을 못 받았을 때 state를 A, 1을 입력 받은 상태는 B, 10을 입력 받은 상태는 C, 101을 입력받은 상태는 D, 1010을 입력받은 상태는 E, 10101을 입력받은 상태는 F라고 하자.

이에 해당하는 state table을 작성하면,

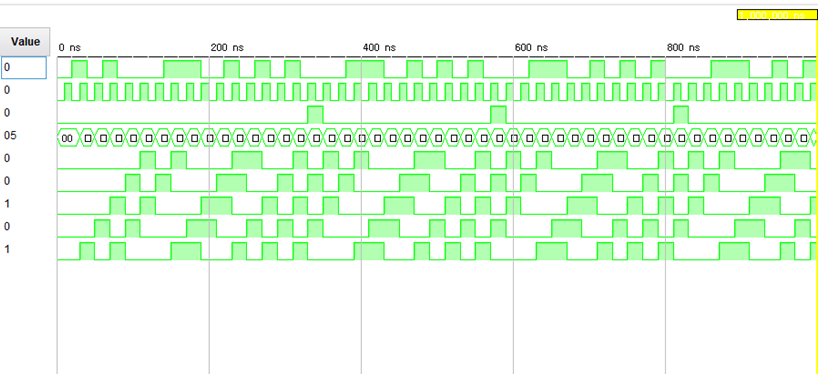
|  |  |  |  |
| --- | --- | --- | --- |
| Present State | Next State | | Output |
|  | X= 0 | X = 1 |  |
| A | A | B | 0 |
| B | C | B | 0 |
| C | A | D | 0 |
| D | E | B | 0 |
| E | A | F | 0 |
| F | E | B | 1 |

이에 대응하는 state diagram을 작성하면,

이제 이를 바탕으로, Verilog 코드를 작성해보겠다.

|  |  |
| --- | --- |
| 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명 | 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진  자동 생성된 설명 |

위 mealy machine과 코드가 거의 유사하다. 이 moore machine은 mealy machine에서 state가 5개이지만, moore machine은 state가 하나 더 늘어났다. 그 이유는 쉽게 설명하자면, mealy machine에서 마지막 state에서 1을 선택한 경우를 하나의 state로 재구성했다고 생각하면 된다. 위 mealy machine과 다르게 non-blocking 구문을 사용하여, 결과적으로 output이 하나씩 밀리게 구성했다.



위 output 즉 ps의 결과가 10101인 경우를 찾아보자, 위 화살표의 경우 10101이 주어진다. posedge 이후 new값에 상관없이, 다음에 output이 1이 된다. 이로 알 수 있듯이 이는 moore machine이고 현재의 상태만 영향을 미친다는 것을 알 수 있다.