13주차 결과보고서

전공 : 국제한국학과 학년 : 4학년 학번 : 20181202 이름 : 김수미

**1. 4-bit Shift Register의 결과 및 Simulation 과정에 대해서 설명하시오.**

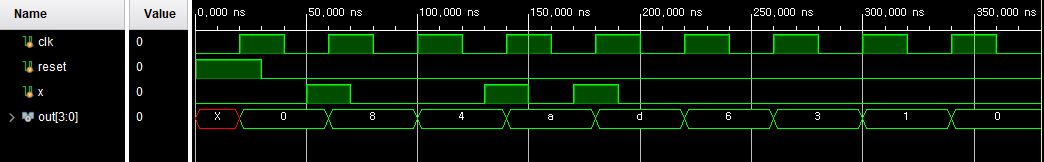
1) Truth Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Shift Register Output Table | | | | | |
| Number of Clk Transitions | Outputs | | | | |
| IN | L1(MSB) | L2 | L3 | L4(LSB) |
| 1 | Reset | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 |
| 4 | 1 | 1 | 0 | 1 | 0 |
| 5 | 1 | 1 | 1 | 0 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 |
| 7 | 0 | 0 | 0 | 1 | 1 |

2) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

3) Simulation



4) 결과 및 과정

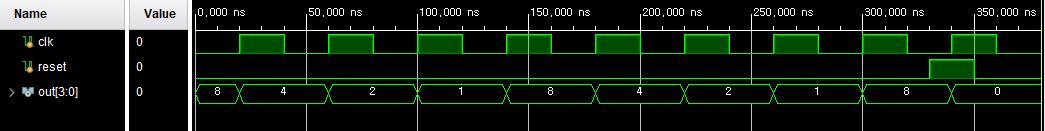
Shift Register는 clock transition이 있을 때 마다 저장된 값이 우측으로(LSB 방향으로) 한 칸씩 밀리게 되며, 새로운 input data의 값은 좌측(MSB)에 저장된다. 회로는 rising edge trigger로 설계하여 clock 값이 0에서 1이 될 때 메모리에 현재 상태의 변화가 적용된다. 따라서 매 clock transition마다 4bit의 메모리에 저장된 값들을 한 칸씩 LSB 방향으로 밀려서 저장하는 연산과 함께 새롭게 들어오는 데이터 값을 MSB에 저장하는 연산을 구현했고, simulation 수행 결과 truth table과 동일하게 잘 작동하는 것을 확인할 수 있다.

**2. 4-bit Ring counter의 결과 및 Simulation 과정에 대해서 설명하시오.**

1) Truth Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Shift Register Output Table | | | | |
| Number of Clk Transitions | Outputs | | | |
| L1(MSB) | L2 | L3 | L4(LSB) |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 |  |
| 4 | 0 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 1 | 0 |

2) Simulation



3) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

4) 결과 및 과정

Ring Counter는 clock transition이 있을 때 마다 저장된 값이 우측으로(LSB 방향으로) 한 칸씩 밀리게 된다. 새로운 input data값이 없다는 점만 제외하면 shift register와 동작 방식이 유사하다. 단 Ring Counter의 경우 LSB의 값이 다시 MSB로 올라가서 전체적으로 메모리에 저장된 4개의 bit 값이 순환하는 방식이 되도록 설계해야 한다. 회로는 rising edge trigger로 설계하여 clock 값이 0에서 1이 될 때 메모리에 현재 상태의 변화가 적용된다. 따라서 매 clock transition마다 4bit의 메모리에 저장된 값들을 한 칸씩 LSB 방향으로 밀려서 저장하되 LSB의 값이 다시 MSB에 저장되도록 연산을 구현했고, simulation 수행 결과 truth table과 동일하게 잘 작동하는 것을 확인할 수 있다.

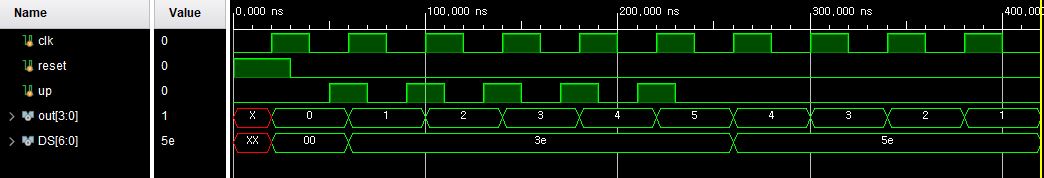
**3. 4-bit Up/Down counter 의 결과 및 Simulation 과정에 대해서 설명하시오.**

1) Truth Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UP Counter Output Table | | | | | |
| Number of Clk Transitions | Outputs | | | | |
| L1(MSB) | L2 | L3 | L4(LSB) | DISPLAY |
| 1 | 0 | 0 | 0 | 1 | U |
| 2 | 0 | 0 | 1 | 0 | U |
| 3 | 0 | 0 | 1 | 1 | U |
| 4 | 0 | 1 | 0 | 0 | U |
| 5 | 0 | 1 | 0 | 1 | U |
| 6 | 0 | 1 | 1 | 0 | U |
| 7 | 0 | 1 | 1 | 1 | U |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DOWN Counter Output Table | | | | | |
| Number of Clk Transitions | Outputs | | | | |
| L1(MSB) | L2 | L3 | L4(LSB) | DISPLAY |
| 1 | 0 | 1 | 1 | 1 | d |
| 2 | 0 | 1 | 1 | 0 | d |
| 3 | 0 | 1 | 0 | 1 | d |
| 4 | 0 | 1 | 0 | 0 | d |
| 5 | 0 | 0 | 1 | 1 | d |
| 6 | 0 | 0 | 1 | 0 | d |
| 7 | 0 | 0 | 0 | 1 | d |

2) Simulation

3) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

4) 결과 및 과정

UP/DOWN Counter는 clock transition이 있을 때 마다 UP의 input이 들어오면 저장된 값이 1씩 증가하고, DOWN의 input이 들어오면 저장된 값이 1씩 감소한다. 4bit counter이기 때문에 나타낼 수 있는 최댓값은 15이고, 최솟값은 0이다. UP의 input이 들어오는 상태에서는 7 segment display에 알파벳 ‘U’가 display 되고 DOWN의 input이 들어오는 상태에서는 알파벳 ‘d’가 display 되도록 설계했다. 회로는 rising edge trigger로 설계하여 clock 값이 0에서 1이 될 때 메모리에 현재 상태의 변화가 적용된다. Simulation 수행 결과 truth table과 동일하게 잘 작동하는 것을 확인할 수 있다.

**4. 결과 검토 및 논의 사항**

7 Segment Display를 사용할 때는 constraint 파일을 작성할 때 7개의 변수를 7 segment display에 할당해 주는 것 뿐만 아니라, dp와 digit 변수를 design source 파일에서 적절하게 선언한 뒤 할당해 주는 것을 잊어버리지 말자. 7 segment display를 제대로 할당 했더라도 dp와 digit 변수를 할당하지 않으면 display가 작동하지 않는다.

**5. 추가 이론 조사 및 작성**

1) Verilog에서 +1 연산 사용하기

지난주 실습에서 decade counter를 구현할 때 4bit의 숫자에 +1을 계산하는 연산을 수행하기 위해 각각의 bit에 대한 sum과 carryout을 연산하는 등 복잡하게 식을 작성했었다. 하지만 그럴 필요 없이, 4bit의 변수에 +1 을 해주기만 하면 동일한 결과를 얻을 수 있다는 것을 이번 실습을 통해 알게 되었다. 정리하면 아래와 같다.

|  |  |
| --- | --- |
| Bit 별로 계산하는 경우 | 전체에 +1 을 해주는 경우 |
| if (out[0]==1’b1) begin  carryout <= 1’b1;  out[0] <= 1’b0;  end  else begin …. | out = out+1; |