13주차 결과보고서

전공 : 컴퓨터공학과 학년 : 3학년 학번 : 20211558 이름 : 윤준서

**1. 4-bit Shift Register**

4-bit Shift Register의 verilog source는 다음과 같다.

텍스트, 영수증, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷, 영수증이(가) 표시된 사진

자동 생성된 설명

이를 통해 구현한 simulation의 결과는 다음과 같다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

4-bit Shift Register는 Clock 신호의 Trigger가 발동할 때마다 각 비트에 저장된 데이터를 한 칸 씩 옮기는 역할을 한다. 초기 입력 값이 1000일 때, 다음 입력 값에 따라 out[0]의 값이 결정된다. 그리고 나머지 비트는 이전 비트의 데이터를 옮겨 받는다. Clock Trigger에 따른 입력 값이 순서대로 1, 1, 0 이라고 할 때, 출력 값은 1000, 0001, 0011, 0110, ... 를 순서대로 가지게 된다. 즉 하나의 데이터 값이 모든 비트를 거치면 그 데이터는 사라지게 된다. 그리고 새로운 입력 값이 초기 비트에 다시 저장된다.

이를 통해 정리한 4-bit Shift Register의 입력 값에 따른 출력 값의 표는 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Clock Trigger 횟수 | input | output | | | |
| x | out[0] | out[1] | out[2] | out[3] |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 1 | 1 | 0 | 1 | 1 |

**2. 4-bit Ring Counter**

4-bit Ring Counter의 verilog source는 다음과 같다.

텍스트, 영수증, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷, 영수증이(가) 표시된 사진

자동 생성된 설명

이를 통해 구현한 simulation의 결과는 다음과 같다.

스크린샷, 텍스트, 사각형이(가) 표시된 사진

자동 생성된 설명

4-bit Ring Counter는 Clock 신호의 Trigger가 발동할 때마다 각 비트에 저장된 데이터를 한 칸 씩 옮기는 역할을 한다. 초기 입력 값이 1000일 때, Clock Trigger가 발동하면서 데이터 값이 0100, 0010, 0001으로 바뀌게 된다. 그리고 다시 Trigger가 발동하면 out[0]의 데이터가 다시 out[3]으로 이동한다. 즉 Ring Counter는 Shift Register와 달리 데이터가 없어지지 않고 다시 처음 비트에 전달하여 데이터 이동을 원형으로 반복한다.

이를 통해 정리한 4-bit Ring Counter의 입력 값에 따른 출력 값의 표는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clock Trigger 횟수 | output | | | |
| out[3] | out[2] | out[1] | out[0] |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 |
| 4 | 1 | 0 | 0 | 0 |

**3. 4-bit UP/DOWN Counter**

4-bit UP/DOWN Counter의 verilog source는 다음과 같다.

텍스트, 영수증, 패턴이(가) 표시된 사진

자동 생성된 설명텍스트, 영수증, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

이를 통해 구현한 simulation의 결과는 다음과 같다.

스크린샷, 텍스트, 디스플레이, 다채로움이(가) 표시된 사진

자동 생성된 설명

4-bit UP/DOWN Counter는 입력 값에 따라 출력 값의 크기가 증가하거나 감소한다. 해당 코드에서는 입력 변수 x의 값이 1일 때 증가를, 0일 때 감소의 의미를 가진다. Clock Trigger가 작동할 때마다 출력 값이 변하는데, 4-bit이므로 출력 값의 최대는 f이므로 f인 상태에서 다시 증가 입력을 받으면 0이 된다. 반대로 0인 상태에서 감소 입력을 받으면 f가 된다.

이를 통해 정리한 4-bit UP/DOWN Counter의 입력 값에 따른 출력 값의 표는 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Clock Trigger 횟수 | input | output | | | | Seg |
| x | out[3] | out[2] | out[1] | out[0] |
| 0 | 1 | 1 | 0 | 0 | 0 | u |
| 1 | 1 | 1 | 0 | 0 | 1 | u |
| 2 | 1 | 1 | 0 | 1 | 0 | u |
| 3 | 1 | 1 | 0 | 1 | 1 | u |
| 4 | 1 | 1 | 1 | 0 | 0 | u |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Clock Trigger 횟수 | input | output | | | | Seg |
| x | out[3] | out[2] | out[1] | out[0] |
| 0 | 0 | 1 | 0 | 0 | 0 | d |
| 1 | 0 | 0 | 1 | 1 | 1 | d |
| 2 | 0 | 0 | 1 | 1 | 0 | d |
| 3 | 0 | 0 | 1 | 0 | 1 | d |
| 4 | 0 | 0 | 1 | 0 | 0 | d |

**4. 결과 검토 및 논의 사항**

해당 실습에서는 4-bit UP/DOWN Counter의 출력 값의 증감을 표현하기 위해 Segment를 이용해 증가 입력을 받으면 u, 감소 입력을 받으면 d를 띄우도록 설정했다.

**u를 표현하기 위한 LED의 값은 다음과 같다.**

a = 0, b = 0, c = 1, d = 1, e = 1, f = 0, g = 0

**d를 표현하기 위한 LED의 값은 다음과 같다.**

a = 0, b = 1, c = 1, d = 1, e = 1, f= 0, g = 0

verilog에서 Segment를 작동시키기 위해선 7개의 LED 변수 말고도 Digit 자리를 나타내는 변수가 필요하다. v 코드에서 해당 변수를 선언하고, 값을 항상 1로 할당한다. 그리고 xdc 파일에서 기타 변수와 같은 방식으로 set\_property를 한다.

**5. 추가 이론**

컴퓨터의 CPU 사용량이 높은 경우 verilog에서 Run Synthesis가 제대로 작동하지 않음을 확인했다. 코드에 오류가 없더라도 메모리 크기가 크면 CPU의 한계에 의해 검사가 제대로 이루어지지 않는다.