13주차 결과보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20191599 이름: 송경호

**1.**

4-bit Shift Register는 데이터의 저장과 이동에 사용되는 회로로 Flip-Flop을 연쇄적으로 이어 붙여 구성한다. 이에 따라 하나의 Flip-Flop의 출력이 다음 Flip-Flop의 입력으로 연결된 구조를 띄고 있다.

도표, 라인, 평면도, 기술 도면이(가) 표시된 사진

자동 생성된 설명

Shift Register는 위와 같은 형태를 띄며, 이때 CLK, 즉 Clock 비트가 각 register에 연결되어 있는 형태이다. 따라서 CLK의 상승 혹은 하강 edge마다 Flip-Flop이 작동하여 데이터가 우측으로 밀려난다.

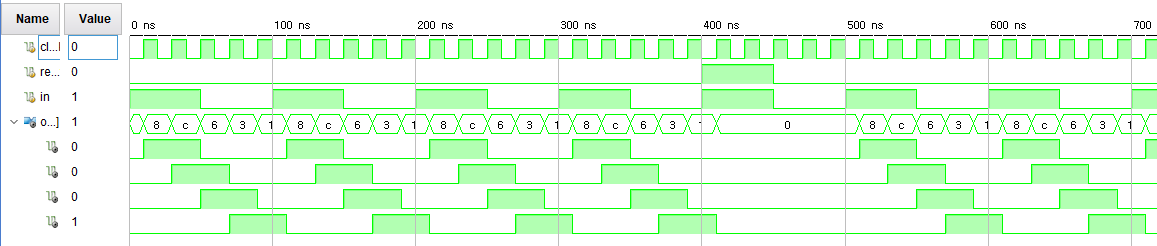
아래는 이를 Verilog로 구현한 디자인과 테스트 밴치코드이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

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

자동 생성된 설명

이전 실습에서 그랬듯 이번 실습 역시 배열과 모듈함수를 이용하여 보다 직관적으로 구현했다. 4-bit Shift Register는 input, clock, reset 비트를 입력으로 받으며 이에 대해 4 bit를 출력한다. 출력값은 최초에 0으로 초기화하며, 매 clock의 positive edge마다 출력이 한단계 씩 밀리게 된다. 따라서 4 bit shift register는 input이 한번 들어왔다는 가정하에 1000 > 0100 > 0010 > 0001 순으로 한단계 씩 밀리게 된다. 테스트 밴치코드는 이를 쉽게 확인하기 위해 매 10ns 마다 clock 비트를 변화시켰으며 결론적으로 20ns마다 pos edge가 발생한다. input 비트는 50ns마다 변화를 주고 reset 비트는 400ns부터 450ns까지만 set 시켰다.



다음은 해당 Verilog 코드의 simulation 결과이다. input 비트가 50ns 동안 set되어 있고, 그사이에 pos edge가 두 번 발생하는 것을 볼 수 있다. 때문에 첫번째 pos edge인 10ns에서 1000의 출력이 발생하고 두번째 pos edge인 30ns에서 해당 비트가 이동하여 0100의 출력이 발생한다. 동시에 이때 input 비트가 set 되어 있어 결과적으로 1100의 출력이 발생한다. 이후의 pos edge인 50과 70ns에서 input bit가 다시 0이 되기 때문에 기존의 출력들의 이동들만 발생하여 0110, 0011 순으로 이동하고 마지막 90ns에서 0001이 된다. 해당 반복이 매 100ns마다 반복되고 있으며, 400ns에서는 input bit가 set 되어 있는 동안 reset bit도 set 되기 때문에 아무런 출력도 발생하지 않는 것을 알 수 있다.

강의 자료에서 제공하는 Shift Register Output Table은 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Shift Register Output Table** | | | | | |
| **Number Clock Transitions** | **output** | | | | |
| ↑ | IN | L1 | L2 | L3 | L4 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | 1 | 0 | 0 |
| 3 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 |
| 5 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 0 | 0 | 1 | 1 |
| 7 | 0 | 0 | 0 | 0 | 1 |

**2.**

4-bit Ring Counter는 4-bit Shift Register와 유사한 방식으로 동작한다. 동일하게 4개 Flip-Flop으로 이루어져 있으며, 하나의 출력이 다음의 입력으로 들어가는 형태를 취한다. 또한 Clock bit가 각 각의 Register에 연결되어 있어 하강/상승 엣지에서 데이터의 이동이 발생한다. 이때 Ring Counter의 차이점은 마지막 Flip-Flop이 단순히 출력에 그치지 않고 다시 첫번째 Flip-Flop의 입력으로 사용되며 순환이 발생한다는 점이다.

**도표, 라인, 기술 도면, 평면도이(가) 표시된 사진

자동 생성된 설명**

해당 회로를 Verilog 코드로 구현한 결과는 다음과 같다.

텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

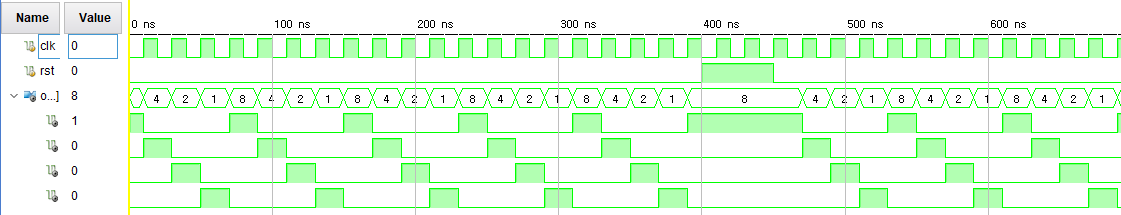
자동 생성된 설명텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

Shift Register와 동일하게 배열과 모듈을 이용하여 구현했다. 또한 역시 clock비트와 reset비트를 입력으로 받고 4 bit를 출력한다. 이때 input 비트는 따로 존재하지 않는 것이 특징이다.

최초 출력은 1000이며 매 posedge마다 4번 비트에 1번 비트가, 3번 비트에 4번 비트가, 2번 비트에 3버 비트가, 1번 비트에 2번 비트가 입력되어 결과적으로 4개의 연쇄된 shift register 형태를 구현했다. 또한 reset bit가 set된 경우 출력을 1000으로 다시 초기화 하는데, 이때 ring counter는 따로 input 비트가 존재하지 않기 때문에 해당 출력을 0000으로 초기화하면 아무런 출력도 발생하지 않기 때문에 주의해야 한다.

테스트 밴치는 동일하게 10ns마다 clock bit를 변화시켜, 결론적으로 20ns마다 pos edge가 발생하도록 설정하였으며 reset bit는 400ns에서 1로 50ns동안 set 되어 있다. 다음은 해당 코드의 시뮬레이션 결과이다.



첫번째 pos edge인 10ns 이전에도 출력은 1000으로 초기화 되어 있다. 이후 reset bit가 0인 경우, 10, 30, 50, 70ns 즉 매 pose edge마다 1000 > 0100 > 0010 > 0001 순으로 출력 비트가 한 칸씩 이동하는 것을 알 수 있다. 또한 마지막 flip-flop의 출력이 다시 첫 flip-flop으로 입력되는 만큼 0001이후에 바로 1000 출력이 발생한다.

이후 400ns 시점에서 50ns 동안 reset bit가 set 되어 있어 이 시점에는 pos edge가 발생해도 출력이 1000으로 고정되어 있는 것을 알 수 있다.

강의자료에서 제공하는 Ring Counter Output Table의 결과는 아래와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ring Counter Output Table** | | | | |
| **Number Clock Transitions** | **Outputs** | | | |
| ↑ | L1 | L2 | L3 | L4 |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 1 | 0 |

**3.**

Up/Down Counter는 이전 실습에서 다룬 counter의 응용회로로 한 방향, 즉 단순 증가 혹은 단순 감소만이 가능했던 counter와 달리 증가와 감소를 모두 수행할 수 있는 counter이다. 또한 Up/Down Counter의 특징은 아무 지점에서 Up과 Down을 지정할 수 있다는 점이다.

아래는 Up/Down Counter의 Verilog 코드이다.

**텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

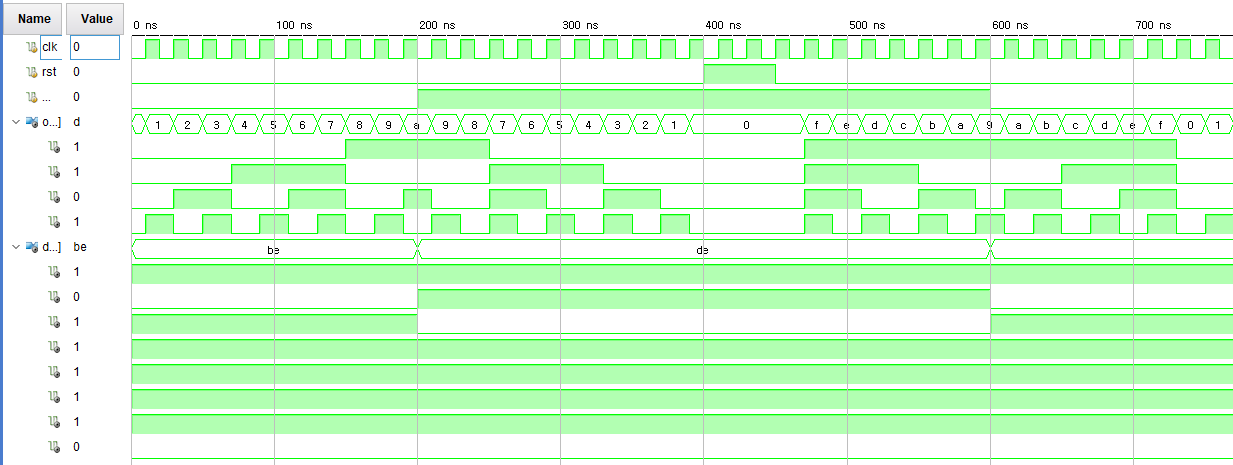
자동 생성된 설명텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명**

Up/Down counter는 clock, reset, mode 비트를 입력으로 받는다. 또한 4 bit 출력과 현재의 모드를 표시하기 위한 7 segement display용 출력을 포함한다.

초기 출력값은 0000이며, 매 posedge마다 mode bit가 0일 경우 Up counter로, 1일 경우 Down counter로 작용한다. 또한 reset bit가 set되어 있는 경우 출력은 다시 0000으로 초기화 된다. 마지막으로 mode bit에 따라 7 segement의 F와 G segment를 다르게 출력하여 mode가 0인 경우 Up의 U를 mode가 1인 경우 down의 d를 디스플레이 한다.

테스트 밴치 코드에서는, clock bit를 이전과 동일하게 10ns마다 전환 했으며 mode bit는 200ns~600ns까지 1로 나머지는 0으로 작동하도록 설정하였고 reset bit는 동일하게 400 ~ 450ns에 set 되도록 설정했다. 아래는 이에 대한 시뮬레이션 결과이다.



200ns까지는 mode bit가 0으로 설정되어 있다. 따라서 Up Counter로 작용하며 10, 30, 50 …의 pos edge 마다 1씩 값이 증가하여 0001 > 0010 > 0011 순으로 증가하여 마지막 190ns에서 1010의 값을 가진다. 이후 200ns부터 mode bit가 1이 되어 Down Counter로 작용해 다시 1010 > 1001 > 0111 … 순으로 감소하여 390ns에서 다시 0000이 되는 것을 알 수 있다. 7 Segement bit를 보면 mode에 따라 F와 G segement의 상태가 바뀌는 것을 알 수 있다. 이 후 400ns부터 450ns까지 reset bit가 set되어 pos edge에도 출력의 변화가 없다. 이후 470ns 부터 다시 Down Counter가 작용하는데, reset bit로 인해 출력이 0000으로 초기화 되었었기 때문에 Down을 수행하여 1111이 되는 것을 알 수 있다.

**4.**

본 13주차 실험에서는 flip-flop을 응용하여 4 bit shift register, 4 bit ring counter를 구현했고, Up Down Counter를 구현했다.

또한 모든 회로가 clock bit의 상승 엣지에서 특정 변화를 수행하므로, 이를 디자인 코드에 구현했으며 보다 직관적인 코드를 위해 이전 실습과 마찬가지로 배열을 이용해 출력했다. 또한 해당 코드를 FPGA를 통해 동작을 확인했다.

실습 도중 문제는 마지막 Up / Down Counter 실습에서 발생했는데, 모든 동작이 올바르게 발생함에도 7 Segment display가 나타나지 않는다는 문제가 있었다. 해당 문제는 dp bit와 digit bit를 설정하지 않았기 때문에 발생한 문제로, 단순히 u와 d만을 출력해야 했기 때문에 해당 bit들을 고려하지 않았는데 이로 인해 7 Segment display가 아예 출력되지 않는 문제가 발생했다.

또한 강의자료에서 제공한 table들과 실제 simulation 결과를 비교하며 모두 동일한 것을 확인했고 올바르게 코드가 작성됐다는 사실을 크로스 체크할 수 있었다.

**5.**

**5-1) Cascading Counter**

Counter 회로의 경우 Ringr과 Up/Down 외에도 Cascading Counter가 존재한다. Cascading은 순차적으로 전달됨을 의미하며 가장 대표적인 예로는 웹의 스타일링을 위한 CSS의 C가 Cascading의 약어의다.

Cascading은 Counter에서 최대 비트 + 1 수행 시 다시 0으로 떨어지는 것을 의미하는데 이는 rollover라고 불리며, rollover를 이용하여 multi bit counter를 만들 수 있다.

예를 들어, BCD counter가 여러 개 있는 경우 하나의 BCD Counter에서 rollover가 발생하는 신호를 다음 BCD Counter의 Clock으로 연결하게 되면 n 자리의 BCD Counter를 구성할 수 있다. 이러한 기법은 비동기식 counter의 기법이다.

**5-2) Synchronous, Asynchronous**

Counter에는 동기식과 비동기식 카운터가 존재한다. 동기식 카운터는 하나의 clock 입력을 카운터를 구성하는 모든 flipflop이 공유하는 방식이며 반대로 비동기식 카운터는 최하위 플립 플롭만이 clock입력을 받고 해당 플립 플롭이 상위 플립 플롭의 출력을 결정하는 방식이다. 이번 실험에서는 모두 비동기식 방식으로 카운터를 구현하였으며 2 bit counter의 동기식 구현은 아래와 같다.

도표, 평면도, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

동기식 카운터의 경우 gate delay가 발생하지 않아 비동기식 카운터에 비해 값의 변경이 빠르다는 특징이 있다.

**5-3) Counter의 활용**

Counter가 일상생활에서 사용되는 가장 대표적인 예시는 디지털 시계이다. 이전에 배운 7 segment display와 BCD-to-7 Segment decoder, 그리고 counter를 이용하여 디지털 시계를 구성한다. counter를 통해 발생한 값을 decoder를 통해 display에 전달하는 방식으로 이루어진다.

텍스트, 도표, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명

이외에도 주파수 측정기, 주소 생성기 등 주기적으로 동작을 제어해야 하는 다양한 분야에서 counter가 활발하게 사용된다.