11주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

RS Flip-Flop은 NAND를 이용하는 것과 NOR를 이용하는 것 두 가지가 존재하는데, 우선 NAND로 구현하는 RS Flip-Flop의 회로도는 아래와 같다.

도표, 라인, 평면도, 스크린샷이(가) 표시된 사진

자동 생성된 설명

이를 이용하여 Verilog로 나타낸 코드는 아래와 같이 구현할 수 있다. 각각 clk는 Clock Pulse, R, S의 입력을 나타내고, Q1과 Q2는 각각 Q와 ~Q를 의미한다.

텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

이를 Simulation할 때 사용한 testbench는 다음과 같다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

해당 코드는 강의 자료에 주어진 table에서 필요한 입력의 순서대로 R과 S가 들어가도록 배치하였고, clk 즉 Clock Pulse가 10ns의 주기로 바뀌도록 하였다. Simulation의 결과는 아래와 같다.

스크린샷, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

해당 Simulation에서 CLK가 0일 때는 입력에 영향을 받지 않으므로 뒷부분만 확인하면 되는데, 이를 table로 나타내면 아래와 같이 나타낼 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| In R | In S | Q | ~Q |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

여기서 R=1, S=1인 출력은 RS Flip-Flop에서 허용되지 않는 입력의 부분이므로 결과 해석에 중요하지 않고 나머지 부분을 살펴보면 처음 R=0, S=1이 주어졌을 때 Q=1로 나온다. 그 후 R=0, S=0일 때는 값이 그대로 유지되고, R=1, S=0이 주어졌을 때 Q=0으로 변환된다. 그 후 R=0, S=0일 때 유지되고, R=1, S=0이 한 번 더 주어져 값이 변하지 않는다. 이를 통해 RS Flip-Flop이 제대로 동작하는 점을 확인할 수 있다.

다음으로 NOR를 이용하여 구현하는 Flip-Flop은 아래와 같은 회로도로 나타낼 수 있다.

도표, 라인, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

이를 이용하여 Verilog 코드로 나타내면 아래와 같다. 변수의 이름이나 활용은 NAND를 활용하였을 때와 똑같이 사용하였다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Simulation에 사용한 testbench 코드는 아래와 같다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

이를 사용한 Simulation 결과는 아래와 같다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

결과를 확인하면 NAND를 활용하였을 때와 같은 결과가 나오는 것을 확인할 수 있고 이를 table로 나타내면 아래와 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| In R | In S | Q | ~Q |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

마지막으로 State Table로 나타내면 아래와 같다.

|  |  |  |
| --- | --- | --- |
| R | S | Q\* |
| 0 | 0 | Q |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | Not alloweds |

FPGA 보드에서 실습을 진행할 때는 아래와 같은 constraints 코드를 사용하였다.

텍스트, 스크린샷, 폰트, 바이올렛색이(가) 표시된 사진

자동 생성된 설명

NAND를 활용한 RS Flip-Flop의 경우만 보드에 올려 실습하였고 위의 테이블과 동일하게 출력되는 점을 볼 수 있었다.

................

**2.**

.......................

D Flip-Flop은 아래와 같은 회로도로 나타낼 수 있다. 해당 회로도에서 E가 Clock Pulse인 경우로 생각할 수 있다.

도표, 라인, 스케치, 폰트이(가) 표시된 사진

자동 생성된 설명

이를 이용하여 Verilog 코드로 나타내면 아래와 같다.

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

여기서 D가 입력, clk는 Clock Pulse를 나타내고 Q1과 Q2가 각각 Q와 ~Q를 의미한다. Simulation 시 사용한 testbench 코드는 아래와 같다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

이는 강의 자료에 있는 Table의 입력 순서에 맞춰 입력이 들어오도록 하였고, Clock Pulse인 clk는 10ns 간격으로 값이 바뀌도록 하였다. 이를 이용하여 Simulation한 결과는 아래와 같다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

여기서 CLK가 1로 들어왔을 때만 확인해보면, D가 1이 들어올 때 Q1, 즉 Q가 1이 되는 것을 확인할 수 있고, D에 0이 들어올 때 Q1, 즉 Q가 0으로 바뀌는 것을 확인할 수 있다.이를 Table로 나타내면 아래와 같다.

|  |  |  |
| --- | --- | --- |
| In D | Q | ~Q |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 0 |

다음으로 State Table로 나타내면 아래와 같다.

|  |  |
| --- | --- |
| D | Q\* |
| 0 | 0 |
| 1 | 1 |

FPGA 보드에 올려서 실습할 때는 아래와 같은 constraints 코드를 사용하였다.

텍스트, 스크린샷, 폰트, 바이올렛색이(가) 표시된 사진

자동 생성된 설명

보드에서도 위와 같이 동작하는 것을 확인할 수 있었다.

.........................

**3.**

.......................

본 실험에서는 RS Flip-Flop과 D Flip-Flop에 대해 알아보고, Verilog를 이용하여 구현하는 실습을 진행하였다. Flip-Flop부터는 실습 시에 Clock이 들어가는 방식이 사용되는 점을 볼 수 있고, 이번에는 원하는 입력 순서대로 고정하기 위해서 testbench에서 입력을 지정하는 방식을 새롭게 써볼 수 있었다. 또한 Flip-Flop에서는 다른 쪽의 출력이 입력으로도 쓰이므로 xdc 파일에서 작성시에 ALLOW\_COMBINATORIAL\_LOOPS에 대한 처리도 사용하였다.

Flip-Flop에 대해서 찾아보면 원래는 Edge-trigger 방식을 사용하는 점을 확인할 수 있는데, 이번 실습에서는 Clock을 그러한 방식으로 구현한 것이 아닌 1로 활성화되었을 때로 구현하였다. 이에 대한 부분은 posedge 등의 처리를 통해서 Rising edge나 Falling edge에서만 동작하도록 구현할 수 있는 점에 대해 알 수 있었고 이를 이용하면 더 정확히 구현할 수 있을 것이라 생각한다.

.........................

**4.**

.......................

Flip-Flop을 묶어서 나타낼 수 있는 회로 중에서 레지스터에 대해서 알아보았다. 레지스터는 D Flip-Flop을 묶어서 나타내는 회로로 외부로부터 들어오는 데이터를 저장하거나 이동하는 목적으로 사용된다. Flip-Flop이 1 bit를 저장하는 회로라고 했으므로 이는 여러 개 연결하는 것에 따라 bit를 저장하는 수가 달라지게 된다. 저장의 기능 시에는 Clock이 들어오기 전, 즉 Rising edge던 Falling edge던 정해둔 edge 전까지 입력된 데이터를 저장한다. 데이터의 이동은 직렬과 병렬을 각각 입력과 출력에 활용하여 총 4가지의 경우가 존재한다. 여기서 모든 bit를 한 bit씩 옮기는 레지스터를 시프트 레지스터라고 한다. 이중에서 직렬로 입력을 받아 직렬로 출력을 하는 경우에 대해서 알아보면 아래와 같이 블록으로 나타낼 수 있다.

라인, 도표, 폰트, 텍스트이(가) 표시된 사진

자동 생성된 설명

각각의 경우 값이 어떤 결과가 나오는지는 아래의 그림으로 확인할 수 있다.

텍스트, 도표, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

여기서는 Falling edge 시 Clock이 활성화되도록 구현되어 있다. 이외에는 병렬을 활용한 경우들이 있고 직렬로 연결 시에는 속도가 느리지만 회로가 간단하고, 병렬은 속도가 빠르다는 장점이 있다.

.........................

출처

<https://www.geeksforgeeks.org/rs-flip-flop/>

<https://blog.naver.com/asd7979/30110973727>

<https://m.blog.naver.com/leeyunghuk1/220986979536>