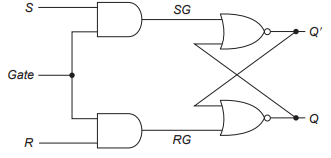
11주차 결과보고서

전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

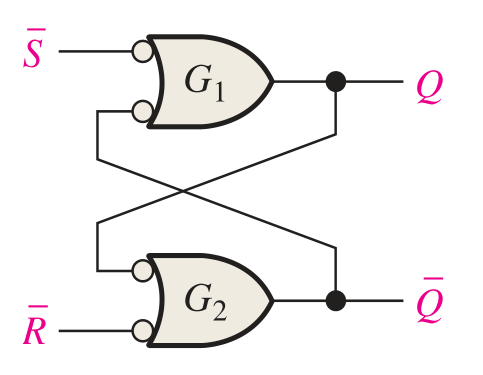
1. **RS Flip-Flop의 결과 및 simulation 과정에 대해서 설명하시오. ( Verilog source, simulatinon 출력 예시, table 등의 과정 상세히 적을 것) NAND와 NOR 두가지 모두 작성할 것.**

지난 발표와 예비보고서 작성하면서 RS flip-flop에 대해서 조사하였다. RS flip-flop의 RS는 Reset과 Set 약자로, 정보를 저장하는 기능을 함에 있어 Reset은 0으로, Set은 1로 정보를 바꾸는 역할을 한다. 따라서 active-high인 경우 R의 값으로 1이 입력되면 저장된 정보를 나타내는 Q의 값이 0으로 세팅된다. 반대로 S의 값이 1로 입력되면 저장된 정보를 나타내는 Q의 값이 1로 세팅된다. 여기서 추가로 두개의 입력 모두 1인 경우 주의해야한다. Flip-flop 회로에서는 저장된 정보 외에 출력으로 저장된 정보의 inverter값이 같이 출력되는데 회로의 feedback 특성으로 인해 RS=11인 경우 Q와 Q’값이 모두 0이 되어 유효하지 않은 상황이 나오므로 이런 경우는 나타나지 않도록 한다. 그 외, RS=00인 경우는 이전 저장된 정보인 Q의 값을 그대로 유지한다. 작동을 table로 작성하고, 위에 까지의 회로를 schematic으로 나타내면 다음과 같다.

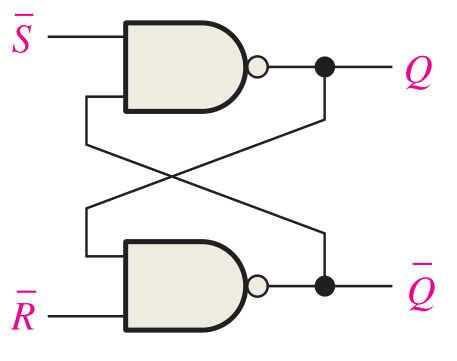
|  |  |  |  |
| --- | --- | --- | --- |
| **R** | **S** | **Q\*(다음 정보)** | **동작** |
| **0** | **0** | **Q(이전 정보)** | **No change** |
| **0** | **1** | **1** | **SET** |
| **1** | **0** | **0** | **RESET** |
| **1** | **1** | **X** | **Invalid** |



아래의 그림에서 Gate는 latch에서 사용하는 enable의 역할을 하는데 이를 flip-flop에 맞도록 clock으로 조정해주면 된다. 위에서 확인할 수 있듯 NOR게이트를 이용하여 회로가 구성되어 있는 것을 확인할 수 있다. 이를 active-low로 바꿔주게 되면 다음과 같다.



Active-low로 S와 R에 각각 inverter를 해주고 OR게이트를 사용할 수 있다. 이 inverter와 OR게이트를 NAND게이트로 바꾸어 회로를 다시 구성한다.



이렇게 구성하면 NAND게이트와 NOR게이트를 사용하여 RS flip-flop의 회로를 만들 수 있다. 이제 Verilog코드에서 작성하여 결과를 확인해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

Verilog source코드와 tb코드이다. 우선 NOR게이트에 대한 회로부터 구성했다. Q는 정보를 저장하는 변수이고, nQ는 Q의 보수값을 갖는 변수이다. RS는 각각 flipflop의 input이고, clk는 clock을 의미한다. tb코드도 살펴보자. S, clk변수는 시작시 1의 값을 갖도록, R는 0의 값을 갖도록 세팅하고, S는 500ns마다, R은 250ns마다, clk는 100ns마다 이전의 값에 보수의 값을 가지도록 세팅했다. 이에 따른 Q와 nQ의 값을 확인한다. Simulation 결과는 다음과 같다.

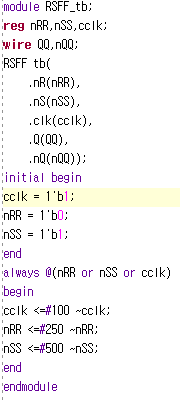
텍스트, 화면, 스크린샷, 닫기이(가) 표시된 사진

자동 생성된 설명

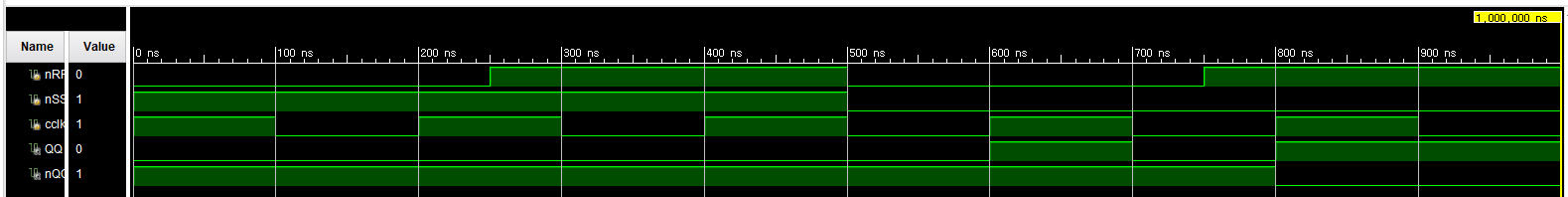
Clk의 역할은 flip-flop의 작동을 제어하는데 verilog 코드로 구현한 회로에서는 level triggered 방식으로 작용했다. 앞에서부터 확인해보자. Clk와 R,S가 1로 설정이 되어있다. 시작하자마자 clk의 값이 1이므로 회로가 작동하고, S는 1, R은 0이므로 Set의 동작을 수행, Q에 1을 저장한다. 이때, nQ는 보수의 값을 취해야하는데 0으로 되어있는 것을 보면 잘 작동한 것을 알 수 있다. 그 이후, clk의 값이 0이 되면 회로가 작동하지 않으므로 Q와 nQ가 그대로 유지된다. 다음 clk가 1인 구간을 보자. 이떄 처음 clk가 1이되고 나서는 R=0, S=1이다. 이때는 이전 상황과 같으므로 Q의 정보가 유지된다. 하지만 R이 1로 변하고 R과 S가 1이 되면 Q와 nQ의 값이 모두 0이 되는 것을 확인할 수 있다. 이는 회로 구성 단계에서 보수값을 취하자는 것에 모순되므로 이런 경우는 없도록 한다. 이후 600ns와 800ns를 보자. 600ns의 경우 S와 R의 값이 모두 0이고 not change 동작을 수행해 Q값의 변화가 없는 것을 알 수 있다. 800ns에서는 R=1, S=0으로 Reset동작을 수행하는데 이전 저장 정보가 0이므로 이 또한 변화가 없는 것을 알 수 있다.

NAND게이트로 구성한 회로도 verilog 코드로 구현해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명 

tb코드는 변경하지 않았고, source만 바꿨다. nR과 nS는 R과 S의 보수를 의미하며 nR과 nS가 0일 때, 그에 맞는 동작을 수행하도록 한다. Q와 nQ는 NAND게이트를 사용한 것을 확인할 수 있다.



Active-low인 경우, nR과 nS가 둘다 0의 값을 가질 때 invalid하고, nR와 nS가 둘 다 1일 때, no change한다. 앞에서부터 확인해보면, 처음 clk가 1일 때, nR만 0이므로 Reset 동작을 수행하여, Q에 0이 저장된다. 이후 400ns를 보면 nR, nS 모두 1이므로 no change한다. 600ns에서 nR, nS모두 0이므로 invalid를 확인할 수 있고, 800ns에서 nS만 0으로 Set 동작을 수행하여 Q의 정보가 1로 set된 것을 확인할 수 있다.

1. **D Flip-Flop의 결과 및 simulation 과정에 대해서 설명하시오. ( Verilog source, simulatinon 출력 예시, table 등의 과정 상세히 적을 것)**

D flip-flop은 RS flip-flop의 input을 간소화한 회로이다. 기존 SET과 RESET을 S와 R, 두가지의 input을 통해 진행했는데 이를 합하여 D로 간소화하여 D가 1인 경우 SET을 D가 0인 경우 RESET의 동작을 수행한다. 이를 table로 작성하며 다음과 같다.

|  |  |  |
| --- | --- | --- |
| **D** | **Q\*** | **동작** |
| **0** | **0** | **RESET** |
| **1** | **1** | **SET** |

D flip-flop은 RS를 간소화했기 때문에 회로의 구성은 RS와 매우 흡사하다. 이를 schematic으로 보면 다음과 같다.

밤하늘이(가) 표시된 사진

자동 생성된 설명

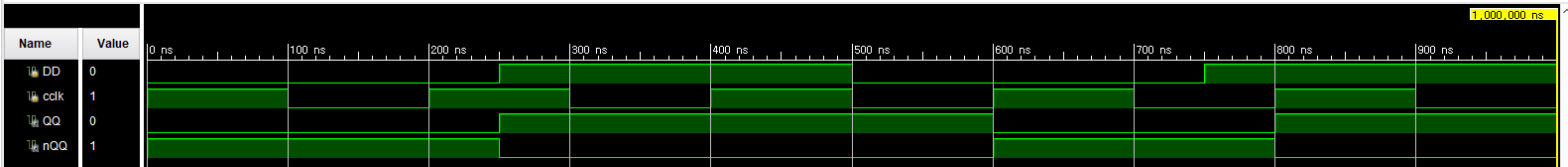
E는 latch에서 enable의 역할을 하는 것으로 flip-flop에선 clock변수로 대체된다. 회로 구성은 RS와 동일한 것도 확인할 수 있고 다른 점은 RS가 D로 대체되어 D가 그대로, inverter를 통해 회로의 input으로 들어간다. 이를 verilog코드로 작성해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

위의 schematic그대로 D와 clk input과 Q, nQ output으로 코드를 작성했다. Q와 nQ는 보수의 관계를 갖는다. tb코드를 보면 clk는 시작 시 1로, D는 시작 시 0으로 세팅하고, clk는 100ns마다 이전 값의 보수값을 취하도록, D는 250ns마다 이전 값의 보수값을 취하도록 했다. Simulation 결과는 다음과 같다.



Simulation 결과에서 clk가 1로 purse가 유지되는 동안 D의 값에 따라 Q의 값이 변화하는 것을 알 수 있다. Purse가 유지되지 않는 clk=0인 구간에서 회로는 작동하지 않고 이전의 정보를 그대로 저장하고 있는 것도 확인할 수 있다.

이 D flip-flop은 RS와 다르게 invalid의 값이 없다. simulation에서도 Q와 nQ의 값이 항상 보수의 값을 유지한다.

1. **결과 검토 및 논의 사항.**

RS와 D Flip-flop의 기능과 구성 대로 Verilog 코드를 작성하여 simulation 결과를 확인하고 fpga로도 확인했다. fpga에서는 핀을 할당하여 RS와 D, clock을 SW에 배정하고, LED로 Q와 nQ의 값을 확인했다. simulation결과와 FPGA 동작 모두 구성하고자 했던 회로 그대로 나오는 것을 확인할 수 있었다.

1. **추가 이론 조사 및 작성.**

이번 실습에서 JK flip-flop은 구성하지 않았다. JK flip-flop은 RS flip-flop을 보완한 회로로 J가 S의 역할을, K가 R의 역할을 한다. 하지만 RS에서 유효하지 않았던 RS=11의 경우 JK에서는 control할 수 있도록 보완했다. JK=11인 경우 invalid가 아닌 toggle의 동작을 수행한다. toggle이란 Q와 nQ의 값을 바꾸는 동작으로 결과적으로는 이전 정보의 보수값을 취하는 동작으로 이해할 수 있다. 이외, JK=00인 경우, RS=00과 동일하게 no change 동작을 수행한다.