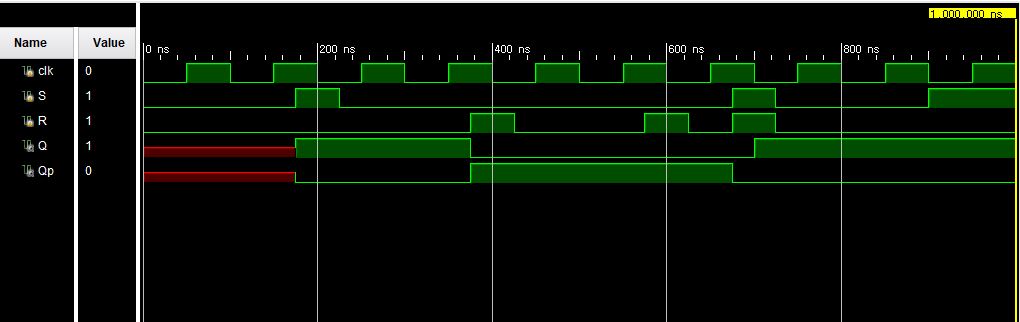
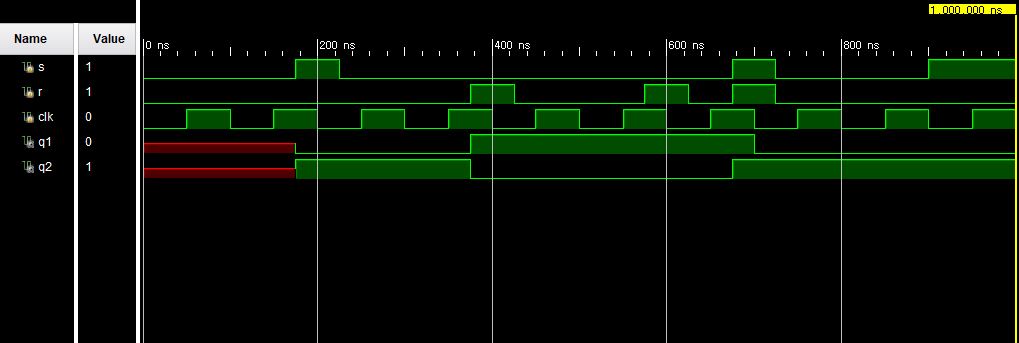
11주차 결과보고서

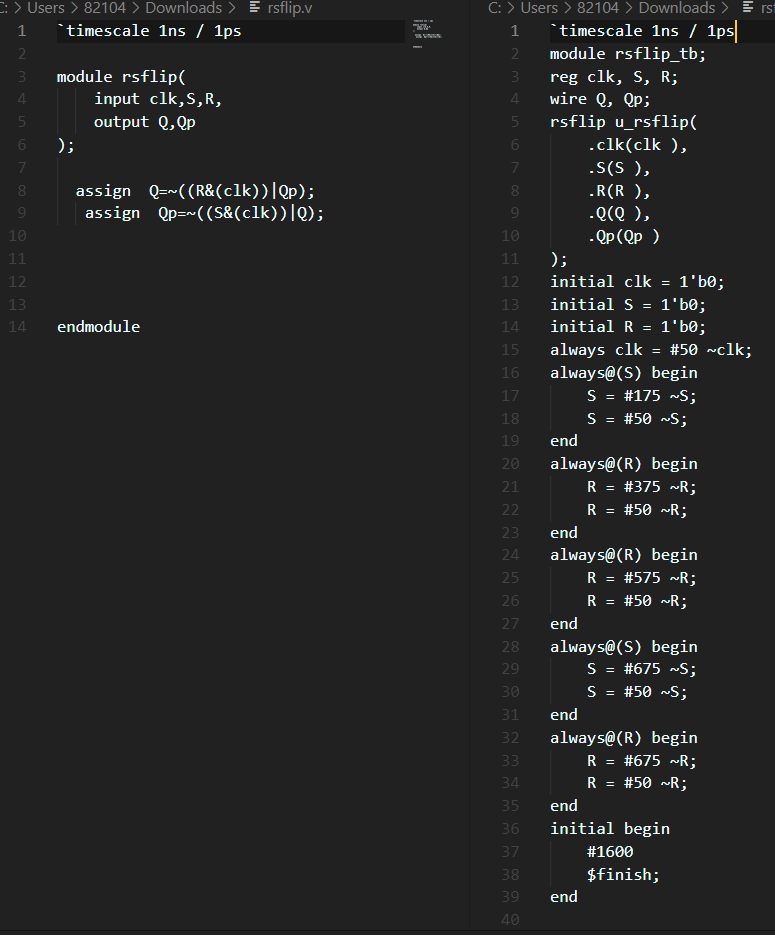
전공: 컴퓨터공학과 학년: 3학년 학번: 20191612 이름: 윤기웅

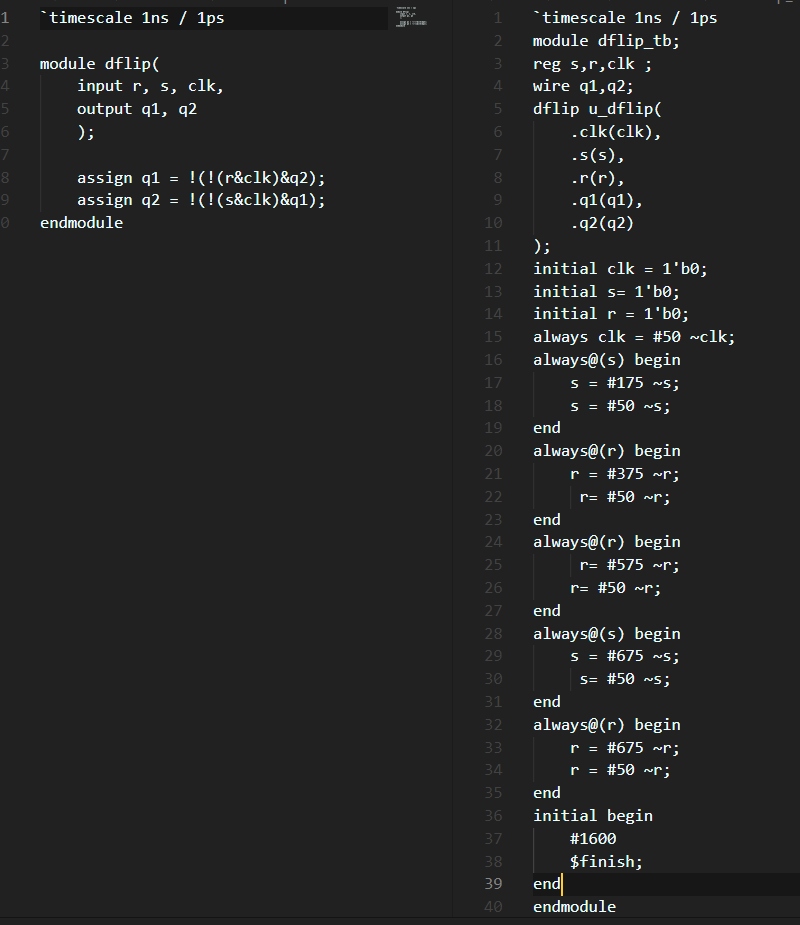
1.

Rs 플립플롭을 만들 때 NAND, NOR로 만들 수가 있다. 이 두 경우가 각각 회로도의 구성이 달라지고 코드도 달라지지만 실행은 둘 다 같게 되어야 한다. 먼저 입력으로 R과 S가 있고 Clock의 변화를 반영하기 위한 Clk 변수도 만들어 주어야 한다. R과 S가 모두 1인 경우 결과값 Q, Q'는 정의하지 못하고 R만 1인 경우 Q는 리셋을 해주어서 0으로 S만 1인 경우는 셋을 해주어서 Q를 1로 만들어야 한다. 그리고 입력 두 개가 모두 0인 경우는 결과값에 영향을 주지 않고 이전의 값을 유지한다. 공통적으로 플립플롭이기 때문에 clock 값의 영향을 받아서 Q와 ~Q가 결정된다. NOR를 사용하는 경우는 Q= ~(Q' or (R and Clock)), Q'=~(Q or (S and Clock) )의 식으로 정의가 가능하다. verilog code를 작성하는 경우 Clock도 하나의 변수로 설정하고 시뮬레이션을 실행시 값의 변화를 준다. NAND를 사용하는 경우는 Q=~(Q' and (~(S and C))) , Q' = ~(Q and (~(C and R)))의 방식으로 결과값을 정의 가능하다.





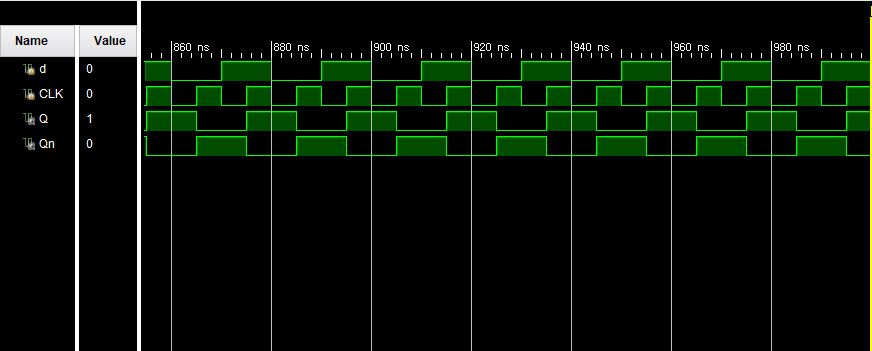


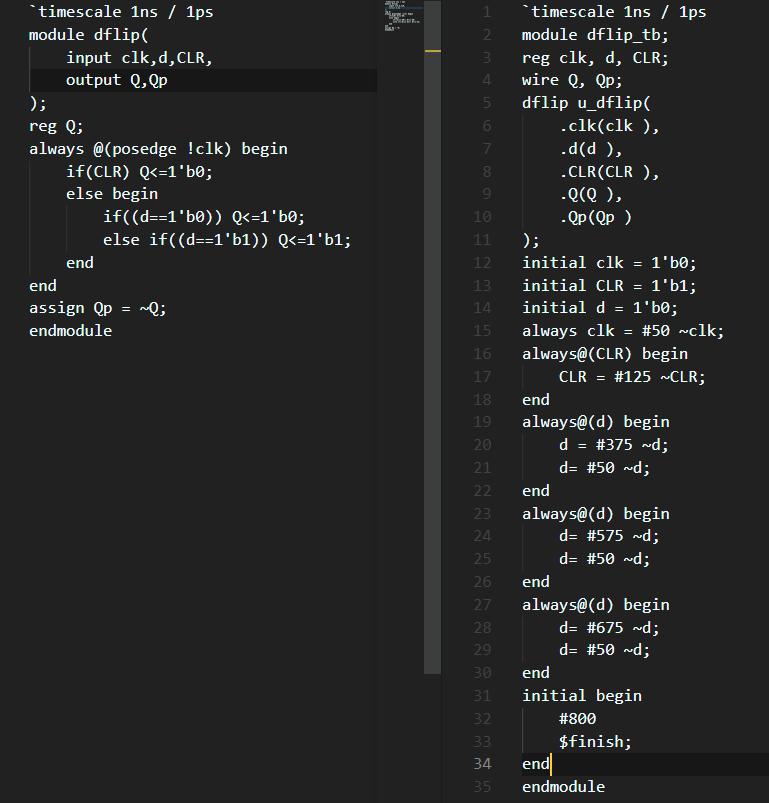


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | | | Output | |
| 차례 | R | S | Q | ~Q |
| 1 | 0 | 1 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 1 | X | X |

2.

D flip flop은 입력으로 d 하나만 존재하고 다른 플립플롭과 마찬가지로 clock의 영향을 받는다. 이는 하지만 d의 영향만 받고 이전의 상태는 결과값에 영향을 주지 못한다. 입력이 0이면 결과 값이 0이 되고 입력이 1이면 결과값이 1이 된다. 이번 코드에서는 하강 엣지에서 d에 대한 결과값을 받을 수 있게 한다. CLR 또한 입력으로 받았는데 이는 입력 활성화를 확인하기 위함이다.





|  |  |  |  |
| --- | --- | --- | --- |
| 입력 | | 출력 | |
| 차례 | D | Q | ~Q |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 |
| 3 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 |

3.

디지털 회로 개론에서 하강 엣지가 기본적인 설정이라고 해서 하강 엣지로 만들려고 했는데 다음에는 상승 엣지로도 한 번 만들어 보면 좋을 듯 하다. 원래는 Rs플립플롭도 조건문을 사용하고 CLR를 적용하려고 했지만 입력 두 개가 모두 0인 경우에 clock을 변경시키면 결과값 두 개가 모두 불이 들어오는 오류가 생겼다. 그래서 clock 또한 하나의 변수로 지정해서 코드를 다시 작성하니 오류가 사라지고 Q, Q'가 항상 보수가 되어 결과가 나타났다.

4.

JK 플립플롭 : 입력 J, K는 각각 RS플립플롭의 S, R에 해당한다. 입력 2개 모두 1이 될 때 발생하는 오류를 해결하기 위해서 새로 만들어진 플립플롭이다. 그 특성을 정리하면 J, K가 모두 0인 경우 결과값은 현재가 그대로 출력되고 K만 1인 경우는 0의 출력을 갖게 되고 J만 1인 경우 1의 결과값을 갖는다. 그리고 두 입력 모두 1인 경우는 현재 상태의 보수가 출력된다. 이를 만약 베릴로그 코드로 작성하면 always @()을 사용해서 경우의 수들을 나누고 각각에 경우에 대해서 결과값을 배정해주면 된다.

