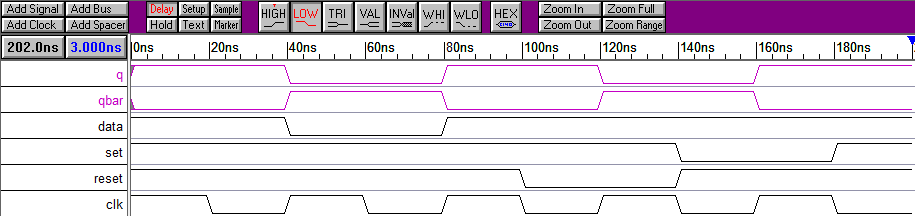
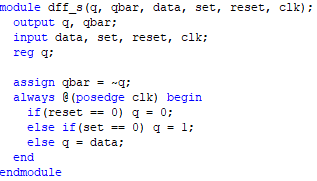
임베디드하드웨어설계

실습 4: Verilog의 동작적 모델링 논리 설계

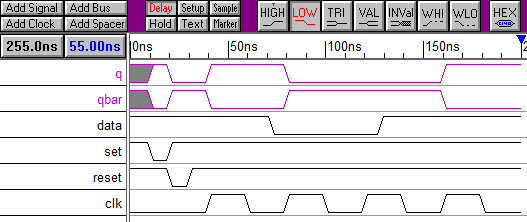
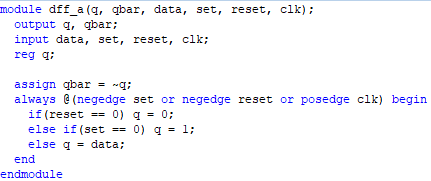
2014253005 박세현

**1** 플립플롭

(1) 동기 reset 제어입력이 있는 **positive edge triggered D** 플립플롭을 always 문을 사용하여 설계하고 적절한 입력을 제공하여 동작을 검증하시오. (reset이 0일 때에 클럭의 postive edge에서 출력이 0이 된다.)



(2) 비동기 reset 제어입력이 있는 **positive edge triggered** D 플립플롭을 설계하고 동작을 검증하시오. (reset이 0일 때에 즉시 출력이 0이 된다.)



(3) 위의 두 가지 D 플립플롭의 동작을 비교하고 설계할 때에 차이점을 적으시오.

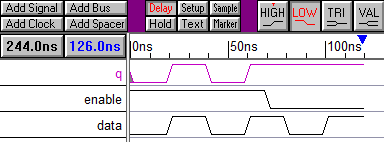
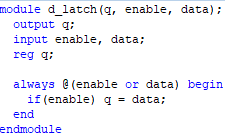
- 동기 reset 제어입력이 있는 positive edge triggered D 플리플롭은 set과 reset이 clk와 synchronous 하다. 따라서 set과 reset의 값이 0으로 바뀌어도 clk가 positive edge 신호를 보내기 전까지는 적용되지 않는다.

비동기 reset 제어입력이 있는 positive edge triggered D 플리플롭은 clk가 positive edge 신호를 보내지 않더라도 set과 reset이 negative edge가 될 때 set과 reset이 적용된다.

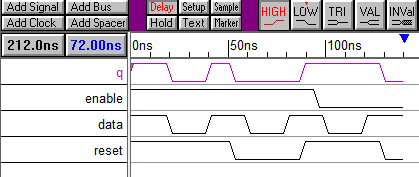
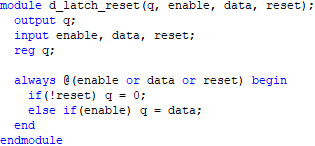
이 차이는 설계 시 이벤트제어로 negedge set or negedge reset의 조건을 같이 작성하면 비동기로 동작하고 조건을 적지 않으면 동기식으로 동작한다.

**2. D**래치 – **behavioral model** 설계

(1) 1번에서 **D**래치를 always문을 사용한 동작적 모델링 방법으로 다시 설계하고 적절한 입력을 제공하여 동작을 검증하시오.



(2) reset 제어입력(reset이 0일 때에 reset 동작 수행)이 있는 **D**래치를 (1)번 설계를 수정하여 설계하고 적절한 입력을 제공하여 동작을 검증하시오.



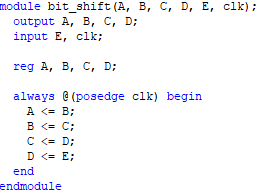
**3. shift register**

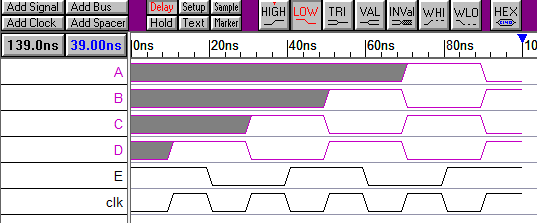
(1) blocking 할당문과 nonblocking 할당문의 동작을 비교하여 설명하시오.

- blocking 할당문은 always 안에 있는 할당문들을 순차적으로 실행한다. 한 문장의 실행이 종료되고 다음 문장이 실행되어 새로 할당된 결과가 다음 문장의 실행에 영향을 줄 수 있다. 따라서 문장의 순서에 결과가 영향을 받을 수 있다.

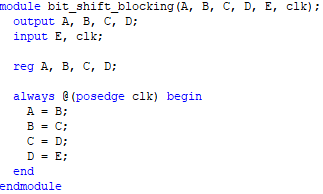
Nonblocking 할당문은 always 안에 있는 할당문들을 concurrent하게 실행하는 효과가 있다. RHS 값들을 계산한 뒤에 LHS 변수 값을 갱신하기 때문에 문장의 순서에 결과가 영향을 받지 않는다.

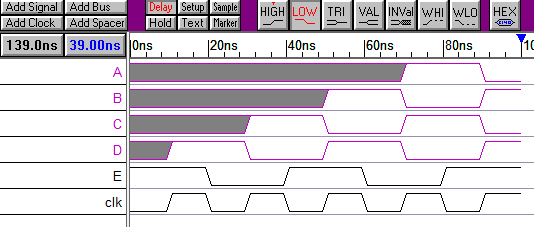
(2) 4비트 shift register를 nonblocking 할당문을 사용하여 설계하고 동작을 검증하시오.





(3) 앞의 shift register를 blocking 할당문을 사용하여 설계할 때에는 어떻게 설계해야 제대로 동작하는 지를 말하시오.

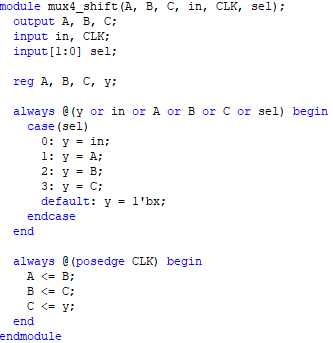


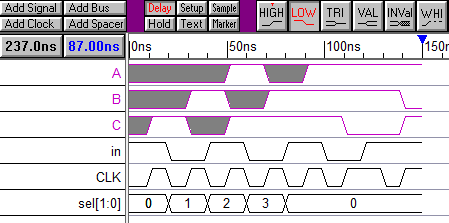


- Nonblocking 구문을 Blocking 구문으로 바꾸어 설계하려면 새로 할당된 결과가 다음 문장의 실행에 사용되지 않도록 주의하여 설계해야 한다. 위 코드의 경우 A에 B가 할당되어도 다음 문장의 실행에 사용되지 않으므로 잘 작성되었다고 볼 수 있다. 또한 이를 파형을 통해 확인할 수 있다.

**4. multiplexer**와 **shift register**의 결합 회로

(1) 아래 그림과 같은 회로를 설계하고 동작을 검증하시오. (멀티플렉서의 A, B, C 입력은 shift register의 각 출력이 피드백 되어 사용된 것을 의미한다.). 이 회로에서 in, CLK, sel을 입력으로, A, B, C는 출력으로 사용하고, 멀티플렉서는 case문으로 설계하시오.





(2) 이 회로를 blocking 할당문을 사용하여 설계할 수 있겠는가? 가능하다면 설계를 하고, 불가능하다면 그 이유는 무엇인가?

