+

**컴퓨터공학 설계 및 실험Ⅱ**

2주차 결과보고서

서강대학교 공학부 컴퓨터공학 전공

20171646 박태윤

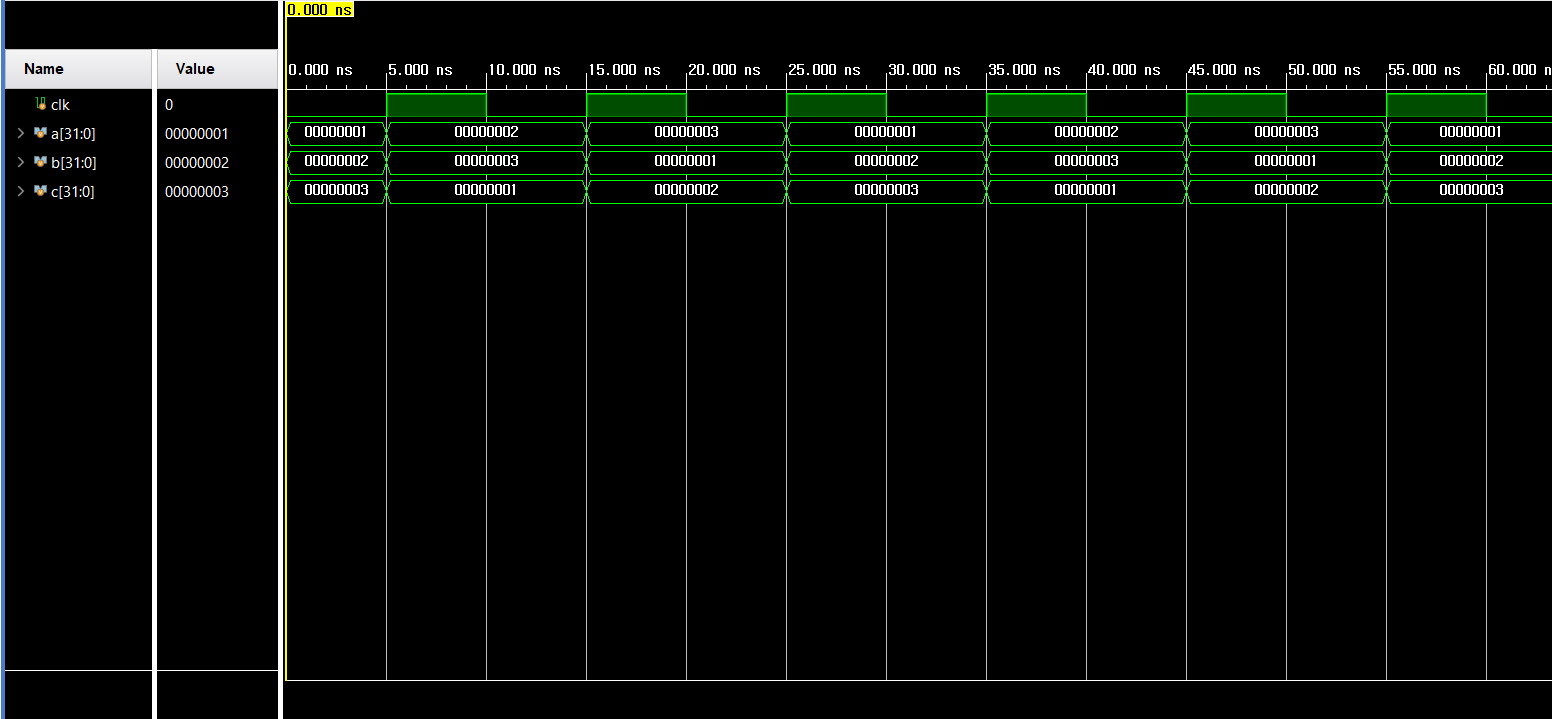
**1. 연속 할당문, 절차형 할당문의 차이를 비교하여 설명하시오.**

객체에 값을 줄 수 있는 할당문은 Verilog에서 연속 할당문, 절차형 할당문으로 두 가지가 존재한다. 우선 연속 할당문은 assign문을 통해서 net 자료형에 특정 논리 값을 지정하는 반면 절차형 할당문은 blocking or nonblocking문으로 나뉘어 reg, integer, real, time, realtime 등 variable에 값을 할당한다는 차이가 있다. 또한 연속 할당문에서는 우변 수식에 event가 발생했을 시 좌변 객체에 값을 할당하지만 절차형 할당문에서는 우변 수식의 event 발생유무에 관계없이 문장의 실행에 의해서 변수에 값이 할당된다. 즉, 절차형 할당문의 변수는 다음 값이 할당되기까지 값을 유지한다.

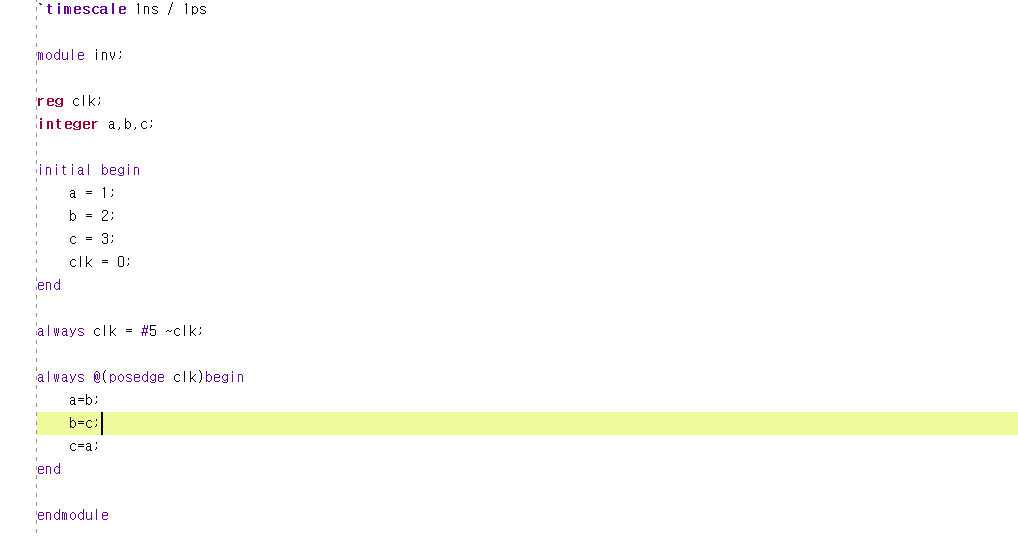
**2. Blocking 및 Nonblocking 문법의 차이를 simulation을 통해 설명하시오.**

****

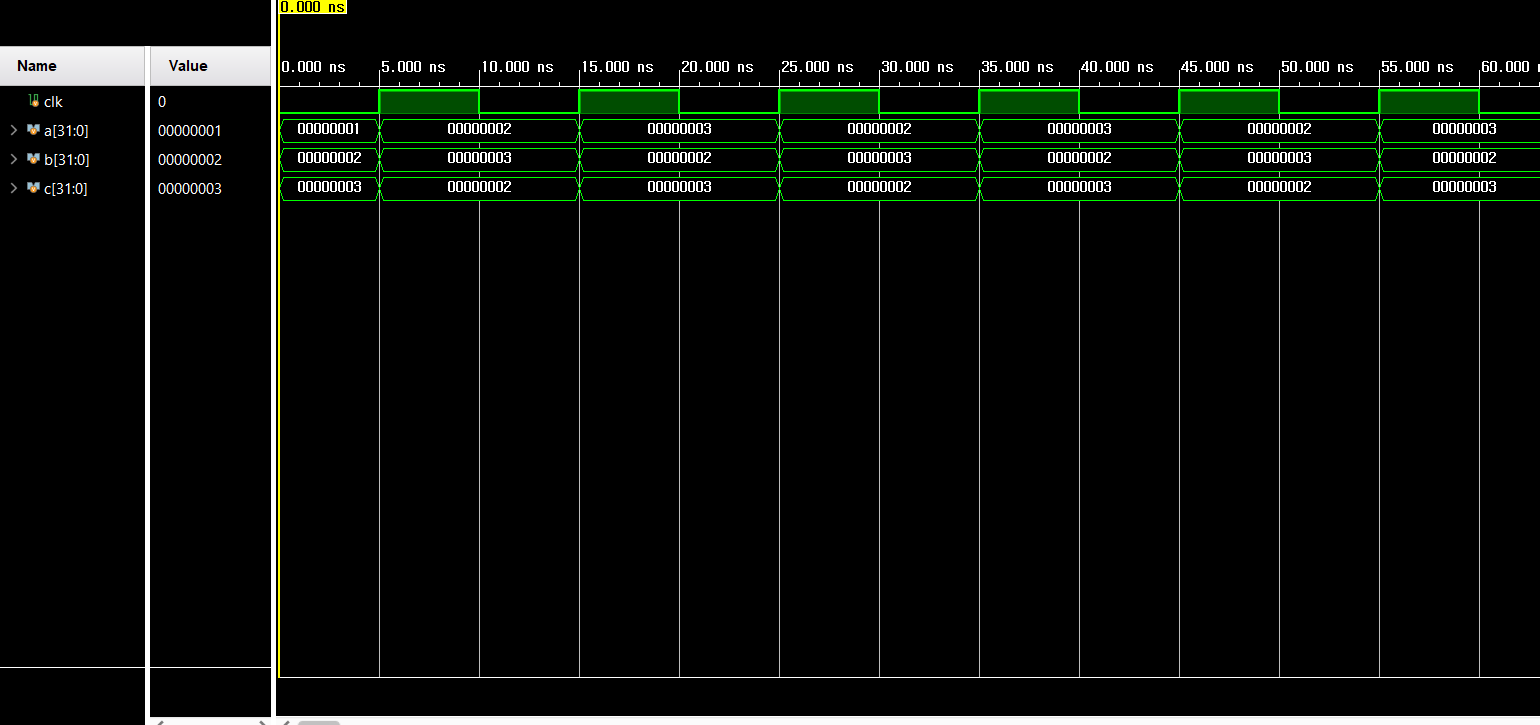
위의 그림과 같이 integer형 변수 a, b, c를 선언한 뒤 각각 1, 2, 3으로 초기화 시키고 b의 값을 a에, c의 값을 b에, a의 값을 c에 넣어주는 코드를 always구문 안에Nonblocking으로 만들어준 후 simulation을 했다.



‘timescale 1ns / 1ps으로 timescale구문을 설정해주었으므로 #5 단위시간은 5ns를 의미한다. 즉 5ns단위로 always문이 실행이 되는데 시뮬레이션과 같이 값이 지속적으로 바뀌는 것을 확인할 수 있다.



다음에는 위의 코드와 비슷하지만 blocking문으로 코드를 짜고 simulation을 했다.



앞의 경우처럼 5ns단위로 always문이 실행되어 integer variable a, b, c가 바뀌는 것을 확인할 수 있다. 단, 그 값이 non-blocking구문으로 코드를 만들었을 때와 차이가 있는 것 또한 확인할 수 있다. 그 이유는 non-blocking구문은 계산이 완전히 끝난 이후에 저장하는 작업을 실행하지만 blocking구문은 문장이 쓰여진 순서대로 계산과 동시에 저장이 이루어지기 때문이다.

**3. Verilog의 for문, if문, while문, case문을 C언어와 비교하여 설명하시오.**

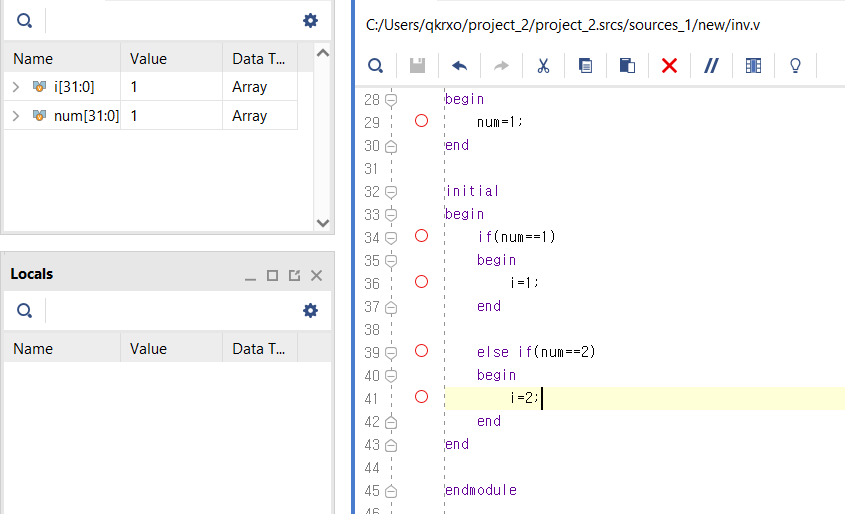
-for문

스크린샷이(가) 표시된 사진

자동 생성된 설명

위의 예시처럼 for문을 사용할 수 있다. C언어의 for문처럼 조건식에 따라 문장블록을 반복한다는 역할은 같지만 문법적으로 C언어와 달리 중괄호 없이 (begin~end)가 쓰인다. 또한 밑줄 친 부분을 i++같은 연산을 사용하면 오류를 발생시킨다는 차이가 있다. Verilog에서 반복문은 always문 또는 initial문 내에서만 사용한다.

-if문



if문 역시 C언어와 비교했을 때 문법적으로 중괄호 대신 (begin~end)를 써야한다는 차이점을 보인다. C언어의 if문과 같이 조건식을 충족시키면 if문 안의 문장 블록을 실행시키고 그렇지 않으면 else문의 문장 블록을 실행시킨다.

-while문

스크린샷이(가) 표시된 사진

자동 생성된 설명

while또한 (begin~end)이외에 큰 차이를 보이지 않는다. C언어의 while문과 마찬가지로 while문의 조건식이 거짓이 될 때까지 문장 블록을 실행시킨다.

-case문

스크린샷이(가) 표시된 사진

자동 생성된 설명

case문은 C언어와는 다소 다른 형태를 보인다. C언어에서는 case문이 switch문과 같이 쓰이지만 Verilog에서는 C언어의 switch문 역할을 case문이 하고 그 밑에

(조건: 처리문)형태로 따라오는 것을 확인할 수 있다.

**4. Verilog의 net형 자료형에 대해서 조사하시오.**

net자료형은 하드웨어 요소들 간의 물리적인 연결을 나타내기 위해 사용한다.주로 wire가 사용되며 이외에 tri, wand, wor, triand, trior, trireg, supply0, supply1, tri0, tri1이 있다.

- wire : 단순한 연결을 위해 사용된다.

- tri : wire와 같은 기능을 하지만 wire와 달리 3상태 net에 사용된다.

- wand, triand : 여러 디바이스의 출력을 연결하여 ‘and’기능을 한다.

- wor, trior : 여러 디바이스의 출력을 연결하여 ‘or’기능을 한다.

- supply0, supply1 : 전압공급기의 세기로 0(ground), 1(Vdd)을 나타낸다.

- tri0, tri1 : tri와 같지만 z값이 인가되면 pull-down(0), pull-up(1)이 된다.

- trireg : 물리적인 net에 저장되는 전하를 모델링한다.

구동자가 연결되어 있지 않을 때 x(알 수 없는 값)를 가지는 trireg를 제외하면 모두 z(high-impedance)값을 가진다.