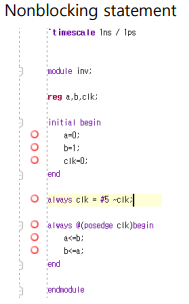
2주차 결과보고서

전공: 생명과학과 학년: 4학년 학번: 20182186 이름: 김승원

1. 연속할당문은 assign 문을 통해서 net형 객체에다가 값을 할당해준다. 만약 우변의 값에 변화가 생겼을 때, 좌변의 객체에다가 값을 할당해주는 것이다. 모든 연속할당문은 우선순위가 따로 없고 우변의 이벤트에 따라 수행되기에 할당문의 기술순서가 시뮬레이션의 결과에 따로 영향을 미치지는 않는다.

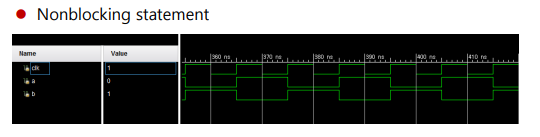
절차형 할당문은 always나 initial block, function과 같은 procedural block을 이용해서 변수에 값을 할당한다. 연속할당문과는 달리 데이터 값의 변화에 따라 할당하는 것이 아닌 구문이 실행되면서 값이 할당되어진다. 즉 연속할당문처럼 우변의 이벤트가 발생했을 때 값이 할당되는 것이 아니기에 구문이 실행되지 않는다면 좌변의 값은 안 변한다. 절차형 할당의 경우 순서대로 수행되기 때문에 할당문의 순서가 중요하다.

2. Blocking assignment는 기술된 순서대로 동작한다. 이름이 뜻하는 바와 같이 만약 현재의 statement가 수행되고 있으면 block이 되기 때문에 다음의 statement 수행되지 않는다. 반면 nonblocking assignment는 동시에 여러 개의 statement를 수행한다. 단순히 assignment 구문들을 읽으며 delta-delay 후에 값이 할당되기 때문에, 다음 statement에서는 현재 값을 사용하게 된다. 하드웨어를 설계할 때는 이러한 할당문의 순서나 타이밍에 의해서 시스템 동작이 달라질 수 있다. 따라서 무조건 현재 데이터 값을 전달해주는 nonblocking 할당문을 사용하는 것이 좋다.

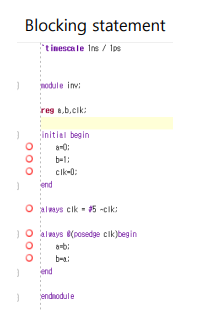


<Nonblocking statement>

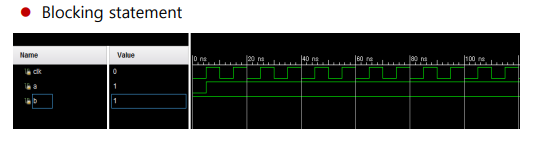
하단에서 <, >표시로 확인할 수 있듯이 이는 Nonblocking이다. Nonblocking에서 a는 a=b일 때는 1, b=a에서 0으로 되고 이후에는 각각 0, 1로 바뀌며 계속해서 반복된다.



따라서 위와 같은 simulation 결과가 나타난다.



Blocking의 경우 a의 초기값은 0, b의 초기값은 1이다. Always 구문안에서 확인할 수 있듯이 blocking statement이기에 차례대로 명령을 수행한다. 따라서 a=b일 때 1, b=a일 때 1로 각각저장된다.



따라서 시뮬레이션 결과는 위와 같다.

3.

<for문 구조>

For (초기값 ; 조건식 ; 오퍼레이션)

처리문 or

Block 처리문

으로 구성된다.

for문의 index는 integer로 선언해주고, 만약 2줄이상이 된다면 begin과 end로 묶어주어야한다.

verilog의 for문은 기본적으로 always문과 initial문에서만 사용한다.

<if 문 구조>

If (조건)

처리문 or block 처리문

Else

처리문 or block 처리문

이때도 만약 2줄이상이 되면 begin과 end로 묶어준다.

<while 문 구조>

While (조건)

처리문 or block 처리문

조건문이 false가 되면 수행을 멈춘다.

<case 문>

case (판정식)

항1 : 처리문 1 ;

항2 : 처리문 2;

:

항N : 처리문 Nl;

Default : 처리문 N+1;

endcase

판정식에 대해 해당하는 항에 가서 처리문을 실행시켜준다.

이때 만약 판정식이 항N까지 일치하지 않으면 처리문 N+1을 실행해준다. 마지막은 endcase로 닫아주면 된다.

Verilog 언어는 C언어의 for문 if문 while문에서 구조가 상당히 유사하다. 다만 다른 점은 C언어에서는 괄호로 처리문을 verilog에서는 begin과 end로 묶어준다. 또한 case문의 경우도 구조가 비슷했는데, case를 통해 판정식을 넣어주는 것에 차이가 있고, c언어의 switch와는 달리 break를 사용해서 닫아주지 않아도 된다. 또한 endcase로 마지막에 닫아주는 것도 차이가 있다.

4. net형 자료형은 device 간 물리적인 연결을 나타내준다. 값이 저장되는 것은 아니고, net 값이 변하면서 자동으로 net에 새로운 값이 전달되는 원리다. 크기를 정해주는 vector를 지정하지 않는다면 1bit net이다. Keyword는 wire, wor wand tri, trior, triland, trireg가 있고, 이를 통해 선언한다.

기본적인 문법은 예시로 wire [범위] [지연] 네트이름, ~, 네트 이름; 이다. 범위는 bit나 vector로 지정하고, 지연은 simulation에서 사용한다.