2주차 결과보고서

전공: 아트앤테크놀로지 학년: 3학년 학번: 20191098 이름: 백승주

**1.**

**1-1.** **연속 할당문**

연속 할당문은 net 변수에 논리 값을 할당할 때 사용하는 방법이다. Net 변수로는 wire, tri 등이 있다. assign 구문을 활용해 net 변수를 선언하며 할당할 수 있는데 그 형태는 다음과 같다.

**wire net\_a = enable & data;**

이 때 assign 구문은 enable & data의 피연산자 값에 변화가 발생할 때마다 우변의 식이 계산되고 그 결과값이 net\_a 변수 값에 할당되는 하드웨어적 특성을 갖는다.

assign 구문으로 변수에 할당한 값을 지울 때는 deassign 구문을 사용할 수 있다.

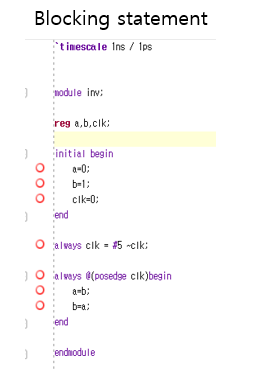
**1-2.** **절차형 할당문**

절차형 할당문은 reg나 integer과 같은 variable 자료형 객체에 값을 할당할 때 사용하는 방법이다. 절차형 할당문은 always 구문이나 initial 구문 내에서 사용이 가능하다. 절차형 할당문은 다음 할당까지 기존에 할당 받은 값을 유지하기 때문에 할당문들의 배치 순서가 결과에 영향을 끼칠 수 있게 된다. Blocking statement와 Nonblocking statement는 절차형 할당문 내부에 포함된다.

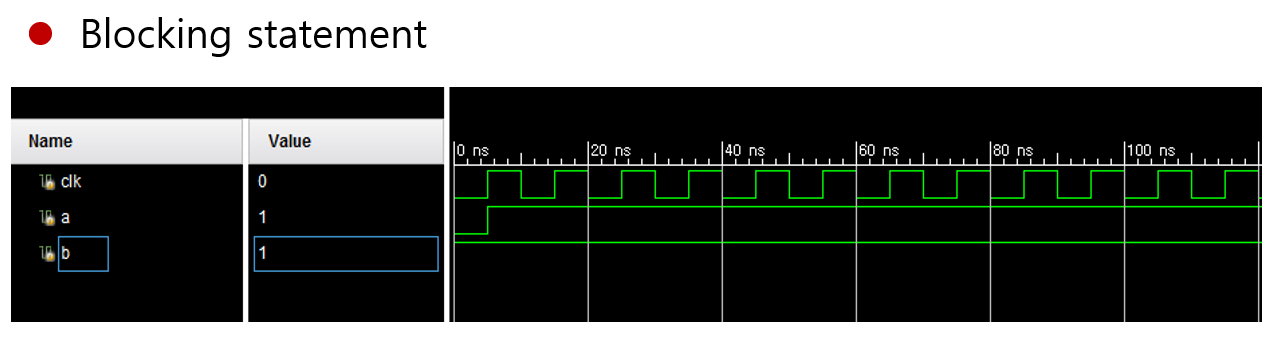
2.

**2-1. Blocking statement 문법**

할당 기호 ‘=’를 사용하는 Blocking statement는 하나의 문장씩 순서대로 실행된다.



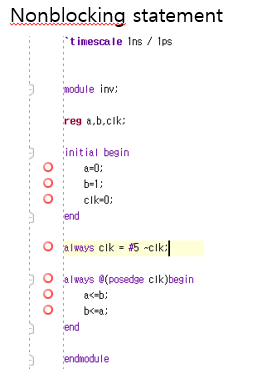
위의 코드는 Blocking statement 코드의 예시이다. a=b; 이후 b=a;식이 나열되었기 때문에 먼저 ‘a=b’에 의해 변수 a에 b의 값인 1이 할당된다. 이후 ‘b=a’ 식에 의해 변수 b에 a값인 1이 할당되어 a ,b 모두 1의 값을 가지게 된다.



위의 시뮬레이션 결과에서 a와 b 모두 1이 된 결과를 확인할 수 있다.

**2-2. Nonblocking statement 문법**

할당 기호 ‘<=’을 사용하는 Nonblocking statement는 문장의 순서와 상관없이 모든 계산과 저장을 한꺼번에 수행한다.



위의 코드는 Nonblocking statement 코드의 예시이다. a<=b; 이후 b<=a;식은 순서와 관계없이 동시에 실행된다. ‘a=b’ 식에 의해 변수 a에 b의 값인 1이 할당되고 이와 동시에 ‘b=a’ 식에 의해 변수 b에 a의 변하기 전의 값인 0이 할당되어 a는 1, b는 0의 값을 가지게 된다.

3.

**3-1. Verilog의 for문과 C언어의 for 문**

for( i =0; i<n; i++){

…

}

for( i =0; i<n; i+1){

begin

…

end

}

Verilog 문법에서의 for문은 for (초기값; 조건; 변수 증가 또는 감소)로 기본 형태가 C언어와 같지만 다르게 ++ 나 – 연산자 사용이 불가능하고 begin과 end로 블록을 설정해 사용해야 한다.

**3-1. Verilog의 for문과 C언어의 for 문**

for( i =0; i<n; i++){

…

}

for( i =0; i<n; i+1)begin

…

end

Verilog 문법에서의 for문은 for (초기값; 조건; 변수 증가 또는 감소)로 기본 형태가 C언어와 같지만 다르게 ++ 나 – 연산자 사용이 불가능하고 begin과 end로 블록을 설정해 사용해야 한다.

**3-2. Verilog의 if문과 C언어의 if 문**

if(조건)begin

…

}

if(조건)begin

…

end

Verilog 문법에서의 if문은 기본 형태가 같지만 C언어와 다르게 begin과 end로 블록을 설정해 사용해야 한다.

**3-3. Verilog의 while문과 C언어의 while 문**

while(조건){

…

}

while(조건)begin

…

end

Verilog 문법에서의 while문은 기본 형태가 같지만 C언어와 다르게 begin과 end로 블록을 설정해 사용해야 한다.

**3-4. Verilog의 case문과 C언어의 while 문**

switch(x){

case 0:

a=1;

case 1:

b=2;

default:

break;

…

}

case(x)

0: a=1;

1: b =1;

end case

Verilog 문법에서는 ‘값 : 명령문’ 형태로 값에 따라 명령을 실행하고 switch와 break을 쓰지 않는다. 또한 begin 없이 endcase만 써서 마무리하면 된다.

**4.** net 자료형 변수는 게이트와 같은 하드웨어 디바이스 간의 물리적 연결을 나타내는 자료형이다. 그 종류로는 wire, tri, triand, wand, war, .. 등이 있다. 이 중 모듈 내 변수들이 어떻게 연결되었는지 나타내는 wire 변수를 주로 사용한다. net 변수는 연속 할당문에 의해 선언되고 그 크기를 지정하는 벡터를 정하지 않으면 디폴트 값인 1 bit net 이 된다. 또한 변수의 값은 디폴트 값인 z가 된다.