2주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

연속할당문은 net형 변수, 즉 wire type과 같은 변수에 값을 할당하는 것으로, assign 문을 활용한다. assign 변수이름 = 값의 형태로 할당을 하고, assign 뒤에 delay를 설정할 수도 있다. 연속할당문은 우변의 값, 즉 할당한 값에 변화가 생기면 변수에 할당이 일어나는 특징이 있다. Delay를 설정했을 경우에는 해당 delay 이후 값의 할당이 일어난다. 또한 assign의 경우에는 순서에 상관없이 회로가 동일하게 그려진다.

절차형할당문은 net형 변수 이외의 경우, 즉 integer, reg와 같은 type의 변수에 값을 할당하는 것이다. 절차형할당문은 always, initial과 구문 내에서 사용한다. 이 경우에는 assign과 다르게 순서가 달라지면 회로가 바뀔 수 있고, 구문이 실행되어야 값이 할당된다. 또한 절차형할당문에서는 blocking과 non-blocking 할당문으로 나뉘는데, 이는 blocking의 경우에는 =로, non-blocking에서는 =>을 사용하는 것으로 구분한다. 자세한 차이는 2번 문항의 답이므로 이에 이어서 설명한다.

................

**2.**

.......................

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Blocking statement는 위처럼 a=b; b=a;처럼 사용하여 값을 할당하고 있는 점을 볼 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

nonBlocking statement는 위처럼 a<=b; b<=a;와 같이 표현하고 있는 점을 볼 수 있다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

첫 번째로 Blocking statement의 simulation을 보면 처음에 a의 값이 바뀐 후 양쪽이 1로 유지되는 것을 확인할 수 있다. 한 줄씩 실행될 때마다 값의 할당이 일어나므로 처음에 0이던 a는 1인 b의 값이 저장되며 1이 되고 이 후에 b=a에서는 a가 1이므로 그대로 1로 유지되어 그 뒤로도 1로 유지된다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

두 번째로 nonBlocking statement에서는 a<=b; b<=a;가 모두 실행된 후에 값의 할당이 일어난다. 따라서 처음의 a<=b에서는 기존 b의 값인 1이 저장되는 연산을 한 후에 우선 b<=a가 연산되어 기존에 a가 0인 값이 전달된 후에 각각 연산에 맞는 값을 할당한다. 따라서 0과 1이 반복해서 서로 바뀌는 결과로 위처럼 나타난다.

.........................

**3.**

.......................

우선 for문은 verilog에서는 for(변수; 조건문; 값 변화)begin~end의 방식으로 작성한다. 여기에서 begin과 end 사이에 코드를 작성한다. C에서는 for(변수; 조건문; 값 변화){~}로 {} 사이에 코드를 작성하는 것으로 차이가 있다. 이외에 프로그램 내에서 사용하는 방법의 자체에 차이는 없다.

if문은 verilog에서 if(조건) begin~end의 방식으로 ~ 사이에 실행되는 코드를 작성하는 방식으로 구현된다. C언어에서는 if(조건) {~}의 구조로 {} 사이에 코드를 작성하는 것과의 차이가 존재한다. else나 else if 역시 위의 if와 사용이 비슷하다. else if(조건) begin~end, else begin~end의 구조를 보인다. 이 역시 C에서는 {}로 감싸는 것과의 차이가 있다. 단 verilog에서 실행문이 한 줄인 경우에는 begin end를 생략가능하다. 이 역시도 작동하는 방식에 있어서는 차이가 없다.

while문은 위의 for문과 비슷하다. while(조건) begin~end의 구조로 begin과 end 사이에 코드를 작성한다는 점에서 C언어와 차이가 있다. 사용하는 방식 자체는 C언어와 동일하다.

case문의 경우에는

case(변수)

value 1: 실행코드

value 2: 실행코드

default: 실행코드

endcase

와 같은 구조로 이루어져서 사용된다. 이 경우에도 실행코드가 두 줄 이상인 경우에는 각각의 실행코드를 begin end로 감싸야 한다. 또한 C언어와의 차이점으로는 실행코드 마지막에 break를 붙이지 않는다는 점이 있고, 마지막에 endcase로 감싸서 끝남을 나타내는 점에 차이가 있다.

.........................

**4.**

.......................

net 자료형은 모듈이나 게이트 등의 물리적인 연결을 나타내기 위해서 사용되는 자료형이다. 첫 번째로 wire가 있다. 이는 논리적 동작이나 기능을 갖지 않는 단순한 연결을 위해 사용하는 자료형이다. 다음으로 wand와 wor은 각각 wire의 확장형으로 각각 and와 or의 기능을 수행한다. 다음으로는 tri 자료형이 있다. 이는 wire와 동일하지만 하드웨어에서 3상태가 되는 부분이 있는 점에 차이가 있다. tri0와 tri1은 각각 저항성 pulldown에 의해 접지로 연결, 저항성 pullup에 의해 전원으로 연결되는 자료형이다. 이는 각각의 경우 0, 1의 값을 가지는 것을 의미한다. triand와 trior은 tri 자료형과 wand, wor이 각각 결합된 형태이다. 하드웨어에서는 3상태를 가지면서 각각 wand와 wor을 출력한다. 다음으로 trireg는 tri와 reg의 결합형태로 이전 값을 저장하는 기능이 있다. 마지막으로 supply1은 전원에 연결되는 것을, supply0는 접지에 연결된 것을 표현하는 자료형이다.

.........................