전공: 컴퓨터공학 학년: 2학년 학번: 20161603 이름: 신민준

1. 연속 할당문, 절차형 할당문의 차이를 비교하여 설명하시오.

Verilog의 할당문은 다른 언어와 동일하게 어떠한 객체에 값을 주는 기본적인 방법이다. Verilog에서 이 할당문은 두 가지 종류가 존재한다.

먼저, 연속 할당(Continuous Assignment)은 net형 객체에 값을 할당하는 데 쓰인다. 이는 assign 키워드를 사용해 net형의 객체에 Scalar/Vector의 값을 할당하고, 우측에 쓰인 수식 값에 이벤트, 다시 말해 변화가 생겼을 때, 좌측에 쓰인 객체의 값에 해당 변화가 적용되어 보여진다. 연속 할당문은 주로 논리식으로 표현되는 회로의 모델링에 이용된다.

절차형 할당(Procedual Assignment)은 always, initial, task, function과 같은 절차(procedure) 내에서 사용되고, 시뮬레이션을 할 때 그 흐름이 특정 procedure 안에서의 할당문에 도달했을 때 이 할당을 위한 트리거가 실행된다. 연속 할당문이 우측 수식의 값 변화 이벤트에 의해 죄측의 net객체 값이 할당되는 방면, 절차형 할당문은 우측 수식의 event와는 상관 없이 해당 수식의 실행에서 좌측의 variable에 값이 할당되는, 소프트웨어에서 흔히 볼 수 있는 할당문의 성격을 지닌다.

1. Blocking 및 nonblocking 문법의 차이를 simulation을 통해 설명하시오.

스크린샷이(가) 표시된 사진

자동 생성된 설명

Figure 1 Blocking statement 예시

Blocking statement에서는 Blocking symbol로 ‘=’를 사용하며, begin부터 end까지 Line by line으로 계산을 순차적으로 수행하고 그와 동시에 저장을 수행한다. 즉, 한 Line에서 할당까지 다 수행을 한 후, 다음 문장을 수행하므로, 한 문장이 수행이 끝나기 전에 blocking된다고 볼 수 있다. t초 후 특정 연산을 수행시키기 위해선 #t variable = operation; 이라 코딩하면 된다.

Figure 2에서는 Figure 1의 예시 코드를 실행한 결과 시뮬레이션을 볼 수 있다. 두 연산 a=b와 b=a가 순차적으로 수행되므로, 결과적으로 소프트웨어 프로그래밍에서의 b=a=b와 같은 작업이 되어 Figure 2와 같은 결과가 나온다.

개체이(가) 표시된 사진

자동 생성된 설명

Figure 2

Non-blocking statement에서는 non-blocking symbol로 ‘<=’를 사용하며, begin에서부터 end까지의 계산을 일괄적으로 수행한다. 특정 연산을 t초 후 수행하기 위해서는 #t variable = value;의 형식을 맞춰 작성하면 된다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

Figure 3 Non-blocking statement 예시

Non-blocking statement는 blocking statement와는 달리, Figure 3를 시뮬레이션한 결과를 Figure 4에서 보면, a<=b와 b<=a 명령이 동시에 실행되므로 값이 매 클럭마다 서로 뒤바뀐다는 것을 확인할 수 있다.

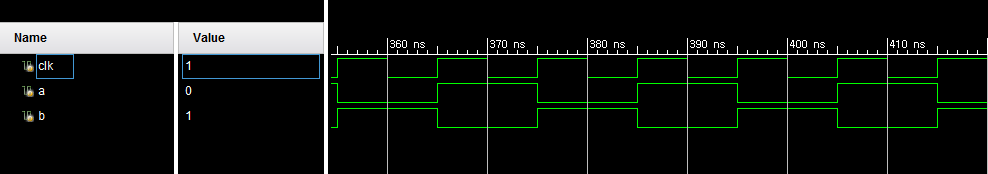


Figure 4

1. Verilog의 for문, if문, while문 case문을 C언어와 비교하여 설명하시오.

조건문은 if와 else 키워드로 구성되는데, 항상 always 블록 안에서만 사용 가능하다. C언어에서의 블록 표시인 ‘{‘, ‘}’는 begin ~ end 키워드로 대체된다. 그 외에는 두 언어는 큰 차이를 보이진 않는다.

Case문은 case()로 시작해 endcase 키워드로 끝나야 한다는 점과, 각 case 뒤에 break;를 붙이지 않는다는 것을 제외하면 C언어의 case 문과 동일하다.

Verilog의 for문과 while문은 블록 표시를 begin ~ end로 한다는 점을 제외하면 C언어에서의 문법과 동일하다.

|  |
| --- |
| **while** (count) **begin**  count = count - 1;  *// ...*  **end**  **for**(idx=0; idx=7; idx=idx+1) **begin**  state = nextstate;  *// ...*  **end** |

Figure 5 Loop문 예시

1. Verilog의 net형 자료형에 대해서 조사하시오.

Verilog에서 사용되는 net 자료형은 wire가 있다. 연결을 의미하고, 하나의 타입으로 취급된다. 일반적인 데이터 값을 이 자료형 자체로는 담을 수 없다는 특징이 있다. 일반적으로 이 wire를 자주 쓰고, wire에는 단일 비트를 전달한다. 선언을 하기 위해서는 wire name1, name2;와 같은 방식으로 프로그램을 만들면 된다. 이렇게 선언한 wire는 module의 Input/Output에 연결할 수 있고, Continuous Assignment를 통해 값을 넣을 수 있다. Wire타입의 여러 비트를 가지는 하나의 변수를 생성할 수 있다. 이를 위해선 다음과 같이 선언하면 된다: wire [최상위 비트:최하위 비트] identifier;