2주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211533 이름: 남정연

**1.**

연속 할당문인 assign 구문을 사용하면 피연산자의 변화에 따라서 net 변수에 값을 변화시켜줄 수 있다. 이와 달리 절차형 할당문은 피연산자의 변화와는 관계 없이 특정 시점에 문장이 나열된 순서대로 실행된다는 특성이 있으며 Blocking과 Non-Blocking 구문으로 구분된다.

**2.**

Blocking 구문은 = 연산자를 사용하며 구문이 끝나기 전까지는 다음 문장이 실행되지 않는다. #t를 구문의 맨 앞에 추가하여 t시간 후 연산 및 할당을 예약할 수 있다. 그러나 <= 연산자를 사용하는 Non-Blocking 구문은 여러 구문들이 동시에 평가된 후 할당된다. 사건 발생 후 여러 데이터 동시 송신을 위해 주로 사용된다. 연산자의 앞에 #t를 추가하여 연산 후 t 후에 변수 할당을 예약할 수 있다.

Input인 aa와 bb, output인 yy와 zz가 각각 아무런 게이트를 거치지 않고(혹은 버퍼만을 거치고) 연결되어 있고 aa=1’b0, bb=1’b1로 초기화되어 있다고 가정하자.

initial begin

aa = ~aa;

bb = aa;

end

위 코드에서는 aa의 값이 반전될 동안 해당 라인에서 멈추고 그 다음 라인이 실행되어 aa에는 aa가 반전된 값이, bb에도 aa가 반전된 값이 들어간다. 따라서 아래와 같은 시뮬레이션 결과가 나타난다.

Timeline

Description automatically generated with medium confidence

initial begin

aa <= ~aa;

bb <= aa;

end

그러나 위와 같이 blocking 구문을 사용하면 aa의 값이 반전될 동안 다음 구문이 그대로 실행되어 결론적으로는 aa에는 반전된 aa의 값이, bb에는 원래의 aa 값이 들어간다. 따라서 아래와 같은 시뮬레이션 결과를 볼 수 있다.

Timeline

Description automatically generated with low confidence

**3.**

for문의 경우 C언어와 동일한 구조를 가진다. 초기값;조건식;실행문의 형태로 구성된다.

if 문의 경우 C언어와 동일한 구조를 가진다. if – elseif – else로 구성된다.

while문의 경우 C언어와 동일한 구조를 가진다. 연산자를 사용하여 while문에 여러 개의 조건을 추가할 수도 있다.

case문의 경우 C언어와 약간의 차이가 있다. C언어에서는 case에 일치하면 break를 만날 때까지 다른 케이스들에 대해서도 계속 실행되는 반면 Verilog에서는 조건에 일치할 시 case문을 빠져나오게 된다.

**4.**

Net 자료형은 하드웨어 요소들 사이의 물리적 연결을 기술하기 위해서 사용된다. 연속 할당문을 사용하여 나타낸다. 크게 wire과 tri가 있다. wire와 tri 모두 단순한 연결을 위한 net 변수인데, tri는 wire와는 다르게 하드웨어 상에서 3상태가 된다.