

## 컴퓨터공학실험2

### 2주차 결과 보고서

20191621 이민영

#### 1. 연속 할당문, 절차형 할당문의 차이를 비교하여 설명하시오.

연속할당문은 net형 객체에 값을 할당하는 것이며, 절차형 할당문은 variable 자료형의 객체에 값을 할당하는 것이다.

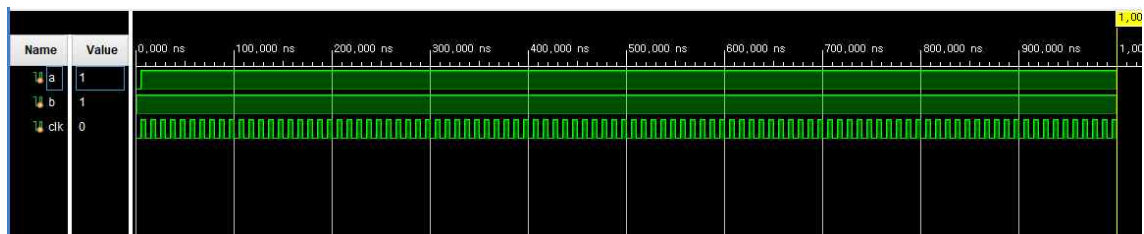
연속할당문은 assign문을 이용해서 net형 객체에 값을 할당한다. 우변의 값에 변화가 발생했을 때 좌변의 객체에 값의 할당이 일어난다. assign구문은 net 변수에 특정 논리 값을 지정하는데 사용한다 deassign 구문은 variable에 대한 assign 문의 영향을 제거시킬 때 사용한다. 연속할당문은 논리식으로 표현된 조합회로의 모델링에 이용된다.

절차형 할당문은 procedure내부에서 사용된다. 절차형 할당문은 연속 할당문과는 달리 우변의 값에 변화가 발생하는 것과 상관 없이 해당 문장이 실행될 때 좌변의 variable에 값이 할당이 되게 된다. 절차형 할당문은 반복문, if문, case문 등을 통해서 제어될 수 있고, blocking할당문과 nonblocking할당문으로 나뉜다.

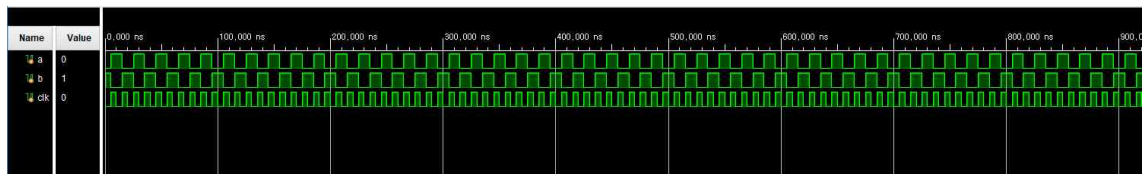
#### 2. Blocking 및 nonBlocking 문법의 차이를 simulation을 통해 설명하시오.

Blocking의 symbol은 = 이고, Begin부터 end까지 한 줄 씩 순차적으로 계산과 저장이 동시에 이루어진다.

Nonblocking의 symbol은 <= 이고, Begin부터 end까지 모든 계산을 수행한 후에 한꺼번에 저장 작업이 수행된다.



<Blocking>



<Nonblocking>

Blocking의 경우 한 줄 씩 순차적으로 계산과 저장이 이루어지기 때문에 a값에 b의 값인 1이 들어가서 a는 1이 되고, 이후에 b=a에서 a의 값이 1로 변했기 때문에 b값도 1이 된다. 즉, 계산과 저장이 동시에 이루어지기 때문에 a와 b가 모두 1로 유지되게 된다.

Nonblocking의 경우에는 begin부터 end까지 계산이 모두 수행된 이후 한꺼번에 저장된다. 즉, 처음에 a<=b에서 b의 값은 1이기 때문에 a값이 1로 변하게 된다. b<=a에서 앞부분에서 a<=b가 있었지만 아직까지 a값이 저장되지 않았기 때문에 원래 a의 값인 0이 b에 들어가게 된다. end를 만나 저장될때는 a는 1, b는 0이 된다. 이후에 다시 돌 때는 같은 방법으로 a는

0, b는 1이 되게 되고, 이렇게 번갈아서 반복되게 된다.

### 3. Verilog의 for문, if문, while문, case문을 C언어와 비교하여 설명하시오.

Verilog와 C언어의 for문은 for(초기화;종료조건;증감) 형태로 이루어진다는 공통점이 있으며 C언어와 달리 Verilog에서는 중괄호 대신 begin와 end를 사용하여 나타낸다.

Verilog의 if문은 always 블록 안에서만 사용해야 한다. 중괄호 대신 begin과 end를 사용하여 나타내는 것을 제외하고는 C언어와 문법이 같다. if를 사용하며, if-else, if-else if-else 모두 사용가능하다.

Verilog의 while문 또한 중괄호 대신 begin과 end를 사용하여 나타내는 것을 제외하고는 같다. while(수식) 형태로 사용하고, 괄호 안의 수식이 거짓이 될 때까지 while문을 반복한다.

Verilog의 case문은 C언어의 switch-case구문과 같이 많은 선택이 필요할 때 사용한다. case 문의 마지막에 endcase를 사용한다는 차이점이 있다. default는 수식에 해당하는 값이 없을 경우에 실행하기 위해서 사용된다.

### 4. Verilog의 net형 자료형에 대해서 조사하시오.

net 자료형은 논리 게이트나 모듈과 같은 하드웨어들 사이의 물리적인 연결을 나타내기 위해서 사용한다. default 자료형은 1비트의 wire이고, default 초기값은 z이다. wire 이외에 tri, wand, wor, triand, trior 등이 있다. 연속 할당문에 의해서 net의 값이 연속적으로 유지된다. wire는 함축된 논리적 동작이나 기능을 가지고있지 않는 단순한 연결을 위한 net이고, tri는 net과 비슷하지만 하드웨어에서 tri-state가 된다는 차이가 있다. wand는 다중 구동자를 가지는 net이고, wired-and 구현을 모델링하기 위해서 사용된다. wor 또한 다중 구동자를 가지는 net이고, wired-or 구현을 모델링하기 위해서 사용된다. supply0는 회로접지에 연결되는 net이며, supply1은 전원에 연결되는 net이다.