2주차 결과보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20212022 이름: 이예준

**1.**

연속할당문의 경우 피연산자의 값에 변화가 생길 때마다 우변의 식이 결정되는 하드웨어적인 특성을 가지고 있다.

예를 들어)

wire val\_1;

assign val\_1 = val\_2 & val\_3;

의 경우 우변의 값(val\_2 & val\_3)에 변화가 발생했을 때마다 좌변의 객체(val\_1)의 값이 결정된다.

\* assign문을 이용해 net형 객체에 특정 논리값을 지정하는데 사용한다.

연속 할당문과 달리 절차형 할당문의 경우에는 우변수식의 변화와는 상관없이 문장의 실행에 의해 좌변의 변수 값을 갱신되는 소프트웨어적 특성을 가진다.

또한 절차형 할당문의 경우 다시 Blocking Statement, Non-Blocking Statement 두가지의 종류로 나눠지게 된다.

**2.**

Blocking 할당문은 기호로 =을 사용하며, begin부터 end까지 각각의 할당문의 우변을 즉각적으로 확인하고 좌변의 값을 변경하게 된다. 그와 반대로 Non-Blocking 할당문의 경우 <=를 기호로 사용하며, begin부터 end까지 모든 할당문의 우변을 확인한 후 모든 할당문의 좌변의 값을 ‘동시에’ 변경하게 된다.

-Verilog코드의 한 부분 <a=0, b=1>

always @(posedge clk) begin

a<=b;

b<=a;

end

always @(posedge clk) begin

a=b;

b=a;

end

Blocking Non-Blocking

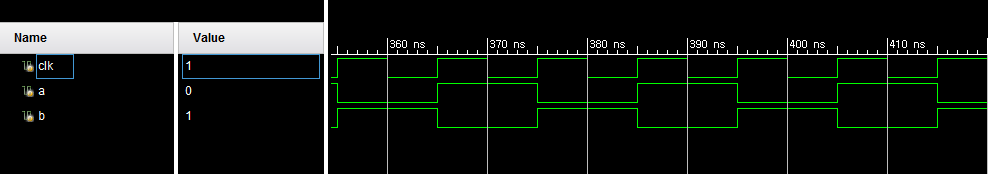
Blocking의 경우 a=b; 문장에서 a가 1로 바뀌고, b=a;에서 그대로 값이 변화가 없어

end에 왔을 때 a=1, b=1로 끝나게 된다.

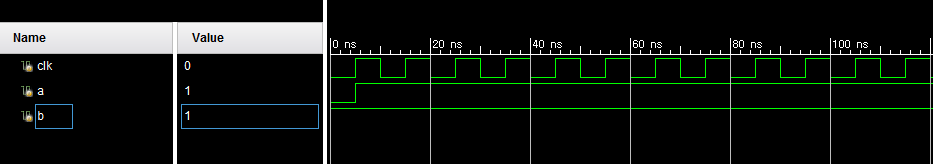
하지만 Non-Blocking의 경우 a가 b값이 됨과 동시에 b가 a값이 되어

end에 왔을 때 a=1, b=0으로 끝나게 된다.

-위 두 코드의 시뮬레이션 결과이다.



Non-Blocking



Blocking

**3.**

Verilog는 HDL로서 디지털 회로와 시스템을 묘사하는데 사용하는 반면에

C언어의 경우 범용 컴퓨터를 구동하기 위한 코드를 짜는데 사용하는 소프트웨어 프로그래밍 언어이다. 따라서 verilog는 C언어를 기반으로 한다.

그렇기 때문에 일정 부분에 대해서 verilog와 C언어는 유사함을 가지고 있다.

그 예로 for문, if문, while문의 경우 C언어와 많이 유사하다고 할 수 있다.

차이점은 C언어의 경우 중괄호를 통해 문자들을 엮지만 verilog의 경우 begin과 end로 엮어줘야 한다.

<switch문(C언어) = case문(verilog)>의 경우 그 차이점이 더 확실하게 드러나는데, C언어는 switch옆에 판별식을 넣으면서 문장을 시작하지만 verilog는 case옆에 판별식을 넣으면서 문장을 시작한다. 또한 각 항마다 case를 붙이지 않아도 괜찮고, 마지막에 break로 끝내지 않고 endcase로 문장을 끝낸다.

\*case문 역시 각 문장이 2줄 이상이면 begin과 end로 엮어준다.

**4.**

verilog의 자료형에는 두가지 종류가 있는데 게이트 또는 모듈 등 간의 물리적인 연결을 나타내는 net형과 절차적 할당문에서 값을 임시로 저장하기 위한 **variable 자료형**이 있다.

값을 저장하는 variable 자료형과 다르게 net값이 변할 때마다 net에 새로운 값이 전달된다.

(trireg net는 예외적인 경우이다)

-net 자료형 종류

wire: 단순한 연결을 위한 net

tri: 단순한 연결을 위한 net이지만, wire와의 차이점은 하드웨어에서 tri-state가 됨

wand: 다중 구동자를 가지며 wired-and 하드웨어 구현을 모델링하기 위해 사용

wor: 다중 구동자를 가지며 wired-or 하드웨어 구현을 모델링하기 위해 사용

triand: wand와 동일하게 다중 구동자를 가지고 있는 net, 하드웨어에서 tri-state가 됨

trior: wor과 동일하게 다중 구동자를 가지고 있는 net, 하드웨어에서 tri-state가 됨

supply0: 회로접지에 연결되며 전원 단자에서 끌어오는 선

supply1: 전원에 연결되며 ground에서 끌어오는 선

tri0: 저항성 pulldown에 의해 접지로 연결되는 net

tri1: 저항성 pullup에 의해 접지로 연결되는 net

trireg: 물리적인 net에 저장되는 전하를 모델링하는 net

-variable 자료형 종류

reg: 하드웨어 레지스터 및 저장소자들을 모델링하기 위해 사용될 수 있음

integer: 정수형 값을 취급

time: 시뮬레이션 시간을 처리하거나 저장하기 위해 사용됨

real, realtime: 실수형 값을 취급