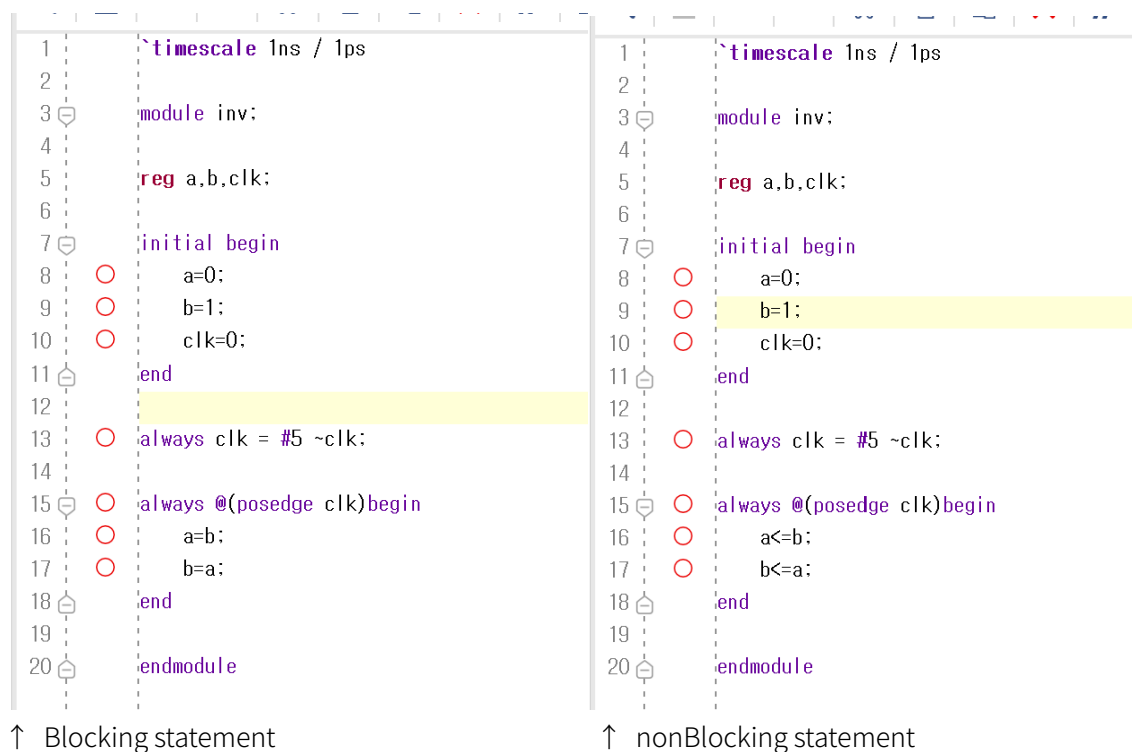


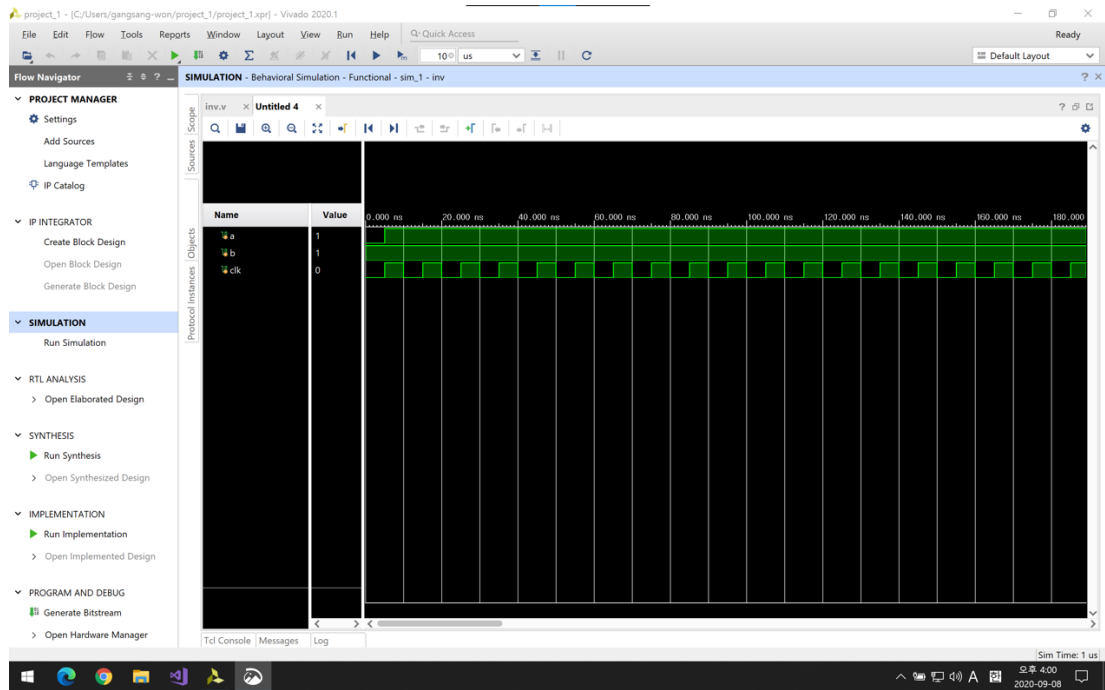
1. 연속 할당문, 절차형 할당문의 차이를 비교하여 설명하시오.

연속할당문은 assign문: ex) assign a = b(수식)이라 했을 때 b에 변화가 생길 때마다 a 값이 갱신되는 특징을 지닌다. 반면 절차형 할당문은 순서(절차)대로 문장이 실행된 후 좌변의 변수의 값이 값이 변하게 된다. 연속 할당문은 assign문을 이용해서 net형(net형 자료형: 3번 질문에서 서술)에 값을 집어넣지만, 절차형 할당문은 initial, always 구문을 이용해서 메모리 안의 변수나 reg에 값을 집어넣는다. 또한 절차형 할당문은 Blocking 할당문과 Non blocking 할당문으로 구분되는데, 이 둘의 차이는 2번 질문에서 서술하겠다.

2. Blocking 및 nonBlocking 문법의 차이를 simulation을 통해 설명하시오.

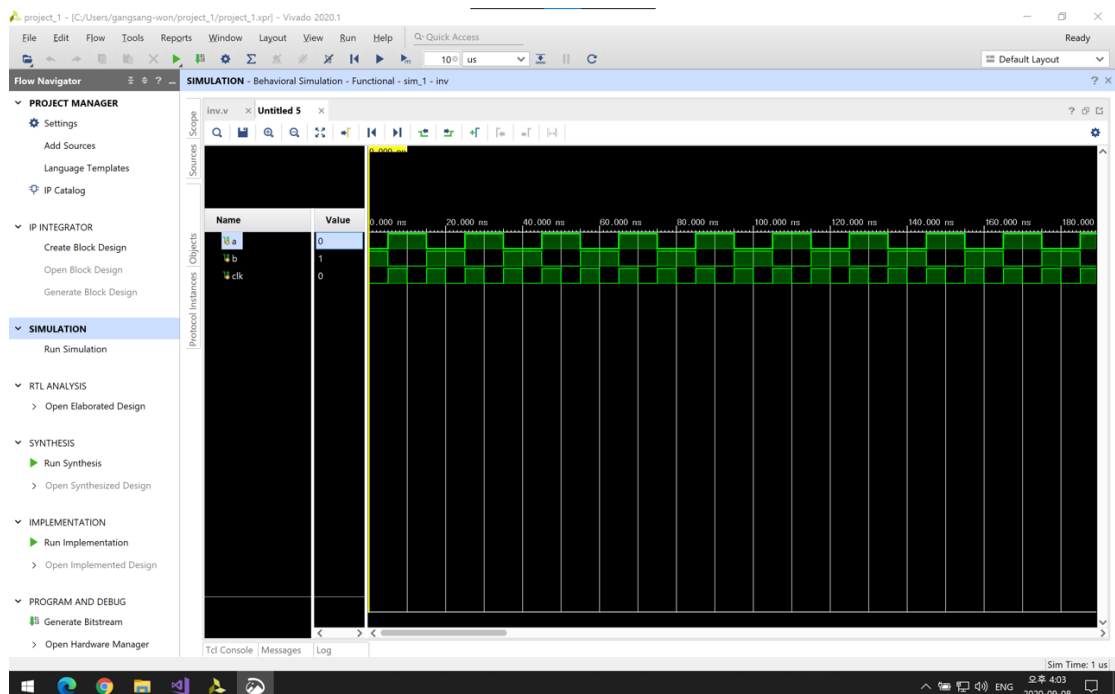


우선 위 코드에서의 차이를 보면, Blocking symbol은 “=” 이고 nonBlocking symbol은 “<=”이다. (16, 17번째 줄)



↑ Blocking statement

Blocking statement는 (일반적인 C 프로그래밍처럼) 한줄씩 계산하며 값에 변화가 생기기 때문에 a가 0에서 1이 된 이후 쪽 가게 된다. b값 할당은 a값의 변화 이후에 이루어지게 되는데, 결과적으로 b = a = 1이 된다. -> b도 값 1로 쪽 이어진다.



↑ nonBlocking statement

nonBlocking statement는 (일반적인 C 프로그래밍과는 다르게) 연산들을 모두 수행한 이후 값 할당이 이루어져 Blocking statement처럼 b = a = 1과 같은 일이 일어나지 않는다. -> 결론적으로 a, b 값이 계속 번갈아가면서 0, 1을 반복하게 된다.

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

기본적으로 C언어의 문법과 상당히 유사하다.

- for문

Verilog	C
for(i = 0; i < 10; i = i + 1) begin ... end	for(i = 0; i < 10; i = i + 1) { ... }
차이점: Verilog의 for문에서는 begin, end로 for문 안의 시작과 끝을 표시한다.	

- if문

Verilog	C
if (조건) begin ... end	if (조건) { ... }
차이점: Verilog의 if문에서는 begin, end로 if문 안의 시작과 끝을 표시한다. ※ 조건문 안의 수식이 한 줄이라면 begin, end를 작성할 필요가 없다.	

- if - else if - else문

Verilog	C
if (조건) begin ... end else if (조건) begin ... end else begin ... end	if (조건) { ... } else if { ... } else { ... }
마찬가지로 각 if, else if, else 구분에 begin, end를 사용한다.	

- while문

Verilog	C
while (조건) begin ... end	while (조건) { ... }
C언어와 마찬가지로 조건 안이 거짓이 될 때까지 안의 수식을 반복 수행한다.	

- case문

Verilog	C
<pre> case (option) opt1: a = 10; opt2: a = 30; opt3: a = 50; opt4: begin a = 50; b = 2; end default: a = 1; endcase </pre>	<pre> case (option) { opt1: a = 10; break; opt2: a = 30; break; opt3: a = 50; break; opt4 : a = 50; b = 2; break; default: a = 1; } </pre>
<p>차이점: Verilog의 case문에서는 분기마다 break;를 작성해주지 않아도 된다. case문의 끝에는 endcase를 쓴다. 이외에 한 case에 여러 줄의 수식이 필요할 경우 begin과 end를 쓴다.</p>	

4. Verilog의 net형 자료형에 대해서 조사하시오.

Module, 논리 게이트 등의 하드웨어적인 부분들 사이의 (구조적)연결을 표현하기 위해 사용되는 자료형이다. 1번 질문에서 살펴보았듯이 연속할당문에 의해 값이 할당된다. 초기값으로 z를 가진다.

다음은 net 자료형의 분류이다.

자료형 분류	쓰임새	기능
wire	단순한 연결 위한 자료형	선 연결에 사용
tri	단순 연결, 3상태 부분에서 wire와 차이	
tri0	Resistive pulldown에 의해 연결	초기값(z)이 각각 0, 1
tri1	Resistive pullup에 의해 연결	
trior	다중 구동자, 3상태 가짐	OR 연산
triand		AND 연산
trireg	전하 모델링에 사용	사용 전에는 과거 값 유지

wor	다중 구동자, 3상태 가짐	OR 연산
wand	각각 wired-or, wired-and 구조화에 쓰임	AND 연산
supply0	Circuit ground에 연결	전압공급의 세기가 각각 0, 1
supply1	Power supply에 연결	