2주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231561 이름: 심소현

**1.**

.................

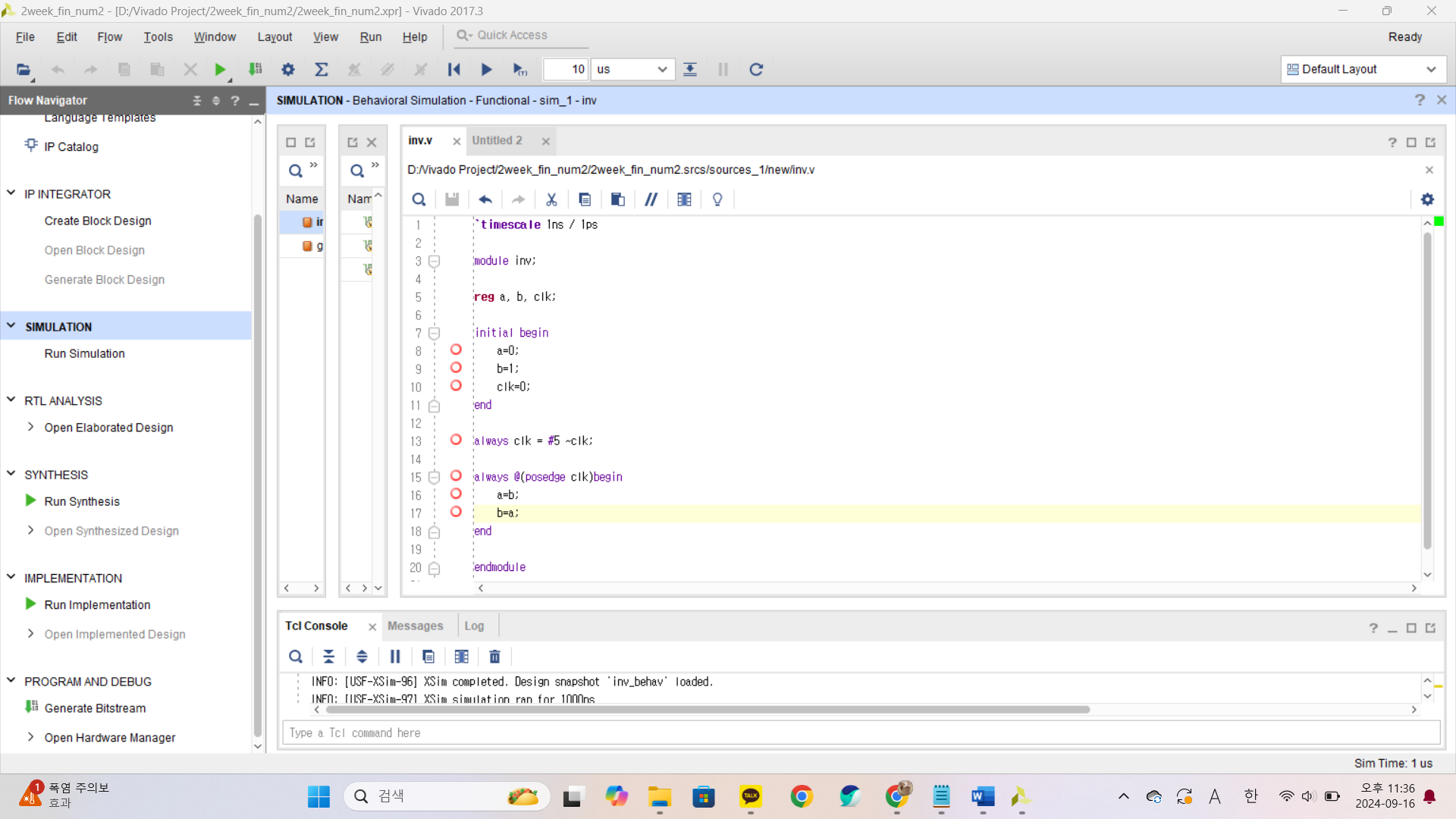
연속 할당문은 입력 피연산자 값에 변화가 일어날 때 마다 우변의 식이 평가되고 이에 따라 net 값을 지정하게 된다. 우변의 값에 변화가 발생할 때 좌변의 객체인 net 객체에 값을 할당하는 것이라고 볼 수 있다. assign문을 사용한다. assign 뒤에 지연 연산자인 “#”을 사용하여 net 할당을 지연시키는 것도 가능하다. 연사자를 사용하는 단순한 논리 표현을 이용하는 회로 모델링에 쓴다.

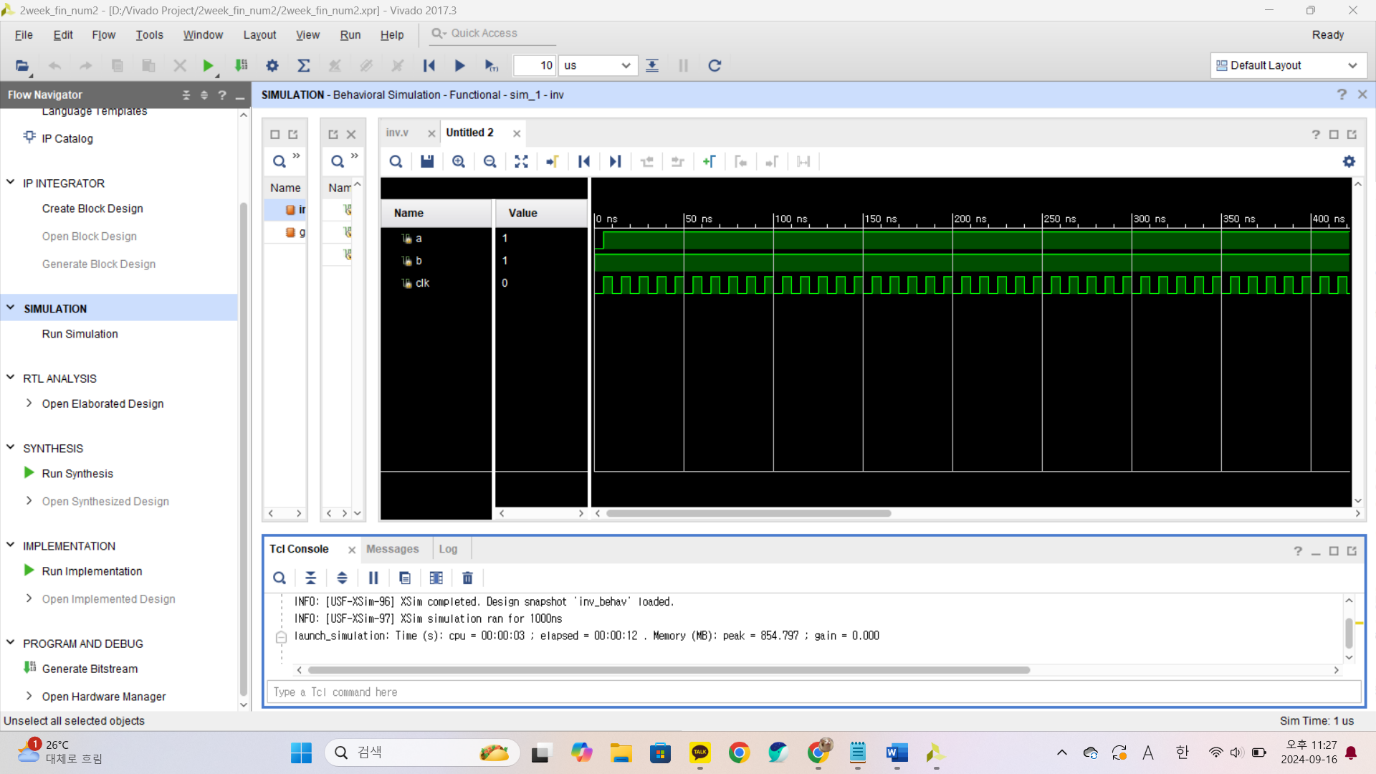
반면에 절차형 할당문은 variable 값을 지정한다고 할 수 있는데, 프로그램 코드 문장이 작성된 순서대로 실행되어 값을 할당한다. 문장의 실행에 의하여 좌변의 변수 값이 할당되는 것이기 때문에 연속 할당문과 달리 우변 수식의 변화와는 무관하다. 다만 문장의 순서가 영향을 끼치게 된다. always 블록, initial 블록, task, function 내부 할당문을 사용한다. 이때는 지연을 할 수 없고 다음 문장에 의하여만 이후 variable 값에 영향을 미칠 수 있다.

................

**2.**

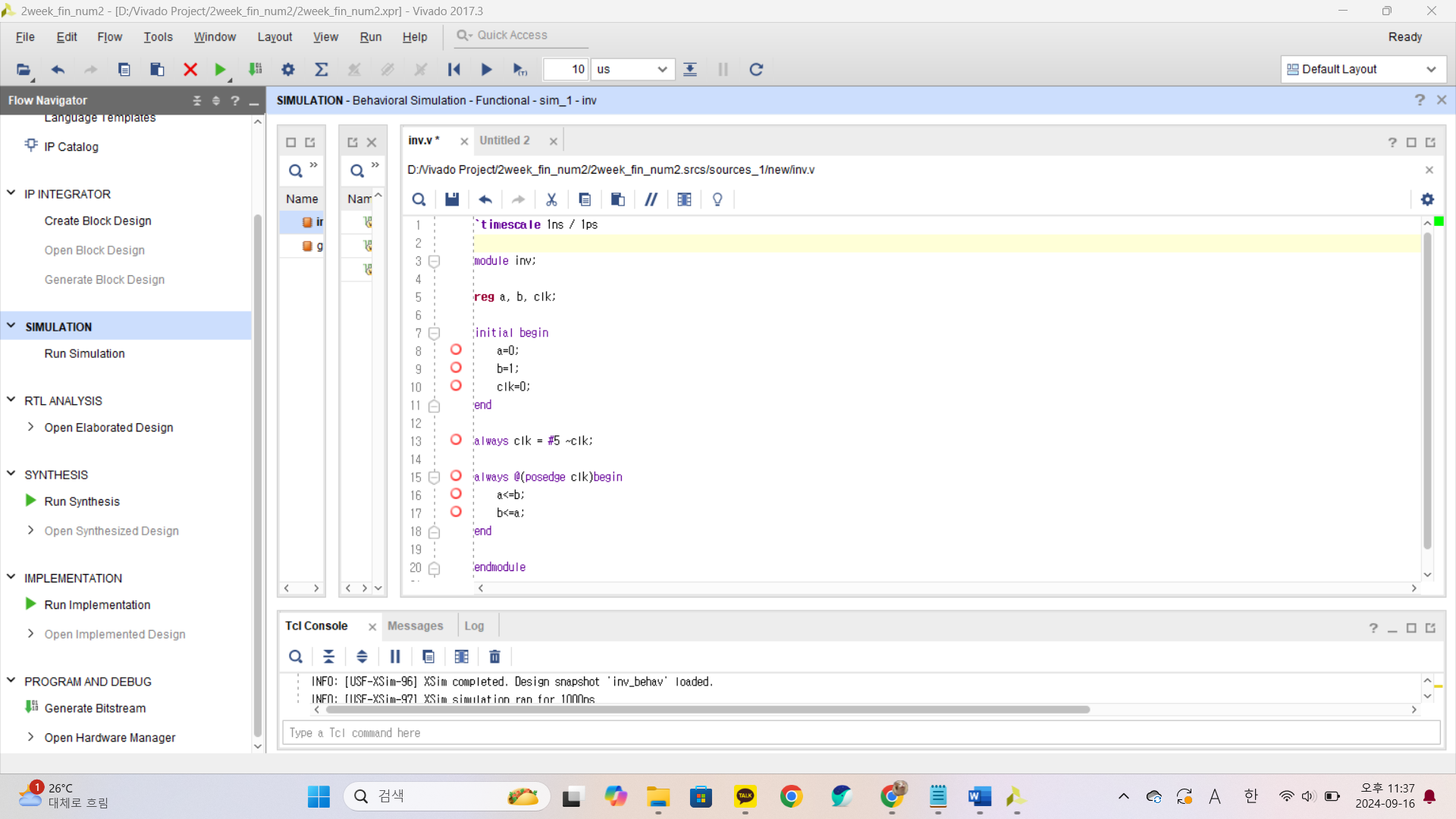
.......................

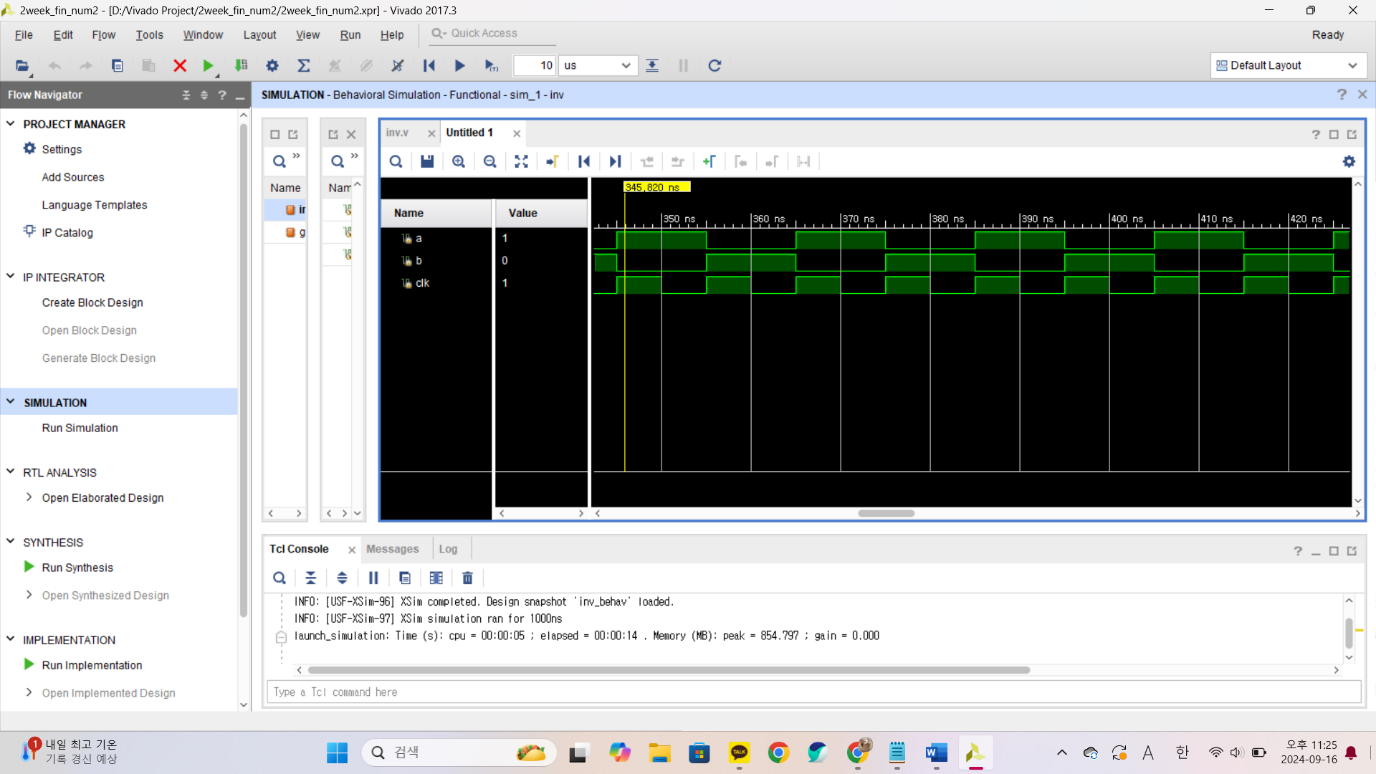
 block



blocking simulation

위의 simulation은 Blocking의 경우이다. Blocking에서 사용하는 연산자는 “=”인데, 문장 마다 계산이 즉각적으로 시행되며 이에 따라 바로 할당까지 일어난다는 것을 알 수 있다. 초기에 설정한 a의 값과 b의 값을 사용해서 always 구문 안에서 a 할당과 b 할당을 끝내 Block이 된다.

 nonblock



nonblocking simulation

위의 simulation은 nonBlocking의 경우이다. 이때 사용하는 연산자는 “<=”이다. 이때는 바로 할당이 이루어지는 것이 아니라 구문 안의 모든 계산을 끝낸 뒤에 할당이 시작된다. 사진을 보면 알 수 있듯이 설정된 a와 b의 값을 앞에서 읽고 “a<=b”와 “b<=a” 계산을 우선 수행한 뒤에 각 a와 b의 값을 할당하는 것을 알 수 있다.

따라서 Blocking과 nonBlocking의 차이는 사용하는 연산자와 문장이 계산되어 할당되는 순서임을 확인할 수 있다.

.........................

**3.**

.......................

Verilog에서 사용하는 문법은 대체적으로 C언어와 유사하다. 우선 for문을 사용하는 방법은 for(초기;조건;할당)이다. 다만 이때 for문 내부를 정하는 기호로 “{~}”를 사용하는 것이 아니라 “begin~end”를 사용한다.

Verilog에서 if문을 사용하는 것도 C언어와 비슷하다. 단독으로 사용할 때는 if(조건) begin~end 로 사용한다. 만약 다른 조건이 필요할 때 사용하는 것도 C언어와 동일하게 “else if”와 “else”이다. if(조건) begin~end else begin~end로 사용할 수 있고, if(조건) begin~end else if(조건) begin~end else begin~end 와 같이 사용하는 것도 가능하다.

while문 또한 C언어와 사용법이 같다. while(조건)을 하여 조건이 거짓이 될 때까지 반복이 일어난다. 따라서 while(조건) begin~end로 Verilog에서 while문을 구현할 수 있다.

case문을 사용하는 것도 위와 동일하게 C언어처럼 사용하면 된다. case(항목) ~:~ ~:~ ~:begin~end default ~ endcase 와 같이 사용한다. 이것도 C언어에서의 case 문법과 동일하게 사용하면 되며 유의할 점은 “{~}” 대신 “begin~end”를 사용해야 한다는 점이다.

.........................

**4.**

.......................

Verilog에서 net 자료형은 logic gate, module과 같은 하드웨어 연결을 표현하기 위하여 사용한다. 하드웨어 연결을 나타내기 위한 것이므로 net 자료형은 자체적으로 특정한 값을 저장하는 역할을 가지지는 않는다. 대표적으로 “wire”가 이 net 자료형에 포함되며, “wire”는 단순한 연결을 위한 net이다. wire은 브레드보드에서 두 구성 요소를 연결하기 위하여 사용하는 전선과 비슷한 역할을 가지고 있다. 이와 달리 의미를 가지는 자료형도 있는데, “wand”는 “wired-and”를 뜻하며 “wor”는 “wired-or”을 뜻한다. net 자료형을 사용하여 변수명을 지정할 때 이미 선언된 이름을 다시 선언하는 것은 불가능하다.

.........................