2주차 결과보고서

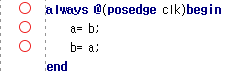
|  |
| --- |
| - blocking & non-blocking 문법의 차이점  - Inverter 구현 |

20141196 김성희

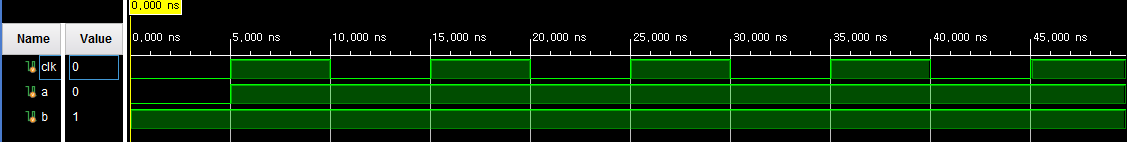
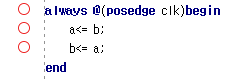
1. blocking & non-blocking 문법의 차이점

-기본적으로 blocking 문법에서는 =을 non-blocking 문법에서는 <=을 쓴다.

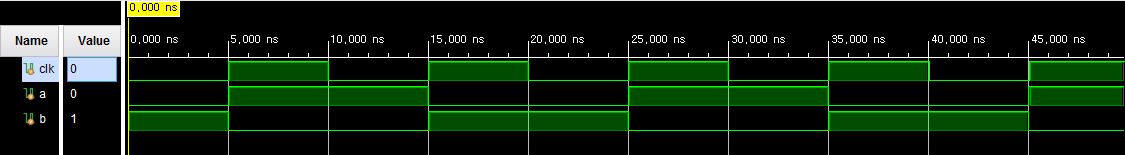
-blocking 문법은 ;으로 구분된 한 줄 한 줄을 순서대로 실행하고 non-blocking 분법은 ;으로 구분된 줄 들의 연산 부분을 모조리 수행한 후에 동시에 할당한다.

다음은 **simulation** 사진이다.

**-blocking 문법**



**-non-blocking문법**

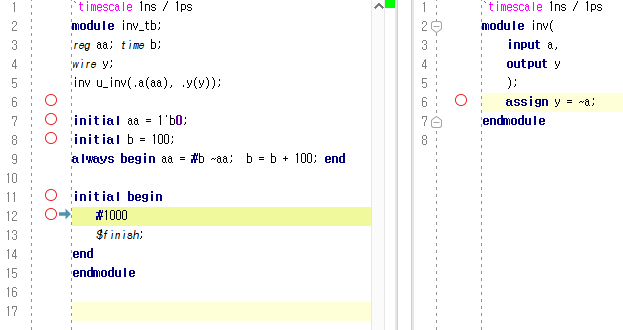


위 두 simulation사진의 코드는 두 코드 사진을 제외하곤 모두 같다. 그럼에도 이와 같은 차이가 나는 이유는 non-blocking 문법의 경우 a와 b를 동시에 바꾸지만 blocking 문법의 경우 a를 바꾸고 난 후에 b를 바꾸기 때문에 이미 바뀌어 버린 a값을 b에 넣게 다. 즉 a의 값도 b의 값도 결국 b의 값으로 고정되어 버린다. 이처럼 동시에 무언가를 바꿔야 는 경우 non-blocking 문법을 사용하는 것이 좋다.

**2. Inverter의 구현**

**-코드**

**Inv\_tb.v Inv.v**

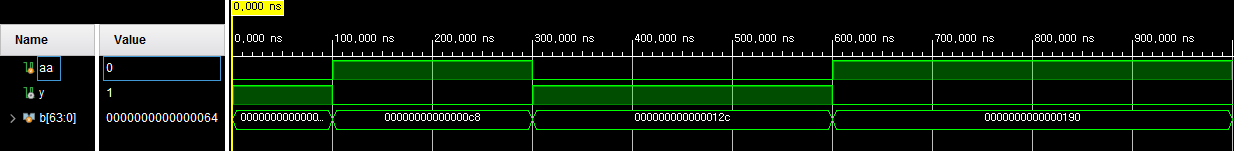


Inv\_tb.v에서 aa를 inv.v의 a(input)에 y를 y(output)에 연동 후 b나노초마다 aa값을 바꾼다. 즉 b나노초마다 input 값을 바꾸고 output에서 ~input을 할당한다.

**-스케메틱 & 진리표**



**-시뮬레이션**

****

첫 번째 value가 aa, 두 번째 value가 y, 세 번째 value가 b다.

b나노초마다 aa의 값이 바뀌고 y의 값에 ~aa가 할당되는 것을 볼 수 있다.