

# VHDL - TestBench

---

2015.07.20

23기 백두현

# 목차

---

- 조합회로 vs 순차회로
- signal vs variable
- TestBench 문법
  - 대기문
  - 지연문
  - 반복문

# 조합회로 vs 순차회로

## Combinational vs Sequential

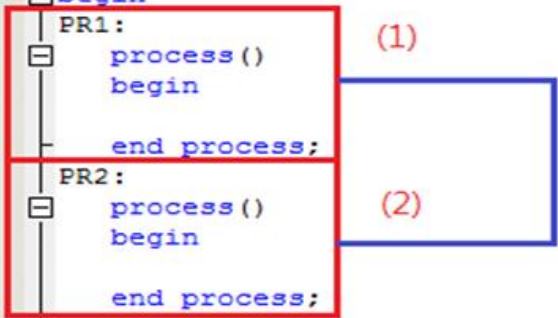


\*Flip-Flop을 포함하고 있는 회로가 무조건 순차적인 것은 아니다. Ex) RAM

# 조합회로 vs 순차회로

## 병행 코드 VS 순차 코드

```
3
4 entity my_project is
5   port (
6
7   );
8   end my_project;
9
10 architecture arc of my_project is
11   begin
12     PR1:
13     process ()
14     begin
15
16     end process;
17     PR2:
18     process ()
19     begin
20
21     end process;
22   end
```



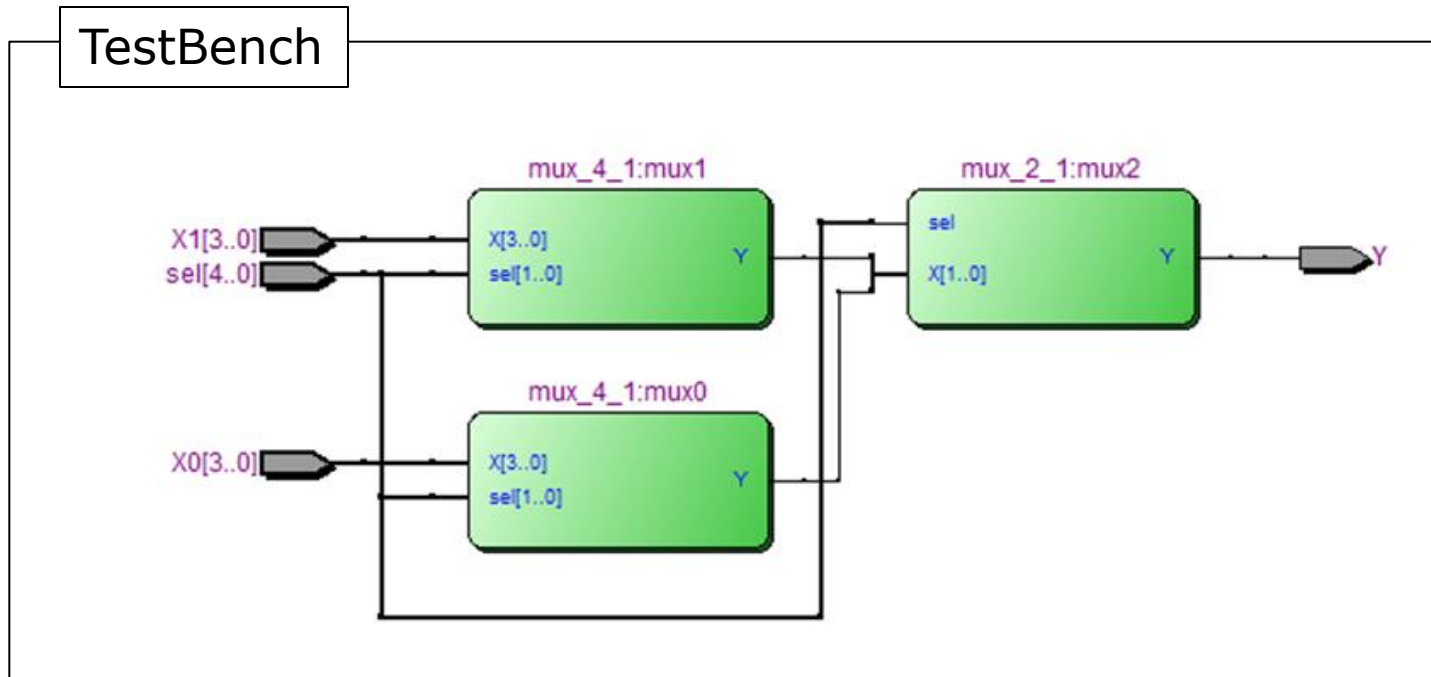
- PR1과 PR2는 병행 관계
- 각각의 Process문은 순차처리문이다.

# 조합회로 vs 순차회로

---

- 조합회로와 순차회로를 구분하는 가장 큰 기준은 메모리의 유무이다.
- 조합회로를 설계함에 있어 주의사항! 원치 않는 메모리(latch)가 발생하지 않도록 주의!
- VHDL에서 메모리가 발생할 수 있는 상황  
-> 순차구문으로 기술했을 때 메모리가 발생!

# TestBench



## □ TestBench

- 시뮬레이션을 위한 모듈.
- 설계한 모듈을 검증하기 위한 목적으로 사용.
- 모듈을 내포하여 입력 값을 생성하고, 출력 값을 관찰

# TestBench - 대기문

---

## □ 대기문(wait)

- Process 내에서 기술된 wait문의 조건이 만족될 때 까지 대기후 만족 시 진행한다.
- 대기문의 종류

### □ wait on

```
process
    wait on A,B,C;    --A,B,C 중 하나가 변할 때 까지 기다린 다음 진행.
end process;
```

### □ wait for

```
process
    wait for 10ns;    --10ns 대기 후 동작.
end process;
```

### □ wait until

```
process
    wait until A = B; --A = B가 TRUE이면 진행, False면 대기
end process;
```

# TestBench - 지연문

## □ 지연문

- 시뮬레이션 시 게이트 지연시간을 고려하기 위하여 사용.

### □ 전달지연

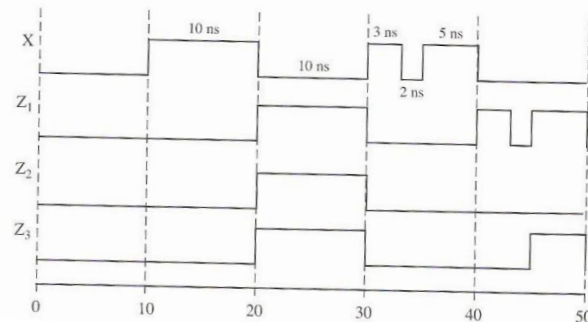
```
z1 <= transport X after 10ns;    -- 전달지연
```

### □ 관성지연

```
z2 <= X after 10ns;             -- 관성지연
```

### □ 차단 펄스 폭이 명시된 지연

```
z3 <= reject 4ns X after 10ns;  -- 차단 펄스폭이 명시된 지연
```





# TestBench - 반복문

---

## □ 무한루프

```
loop  
    sequential statements  
end loop;
```

```
exit;  
  
exit when condition;
```

## □ For문

```
for (loop-index) in (range) loop  
    sequential statements  
end loop;
```

- Loop-index는 loop에 들어갈 때 자동으로 정의
- Loop-index는 range의 첫 번째 값으로 초기화.

## □ While문

```
while (조건) loop  
    sequential statements  
end loop;
```