

# Process Statement

**The PROCESS statement encloses a set of *sequentially executed* statements. Statements within the process are executed in the order they are written. However, when viewed from the “outside” from the “outside”, a process is a single concurrent statement.**

## Format:

```
label:
PROCESS (sensitivity_list) IS
  --declarative statements
  BEGIN
    --
    --sequential activity statements
    --only sequential statements go in here
    --
  END PROCESS [label];
```

## Example:

```
ARCHITECTURE example OF nand_gate IS
  BEGIN
    nand_gate: PROCESS (a,b)
      BEGIN
        IF a = '1' AND b = '1' THEN
          z <= '0';
        ELSE
          z <= '1';
        END IF;
      END PROCESS nand_gate;
```

**Why use a process? Some behavior is easier and more natural to describe in a sequential manner. The next state decoder in a state machine is an example.**

# Process Sensitivity List

The process *sensitivity list* lists the signals that will cause the process statement to be executed.

Any transition on *any* of the signals in the signal sensitivity list will cause the process to execute.

## Example:

```
ARCHITECTURE example OF nand_gate IS
  BEGIN
    bozo: PROCESS (a,b)
      -- wake up process if a and/or b changes
      BEGIN
        IF a = '1' AND b = '1' THEN
          z <= '0' ;
        ELSE
          z <= '1' ;
        END IF ;
      END PROCESS bozo ;
    END example ;
```

## Signals to put in the sensitivity list:

- Signals on the right hand side of assignment statements.
- Signals used in conditional expressions

**What happens if a signal is left out of the sensitivity list?  
What does the synthesis tool do with the sensitivity list?**

Avoid problems with sensitivity list omissions by compiling with “synthesis check” on. Like this:

```
vcom -93 -check_synthesis test.vhd
```