VHDL Additional Simulation Constructs

- time
  - used to create constants

```
constant name:   time   :=   time   units;
constant PER:  time  :=  20 ns;
```

the space is required

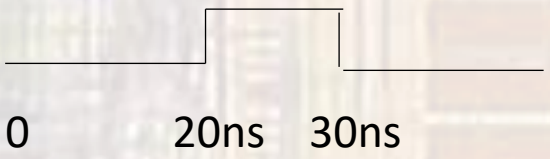
  - used to cause the simulation to stop the current process for a specified time
    - Does not affect other processes
    - Used with the wait for construct

```
wait for 45 ns;
wait for PER * 3;
wait;       // wait forever
```

use relative timing

  - used to specify absolute time in a single concurrent statement
    - Used with the after construct

```
foo  <= '1' after 20 ns;
fool <= '0' after 30 ns;
```



VHDL Additional Simulation Constructs

- for-loop
  - Used to increment an implied integer from a starting value to an ending value
    - Used in processes
    - Requires some delay in the loop to make sense

```
opt_label: for implied_int in start_val to end_val loop
...
end loop;
```

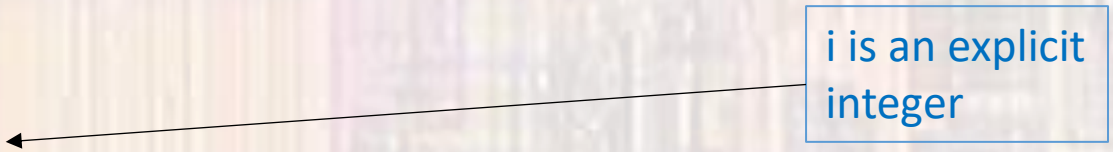
```
in_sigs: for i in 0 to 15 loop
   foo <= foo + 1;              // assumes foo can be added to
   boo <= boo + (3 * i);        // assumes boo can be added to
   val <= std_logic_vector(to_unsigned(i, 4));
   wait for 10 ns;
end loop;
```

i is an implied integer

VHDL Additional Simulation Constructs

- while-loop
  - Used to test an explicit variable for a specific condition
    - Used in processes
    - Requires some delay in the loop to make sense

```
opt_label: while explicit_var test test_val loop
...
end loop;
```




```
    variable i:  integer  :=  0; // declaration – before begin
begin                                // part of the process
    run_til: while i < 32 loop
        foo <= foo + 1;              // assumes foo can be added to
        boo <= boo + (3 * i);        // assumes boo can be added to
        val <= std_logic_vector(to_unsigned(i, 5));
        wait for 10 ns;
    end loop;
```