Delay

Note that so far we haven’t mentioned delay. Why not?

Both propagation delay and wiring delay is a real-world problem that must be eventually dealt with. However, at the model creation stage, it is helpful to not have to consider delay. Instead, the emphasis is to create correct functional behavior.

However, this does not mean the designer can go about designing with no concern about delay. When writing HDL code, you must have a very good idea what the structure you are creating will look like in a schematic sense. Otherwise, the synthesized circuit may have excessive delays, preventing its operation at the desired speed.

VHDL does have statements for representing several different kinds of delay. However, when describing a circuit to be synthesized, we never use them because the synthesis tool ignores them on purpose.

The aspect of delay is added to a synthesized netlist after the functionality has been proven correct. When real delays are inserted into your design (this is done automatically) often a whole world of problems crop up.

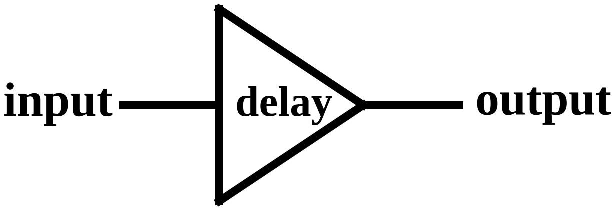
The basic idea is to make a model work, and then make it work at the desired speed. Only experience will help you determine how fast your HDL code will eventually run.

Delay Types

VHDL signal assignment statements prescribe an amount of time that must transpire before a signal assumes its new value.

This prescribed delay can be in one of three forms:

- **Transport:**
  propagation delay only
- **Inertial:**
  minimum input pulse width and propagation delay
- **Delta:**
  the default if no delay time is explicitly specified



Signal assignment is actually a *scheduling* for a future value to be placed on the signal.

Signals maintain their original value until the time for the scheduled update to occur.

<u>Any</u> signal assignment will incur a delay of one of the three types above.

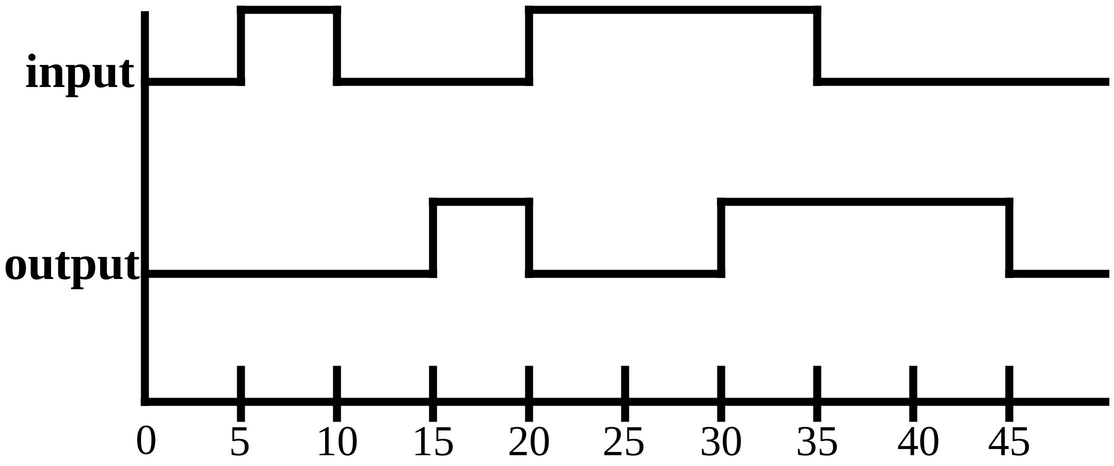
Delay Types - Transport

Delay must be explicitly specified by the user by the keyword TRANSPORT.

The signal will assume the new value after specified delay.

Example:

```
output <= TRANSPORT buffer(input) AFTER 10ns;
```



Transport delay is like a infinite bandwidth transmission line.

Delay Types - Inertial

Inertial delay is the default in VHDL statements which contain the “AFTER” clause.

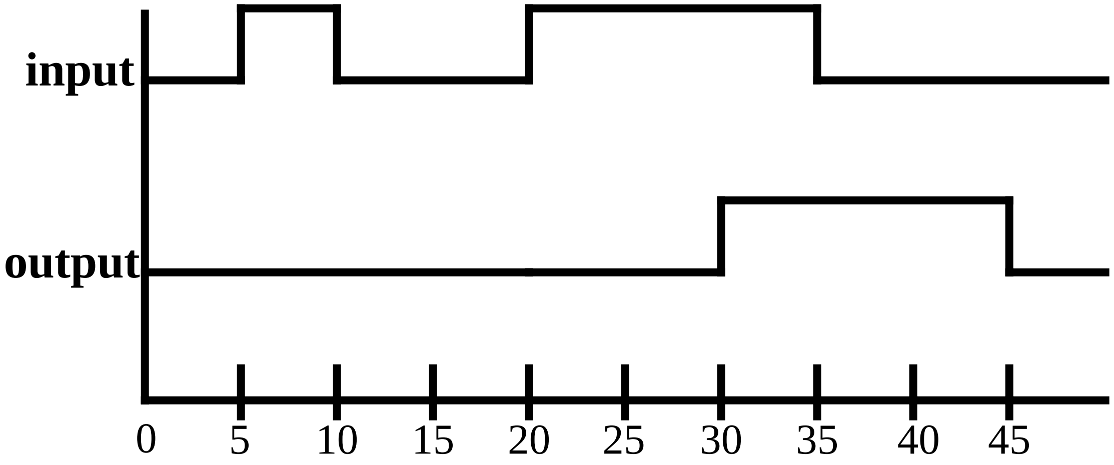
Inertial delay provides for specification of input pulse width, i.e. ‘inertia’ of output, and propagation delay.

Format:

```
target <= [REJECT time_expr] INERTIAL waveform
AFTER time
```

Example (most common):

```
output <= buffer(input) AFTER 10ns;
```



When not used, the REJECT clause defaults to the value of the AFTER clause.

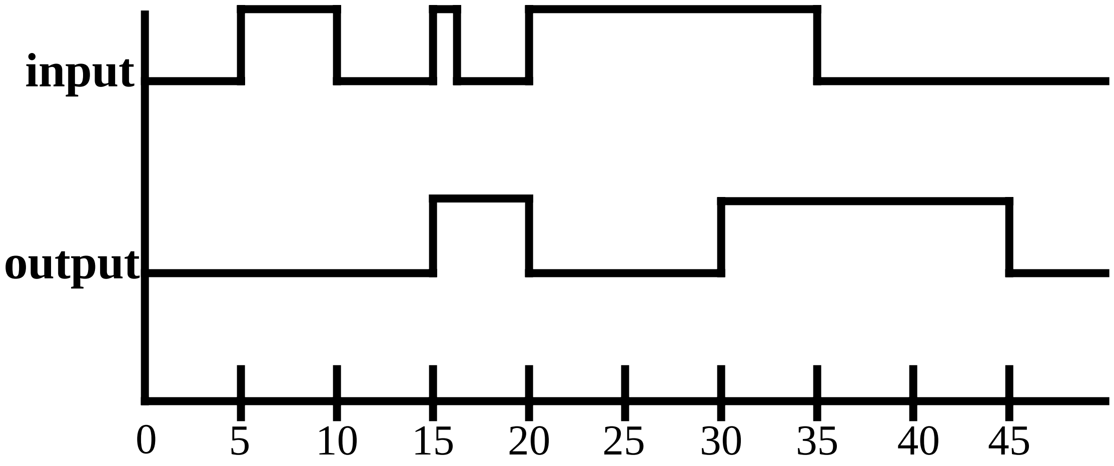
Inertial delay acts like a real gate. It “eats” pulses narrower in width than the propagation delay.

Delay Types - Inertial

Example of gate with "inertia" smaller than propagation delay:

This shows a buffer that has a prop delay of 10ns, but passes pulses greater than 5ns.

```
output <= REJECT 5ns INERTIAL buffer(input)
AFTER 10ns;
```



REJECT can be used only with the keyword INERTIAL.

Delay Types - Delta Delay

***Delta delay* is the signal assignment propagation delay if none is explicitly prescribed.**

A delta time is an infinitesimal, but quantized unit of time.

An <u>infinite</u> number of delta times equals <u>zero</u> simulator time.

The delta delay mechanism provides a minimum delay so that the simulation cycle can operate correctly when no delays are stated explicitly. That is:

- **all active processes to execute in the same simulation cycle**
- **each active process will suspend at some *wait* statement**
- **when all processes are suspended, simulation is advanced the minimum time step necessary so that some signals can take on their new values**
- **processes then determine if the new signal values satisfy the conditions to proceed again from the wait condition**