Delta Delay

• An infinitesimally small delay.
• Not a real delay. Simulation time does not advance.
• Allows for ordering of events.
• Why is it needed?
  – Real signals never change instantaneously.
  – Describe hardware without ambiguities of zero-delay models. Ex. cross coupled latch.
Delta delay example

entity RS_latch is
  port (R, S : in bit := '1'; Q : buffer bit := '1'; Qbar : buffer bit);
end RS_latch;

architecture delta of RS_latch is
begin
  Qbar <= R nand Q;
  Q <= S nand Qbar;
end delta;

configuration RS_cfg of RS_latch is
for delta
end for;
end RS_cfg;
Entity - Port modes

**in**: right side of signal or variable assignment.

**out**: left side of signal assignment.

**inout**: both of above (for bidirectional signals)

**buffer**: similar to inout, but can only have 1 source. (Better - use internal signal for feedback; assign it to out mode port.)
Simulation - Initialization phase

1. Each signal is given an initial value. Explicitly or implicitly defined. (I.e. default for bit = ‘0’, std_logic = ‘U’)

2. Each process is executed until it suspends. Sensitivity list ignored first time. Each concurrent statement is executed.
Simulation cycle

- event-driven: signal changing value = event
- simulation time advances to next event. (I.e. scheduled signal change)
- Scheduled signal changes take place.
- All processes (and concurrent signals) sensitive to those signals execute until they reach a wait statement. Future events scheduled. Cycle repeats until no more events
Signals Assignment vs. Variables Assignment
from p. 5-10 Mazor, A Guide to VHDL

- Signal values are scheduled.
- Signal values are updated only after wait is executed (can be implicit wait). I.e. updated in next simulation cycle.
- Signal assignments can have a delay.

- Variable values are not scheduled.
- Variable values are updated immediately.
- Variable assignment are specified without a delay.
Signals vs. Variables

- VHDL - fundamentally a collection of processes communicating through signals.
- Signals - similar to wires in circuit.
- Variables - temporary storage / intermediate value within process
VHDL Functions

- Execute in zero time. (cannot have wait)
- Access only locally declared objects.
- Cannot directly update signals.
- Returns a single value.
- Parameters of **input** mode.
- Parameters passed as constant by default.
Using Functions

process begin
loop1: for j in 1 to 9 loop
   X <= Xarray(j);
   Y <= Yarray(j);
   Z <= add4 (X, Y, ‘0’);
   wait for 50 ns;    -- Will this process
end loop loop1;    -- work as expected??
end process;
Solutions

• Put $Z \leftarrow \text{add4} (X, Y, ‘0’) \text{ outside process.}$
  (I.e. make it a concurrent statement)
  Example - hex_7seg use in updown counter.

• Put a wait for 0 ns after assignment to $Y$.

• Pass variables to add4 rather than signals.
  No delta delay.
Signal Drivers

- Every signal assignment creates a signal driver.
- Each process creates only one driver or source for a signal, regardless of the number of times the signal is assigned within the process.
- A signal with more than one driver (source) must have a resolution function associated with it; Otherwise it is an error.