

VHDL Modeling Guidelines (*for synthesis*)

Two process model:

- 1) Synchronous process – single-clock, single-edge (or single active value with modeling latches)
 - a. Minimize asynchronous operations of associated with flip-flops
 - i. Reset/clear
 - ii. Set/preset
 - b. Focus on synchronous operation of flip-flops
 - i. Reset/clear
 - ii. Set/preset
 - iii. Clock enable, load
 - iv. Simple counting and/or shifting operations (ie, count enable, shift/load) - keep it simple, remember you can always partition out complicated combinational logic as in the Mealy and Moore models
 - c. Assign only those signals representing flip-flops
- 2) Combinational logic process –include all dependencies in sensitivity list
 - a. Partition logic functions and focus on one at a time
 - i. Use one type of conditional construct for that logic
 1. if-then-else
 2. case-when
 - ii. Completely specify for all conditions
 - iii. Assign don't cares whenever possible (and legitimate)
 - b. Assign only those signals representing the outputs of combinational logic functions (do not assign any signals representing flip-flops or latches)
 - c. For complicated logic functions use multiple combinational processes

Note: Assign any given signal in one and only one process – do not assign values to the same signal in multiple processes!

Example:

```
entity BLAHBLAH is
    generic (...);
    port (...);
end entity BLAHBLAH;
architecture TWOMOD of BLAHBLAH is
    -- signal declarations
    begin
        SYNC: process (CLK, ANY_ASYNC_INPUTS) begin
            if (ANY_ASYNC_INPUTS = 'ACTIVE_VALUE') then
                -- prioritized asynchronous operation
            elsif (CLK'event and CLK = 'ACTIVE_EDGE') then
                -- prioritized synchronous operation
            end if;
        end process SYNC;
        COMB: process (ALL_DEPENDENCIES)
            -- variable declarations
            begin
                -- description of combinational logic
            end process COMB;
```

Exceptions to the rule:

- 1) When the use of multiple clock edges is required (ie, rising-edge for most flip-flops then falling-edge for a few input or output flip-flops), use two synchronous processes – one for each clock edge. But be careful to assign any given signal representing a flip-flop in one and only one of the synchronous processes.
- 2) You can make very simple signal assignments (with no logic or conditions) using concurrent signal assignments (ie, $Z \leq A$;). This is particularly good for primary outputs to avoid the need for *buffer* signal types. It is also good for ensuring that primary inputs and outputs meet I/O naming conventions and specifications while using desired internal signals (ie, *bit_vectors*) for more modeling.

These VHDL modeling guidelines (and similar ones) have been used with great success in industry for the past 20 years. They were actually developed before VHDL came along when Bell Labs was modeling and synthesizing ASICs in C.