

Naming Conventions

- Use *meaningful names* for identifiers. Use the same names as in the block diagrams.
- Names should indicate the purpose (e.g., `AdrCounter`), not the type (e.g., `UpCounter8`) of an object.
- Use mixed casing and less than 15 characters for *identifiers*.
- Use uniform casing for *reserved words*.
- Indicate special properties or types of names by a *suffix*.
- Use the same name for a signal throughout all levels, or attach a prefix to the name at higher levels which indicates the source/destination subcomponent of the signal (e.g., `IdlexDO` -> `CtrlIdlexD`, `InDataxDI` -> `PipeInDataxD`).
- *Rename* signals only where necessary (e.g., for changing the mode).

Proposal

- Names are composed of several 2-6 character *substrings*.
 - The substrings have the first letter in *upper case* and the remaining letters in *lower case*.
 - The substrings should be meaningful abbreviations of actions (e.g., `Sel`, `Inc`, `Add`, `Load`) and objects (e.g., `Reg`, `Mux`, `Data`, `Key`).
- The names should give an appropriate description of the *purpose* of the object.
- Names of objects with a *hardware equivalent* (e.g., signals, entities/architectures, components) start with an upper-case letter.
- Names of objects with *no hardware equivalent* (e.g., variables, types) and labels start with a lower-case letter.
- No "_" is used (except constants), particularly in port names (is reserved character in the VITAL standard).

Constant Names

- Use upper-case letters and "_" only (e.g., `WIDTH`, `RAM_DEPTH`, `LFSR_INIT`).
- Avoid "_" in generics (synthesis attaches generic names to other names with "_" as delimiter).

Signal Names

- Start with an *upper-case* letter.
- Have a suffix with syntax `"x[CRSDTA][NP]?Z?B?[IO]?"` ("`...`" denotes a choice, "?" means optional).
- The suffix part `"[CRSDTA]"` indicates the class of the signal:

Class	Char	Example	Description
clock	C	<code>ClkxC</code>	clock
reset	R	<code>RstxRB</code>	asynchronous reset
control/status	S	<code>SelInputxS</code> , <code>FullxS</code>	control and status signals
data/address	D	<code>SamplexD</code> , <code>RamAdrxD</code>	data and address signals
test	T	<code>ScanEnxT</code> , <code>RamIsolxT</code>	test signals
asynchronous	A	<code>StrobexA</code>	asynchronous signals

- The suffix part `"[NP]?"` indicates next and present state for a signal (e.g., `StatexDN` / `StatexDP`, `AddrCntxDN` / `AddrCntxDP`)
- The suffix part `"Z?"` indicates three-state signals
- The suffix part `"B?"` indicates active low signals.

- The suffix part "[IO]?" indicates input and output signals of an entity (e.g., CoeffxDI, FullxSO, ExtRamxDZIO)

Variable Names

- Start with a *lower-case* letter (e.g., temp, i, currentState).
- Have *no suffix* (as opposed to signal names).

Type Names

- Have a suffix "Type" or a name that implies a type (e.g., stateType, stdLogicArray).

Design Unit Names

Entity

Meaningful name describing the purpose of the circuit (e.g., DesEngine).

Architecture

According to the modeling style used (i.e., Behavioral, Procedural, Dataflow, or Structural) or to some specific architecture property.

Configuration

Name of the corresponding entity with suffix "Cfg" (e.g., DesEngineCfg).

Package

Name of the design (i.e., top-level entity) with suffix "Pkg" (e.g., DesEnginePkg).

Testbench (*entity, architecture, package*)

Name of corresponding entity with suffix "Tb" (e.g., DesEngineTb, DesEngineTbPkg).

Library (*package*)

Library name with suffix "Lib" (e.g., ArithLib).

Labels

- Give labels to all complex *concurrent statements*.
- Use the following *suffices*:

Construct	Suffix	Example
Process	register	Reg
	counter	Cnt
	combinational	Comb
	testbench	Tb
Block	Blk	inputPortBlk
Instance	Inst	datapathInst
Generate statement	Gen	reduceAndGen

File Names

- Have the same name as the contained design unit (possibly with the first or all letters in lower case).
- Have file suffix ".vhd" or ".vhdl".