**VHDL Reserved Words**

Following are the reserved words in the VHDL language. To verify whether a reserved word is supported in Xilinx® designs, see the [XST User Guide](https://www.xilinx.com/itp/xilinx10/books/docs/xst/xst.pdf).

**Note**The ISE® Language Templates provide prepared pieces of code and code syntax for use in your source files. For more information, see [Working with Language Templates](https://www.xilinx.com/itp/xilinx10/isehelp/ise_c_language_templates.htm).

| **Reserved Word** | **Purpose** |
| --- | --- |
| abs | Arithmetic operator for absolute value. Unary operator, predefined for any numeric type. |
| access | A variety of data type whose values are pointers to (or links to, or addresses of) dynamically-allocated objects of some other type. |
| after | Clause used to include delay information in a signal assignment. If there is no after clause, default delay of one simulation delta is assumed. |
| alias | Declares an alternate name for all or part of an existing object. |
| all | Suffix for identifying all declarations that are contained within the package or library denoted by the prefix. |
| and | Logical operator for types bit and Boolean and for one-dimensional arrays of these types. |
| architecture | Statement that contains description of the design. |
| array | A composite type in which all values belong to the same data type (e.g., string is an array of the data type character). |
| assert | Statement that presents a condition to be evaluated. Often used in conjunction with reporting of error messages. |
| attribute | A named characteristic of items belonging to one of the following classes: Types, subtypes Procedures, functions Signals, variables, constants Entities, architectures, configurations, packages Components Statement labels An attribute declaration declares an attribute name and its type. An attribute specification associates an attribute with a name and assigns a value to the attribute. Predefined attributes exist for types, arrays, and signals. |
| begin | Marks the beginning of the statement portion (as opposed to the declarative portion) of a process statement or architecture body. |
| block | Concurrent statement used to partition a design. |
| body | Conjoined with package. A package body stores the definitions of functions, procedures, and the complete constant declarations for any deferred constants that appear in a corresponding package declaration. The name of the package body is the same as that of the package declaration to which it refers. |
| buffer | A mode that enables a port to be read and updated within the entity model. A buffer port cannot have more than one source, and can be connected only to another buffer port or to a signal with no more than one source. |
| bus | A kind of signal that represents a hardware bus. When all drivers to the signal become disconnected, the signal’s value is determined by calling the resolution function with all the drivers off. Any previous value is lost. Bus signals may be either a port or locally declared signal. |
| case | A form of conditional control that selects statements for execution based on the value of a given expression. |
| component | Declaration made in a top level entity to instantiate lower-level entities. |
| configuration | Associates particular component instances with specific design entities, and associates entity declarations with specific architectures. |
| constant | A class of data object. Constants can hold a single value of a given type. If the value is not specified, the constant is a deferred constant, and can appear inside a package declaration only. |
| disconnect | Specifies the disconnect time for a guarded signal. |
| downto | Specifies direction in a range. |
| else | Optional clause in an if statement. An else clause specifies alternative statements when the if clause and any elsif clauses evaluate false. |
| elsif | Clause in an if statement that poses an alternative condition when the if clause evaluates to false. |
| end | Marks the end of a statement, subprogram, or declaration of a library unit. |
| entity | Specifies input and output definitions of the design. |
| exit | Causes execution to jump out of the innermost loop or the loop whose label is specified. |
| file | A category of data type. File types provide a way for a VHDL design to communicate with the host environment. File type is declared with a file type definition, while files are declared with a file declaration. |
| for | Used to iterate a predetermined number of replications in replicated logic, such as generate and loop statements. Also used in specifying blocks, components, and configurations, and in specifying time expression in a timeout clause. |
| function | A subprogram used for computing a single value. Functions are always terminated by a return statement, which returns a value. Functions are specified with a subprogram specification. |
| generate | Replicates one or more concurrent statement. Can be in for or if format. |
| generic | Passes environment information to subcomponents; can be declared in the same constructs in which ports can be declared. Generics are of the object class constant. The declaration of a generic may also include a default value, which will be used if an actual value is missing in the generic map. |
| guarded | Option for a concurrent signal assignment. The guarded option specifies that the signal assignment statement will execute only when the guard condition of the block statement that contains the assignment is true. |
| if | Conditional logic statement. Presents a condition to be evaluated as true or false. |
| impure | Option for a function in a subprogram specification. Use of this reserved word extends the scope of variables and signals declared outside of the function to be available to that function, resulting in the possibility that the function may return different values when called multiple times with the same actual parameter values. |
| in | Port mode that allows the port to be read only. If no mode is specified, in is assumed. |
| inertial | An option for specifying delay mechanism in a signal assignment statement. Inertial delay is characteristic of switching circuits: a pulse whose duration is shorter than the switching time of the circuit will not be transmitted or in the case that a pulse rejection limit is specified, a pulse whose duration is shorter than that limit will not be transmitted. |
| inout | Port mode that allows a bidirectional port to be read and updated within the entity model. |
| is | Reserved word that equates the identity portion to the definition portion of a declaration. |
| label | An entity class, to be stated during attribute specification of user-defined attributes. |
| library | A context clause that makes visible the logical names of design libraries that can be referenced within a design unit. The following library clause is implied for every design unit:  library std, work |
| linkage | A port mode similar to inout used to connect VHDL ports to non-VHDL ports. |
| literal | An entity class, to be stated during attribute specification of user-defined attributes. |
| loop | Statement used to iterate through a set of sequential statements. |
| map | With port or generic, associates port names within a block (local) to names outside a block (external). A port of mode may be left unconnected either by omitting it from the port map, or by connecting it to the reserved word open. In either case, the corresponding port declaration must include a default value. |
| mod | Arithmetic operator for modulus. Modulus is predefined for any integer type; the operands and the result are of the same type. The result of a mod operator has the sign of the second operand and is defined (for some integer n) as:  a mod b = a-b\*n |
| nand | Logical operator for types bit and Boolean and for one-dimensional arrays of these types. Complement of and. |
| new | An allocator that enables objects of a specific type to be created dynamically. These dynamically-created objects are accessed by access types. |
| next | Statement that causes the current iteration of the specified loop to be prematurely terminated, resuming execution with the next iteration of the loop. |
| nor | Logical operator for types bit and Boolean and for one-dimensional arrays of these types. Complement of or. |
| not | Unary logical operator for types bit and Boolean. |
| null | Sequential statement that causes no action to take place; execution continues with the next statement. |
| of | Reserved word used to link an identifier to its entity name, and used when specifying type mark in a file type definition. |
| on | Used to introduce the sensitivity list in the sensitivity clause of a wait statement. |
| open | An entity aspect, used as a binding indication to indicate that binding is not yet specified and that it is to be deferred. |
| or | Logical operator for types bit and Boolean and for one-dimensional arrays of these types. |
| others | When used as the last branch of case statement, used to cover all values not specified by when statements. Can also be used as part of the right-hand side of a signal or variable assignment statement for array types. This assigns values to array elements not otherwise assigned. |
| out | Port mode that enables the port to be updated only. It cannot be read. |
| package | Optional library unit for making shared definitions (usually type definitions). You must issue a use statement to make the package available to other parts of the design. |
| port | Signals through which an entity communicates with the other models in its external environment. |
| postponed | Option for a concurrent signal assignment or process statement. |
| procedure | Subprogram used to partition large behavioral descriptions. Procedures can return zero or more values. |
| process | A process represents a level of hierarchy in a design. The statements contained in the **process\_statement\_part**run sequentially (from top to bottom) rather than concurrently. If the process includes the optional **sensitivity\_list**, the **process\_statement\_part** is executed only when there is an event on one or more of the signals listed in the **sensitivity\_list**.  For simulation, the **process\_statement\_part** of all processes is executed once when the simulation initializes. Processes with sensitivity lists will not execute again until there is an event on one of the signals in the **sensitivity\_list**. Processes without a sensitivity list will continue to re-execute their **process\_statement\_part** for the remainder of the simulation. This implies that the **process\_statement\_part** should include at least one wait statement. Otherwise, the simulation time will not advance and the simulator will appear to be frozen.  For synthesis, processes may be used to infer either sequential (clocked) or combinational logic. Processes intended to infer combinational logic should include in the **sensitivity\_list** all signals that affect the behavior of the **process\_statement\_part**. This includes not only signals appearing on the right-hand side of signal or variable assignment statements, but also signals or variables appearing as part of conditional statements such as if or case.  Processes that infer synchronous logic should include the clock signal and any asynchronous controls (asynchronous resets or presets) in the sensitivity list.  In general, processes may or may not be synthesizable, depending on the details of how they are written. See the IEEE VHDL User Manual for details. |
| pure | Option for a function in a subprogram specification.  A pure function will disallow the use of any signals or variables declared outside of the function.  All functions are pure unless specified as impure. |
| range | Parameter used when specifying subtypes in an array type declaration. |
| record | A composite data type in which the collection of values may belong to the same or different types. |
| register | A kind of signal which models a latch. If all drivers to such a signal are disconnected, the signal retains its old value. |
| reject | An option for specifying delay mechanism in a signal assignment statement. Every inertially delayed signal assignment has a pulse rejection limit. If the delay mechanism specifies inertial delay, and if the reserved word reject followed by a time expression is present, then the time expression specifies the pulse rejection limit. In all other cases, the pulse rejection limit is specified by the time expression associated with the first waveform element. Not supported for synthesis. |
| rem | Arithmetic operator for remainder. Remainder is predefined for any integer type; the operands and the result are of the same type. The result of a rem operator has the sign of the first operand and is defined as:  a rem b = a-(a/b)\*b |
| report | Statement for generating report messages. Not supported for synthesis. |
| return | Statement that causes a subprogram to terminate, returning control back to the calling object. All functions must have a return statement, and the value of the expression in the return statement is returned to the calling program. For procedures, objects of mode out and inout return their values to the calling program. |
| rol | Shift operator: rotate left. Shift operators are defined for any one-dimensional array type whose element type is either bit or Boolean.  The arguments of rol are the array that will be rotated and the amount by a which it will be rotated. |
| ror | Shift operator: rotate right. Shift operators are defined for any one-dimensional array type whose element type is either bit or Boolean.  The arguments of rol are the array that will be rotated and the amount by a which it will be rotated. |
| select | Expression whose value determines different values for a target signal in a selected signal assignment statement. |
| severity | A predefined type in the language with values note, warning, error, and failure. |
| shared | An type of variable that can be declared only in entities, architectures, and generates. A shared variable can be accessed by all three of the subprograms/processes local to the declarative region. |
| signal | Represents a wire or a placeholder for a value. Signals are assigned in signal assignment statements, and declared in signal declarations. Note that signal assignments always occur with some amount of delay. In the absence of the optional delay\_mechanism, signal assignments will occur one delta delay after the signal assignment statement is executed. This fact has major implications when a signal assignment is executed as part of a block of sequential statements within a process. See the IEEE VHDL User Manual for details. |
| sla | Shift operator: shift left arithmetic. Shift operators are defined for any one-dimensional array type whose element type is either bit or Boolean.  The arguments of sla are the array that will be shifted and the amount by a which it will be shifted.  This shift operator will fill with the leftmost bit. |
| sll | Shift operator: shift left logical. Shift operators are defined for any one-dimensional array type whose element type is either bit or Boolean.  The arguments of sll are the array that will be shifted and the amount by a which it will be shifted.  This shift operator will fill with zeros. |
| sra | Shift operator: shift right arithmetic. Shift operators are defined for any one-dimensional array type whose element type is either bit or Boolean.  The arguments of sra are the array that will be shifted and the amount by a which it will be shifted.  This shift operator will fill with the rightmost bit. |
| srl | Shift operator: shift right logical. Shift operators are defined for any one-dimensional array type whose element type is either bit or Boolean.  The arguments of srl are the array that will be shifted and the amount by a which it will be shifted.  This shift operator will fill with zeros. |
| subtype | A declaration that defines a base type and a constraint. The constraint specifies a subset of values for the base type. An object is said to belong to a subtype if it is of the base type and if it satisfies the constraint. |
| then | Introduces statements to execute when the preceding if or elsif statement evaluates true. |
| to | Specifies direction in a range. |
| transport | An option for specifying delay mechanism in a signal assignment statement. Transport delay is characteristic of hardware devices, such as transmission lines, that exhibit nearly infinite frequency response: any pulse is transmitted, no matter how short its duration. Not supported for synthesis. |
| type | Data type. Each data type has a set of values and a set of operations associated with it. User-defined types are created with type declarations. Predefined types can be divided into several categories: scalar, composite, access, and file. In addition, there are non-predefined types established by the IEEE standard 1164. Each of these types is listed below.   * Scalar Types   + Enumerated   + Character (literals: 128 characters of the ASCII character set)   + Bit Boolean (literals: true, false)   + Severity\_level (literals: note, warning, error, failure)   + Numeric   + Integer   + Physical   + Floating\_point * Composite Types   + Array   + String (1-dimensional array of type character)   + Bit\_vector(1-dimensional array of type bit)   + Record   + Access Types (see access)   + File Types (see file) * Non-predefined Types: In addition to the predefined types, IEEE standard 1164 adds the following types that are commonly used for modeling digital logic:   + Std\_ulogic (an enumerated type with the values ’U’, ’X’, ’0’, ’1’, ’Z’, ’W’, ’L’, ’H’, ’-’)   + Std\_logic (same as std\_ulogic except that this is a resolved type)   + Std\_ulogic\_vector (an array of std\_ulogic)   + Std\_logic\_vector (an array of std\_logic) * Predefined-Types: Types defined by the IEEE standard 1076.3 are:   + Unsigned (an array of std\_logic)   + Signed (an array of std\_logic)   + Overloaded arithmetic and conversion operators for types unsigned and signed are defined in the package numeric\_std.   More information on selecting a data type can be found in the Design Considerations section of the IEEE VHDL Reference Manual. |
| unaffected | Concurrent statement that causes no action to take place; execution continues with the next statement. |
| units | An entity class, to be stated during attribute specification of user-defined attributes. Also used in physical type definition statement. |
| until | Part of the condition clause of a wait statement. |
| use | Clause that makes the contents of a package visible from inside an entity or an architecture. |
| variable | Declared inside a process statement with a variable declaration; assigned with a variable assignment statement. Variables are created at the time of elaboration and retain their values throughout the entire simulation run. Note that variable assignments occur without delay (unlike signal assignments). This has major implications when variable assignments are used as part of a block of sequential statements within a process. See the VHDL User Manual for details. |
| wait | Suspends evaluation of a process. There are three basic forms of a wait statement:  wait on sensitivity\_list;  wait until boolean\_expression;  wait for time\_expression;  These forms can be combined:  wait on sensitivity\_list until boolean\_expression for time\_expression; |
| when | Used to present choices for conditional logic in a case statement. |
| while | Used to iterate replications in replicated a loop statement. Also used for conditional logic in selected waveforms. |
| with | Introduces the select expression of a selected signal assignment statement. |
| xnor | Logical operator for types bit and Boolean and for one-dimensional arrays of these types. Logical exclusive nor. |
| xor | Logical operator for types bit and Boolean and for one-dimensional arrays of these types. Logical exclusive or. |