VHDL LANGUAGE ELEMENTS

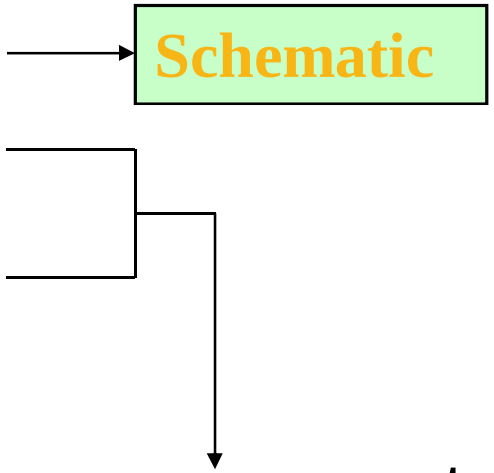
VHDL is composed of language ***building blocks*** that consist of more than **75 *reserved words*** and about 200 ***descriptive words*** or ***word combinations***

Reserved VHDL keywords

ABS ACCESS AFTER ALIAS ALL AND ARCHITECTURE ARRAY ASSERT ATTRIBUTE BEGIN BLOCK BODY BUFFER BUS CASE COMPONENT CONFIGURATION CONSTANT	DISCONNECT DOWNTO ELSE ELSIF END ENTITY EXIT FILE FOR FUNCTION GENERATE GENERIC GROUP GUARDED IF IMPURE	IN INERTIAL INOUT IS LABEL LIBRARY LINKAGE LITERAL LOOP MAP MOD NAND NEW NEXT NOR NOT NULL	OF ON OPEN OR OTHERS OUT PACKAGE PORT POSTPONED PROCEDURE PROCESS PURE RANGE RECORD REGISTER REM REPORT ROL ROR	RETURN SELECT SEVERITY SIGNAL SHARED SLA SLL SRA SRL SUBTYPE THEN TO TRANSPORT TYPE UNAFFECTED UNITS UNTIL USE	VARIABLE WAIT WHEN WHILE WITH XNOR XOR

Design Description Methods

- *Structural Description Method*
- *Behavioral Description Method*
- *Data-Flow Description Method*



- These two are similar in that both use a process to describe the functionality of a circuit

VHDL language abstractions

- VHDL is rich in language abstractions, in addition to which the language can be used to describe different abstraction levels, from functions right down to a gate description
- Abstraction levels are a means of concealing details

Definitions of the Description Methods

- *Structural Description Method:* expresses the design as an arrangement of interconnected components
  - It is basically schematic
- *Behavioral Description Method:* describes the functional behavior of a hardware design in terms of circuits and signal responses to various stimuli
  - The hardware behavior is described algorithmically
- *Data-Flow Description Method:* is similar to a register-transfer language
  - This method describes the function of a design by defining the flow of information from one input or register to another

Functional (system) level

- Algorithms can be describe at this level
- E.g. a controller algorithm can be described and simulated on the computer
- An algorithm does not need to contain any time information
- Specifications written in VHDL will be able to be simulated

Behavioral level

- Behavior and time are described at this level
- No architecture is required here
- The advantage of models at this level is that models for simulation can be built **quickly**
- A *behavioral model* can be described as functional modules and an interface between them
- The modules contain one or more functions and time relations
- In certain cases the architecture can be defined

Comparing the Description Methods

A *Behavioral Description* uses a small number of processes where each process performs a number of sequential signal assignments to multiple signals

In contrast, a *Data-Flow Description* uses a large number of concurrent signal assignment statements

A concurrent statement executes asynchronously with respect to other concurrent statements

Concurrent statements used in *Data-Flow Description* include:

- block statement (used to group one or more concurrent statements)
- concurrent procedure call
- concurrent assertion statement
- concurrent signal assignment statement

RTL = Register Transfer Level

- It consists of a language which describes behavior in
  - asynchronous and synchronous state machines
  - data paths
  - operators (+,*,<,>,...)
  - registers

Electrical level

- Other name is: transistor level
- There are models of
  - transistors
  - capacitances
  - resistances
- This is not supported in VHDL

Layout level

- At layout level models are made of the physical process
- This is not supported in VHDL

Synthesis = Increasing Complexity

- Synthesis is done between each level
- The volume of information increases between the various abstraction levels
  - E.g. technology information is required to synthesize from RT to gate level
  - Each transition (synthesis) generates more information
- In order to implement a function in an ASIC, are required the followings:
  - technology information
  - wiring information
  - gate information
  - set-up times

Why are different abstraction levels used?

- It is usually the requirements that determine the abstraction level at which the information is to be described
- If a short development time is required, a high abstraction level should be chosen as the description language
- In practice RT level (and parts of behavioral) can be synthesized automatically to gate level

Different applications

- **ASIC** = Application Specific Integrated Circuit
  - Usually includes FPGA, gate array, standard cell and full custom designs.
- **PCB** = Printed Circuit Board design
  - On a circuit board there are usually several ASICs together with a microprocessor and its infrastructure
- **System** = a number of PCBs

Model evaluation

- The code for VHDL component can be verified functionally in a simulator
- The simulator simulates (“executes”) the VHDL code with input signals and produces a signal diagram and error messages on the basis of the components
- The input signals are defined either in VHDL or in the simulator’s language
- When the VHDL code is simulated, functional verification takes place
- At a later stage, time verification of the design is also possible

Simulation

- Simulating models is an effective way of verifying the design
- The model in the computer is only a time-discrete model, however, while reality is continuous
- The computer model is more or less like reality
- It is least like reality at a high abstraction level (behavioral) and most like it at the lowest level (layout)

Other languages for describing electronics

- There are several languages which are used to describe electronic designs
- One popular language is called VERILOG
- It is used from RT level down
- In some other languages there are no hierarchies, which causes major problems when working on complex assignments
- There languages are developed by universities and research centers

Mechanisms by which to reduce complexity

- **Language abstractions** use the language to describe complex matters without having to describe small details
  - Functions and procedures are important parts of the language in order to handle complexity
- **Design hierarchy** uses components in order to conceal details - the black box principle
  - The term black box means that only inputs/outputs of a component are visible at a certain level
  - It is the designer who decides how many different hierarchies there are to be in the design

Key feature: delta delay

- VHDL uses the concept of delta delay to keep track of processes that should occur at a given time step, but are actually evaluated in different machine cycles
- A delta delay is a unit of time as far as the simulator hardware is concerned, but in the simulation itself time has no advance

VHDL component

- Components are a central concept in VHDL
- Components are used, among other things, to build up component libraries, e.g.:
  - microprocessors
  - special user circuits
  - other standard circuits
- If a “good” component has been designed, it can be saved in a component library, enabling it to be copied as many times as required, i.e. components are reusable
  - this is called creating instances, i.e. creating the component in a schematic or in the text file

Object-based language

- Staying with computer science a while longer, VHDL is an object-based language, i.e. what separates VHDL from object-oriented languages is that the language does not have inheritance
- Generic components and instantiation are typical for object-based languages
- Generic components are components which can be modified before instantiation, e.g. a generic component which copes with different width for the input and output signals

Using the black box

- The internal structure can be concealed from the designer - the black box principle
- In some cases there is no need to know how to component is structured
- The designer is usually only interested in
  - inputs and outputs
  - a specification function and
  - access times
- The majority of hardware designers are used to working with black boxes such as the 74LSXX circuit family, for example

Concurrent & Sequential Statements

Concurrent statements define interconnected processes and blocks that together describe a design's overall behavior or structure

They can be grouped using block statement. Groups of blocks can also be partitioned into other blocks

At this same level, a VHDL component can be connected to define signals within the blocks

It is a reference to an entity

A process can be a single signal assignment statement or a series of sequential statements (SS)

Within a process, procedures and functions can partition the sequential statements