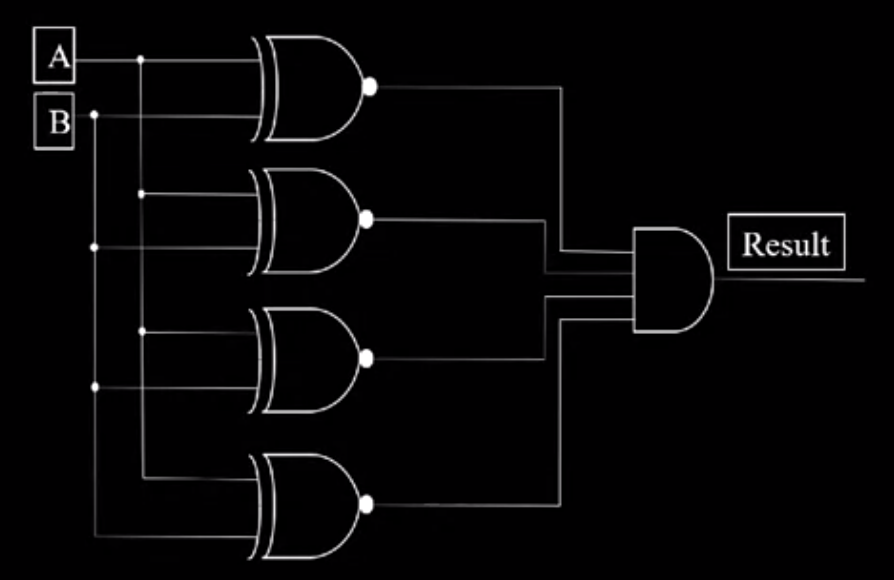
**Notes of the class**

The development of VHDL was initiated in 1981 by the United States Department of Defence to address the hardware life cycle crisis. The cost of reprocuring electronic hardware as technologies became obsolete was reaching crisis point, because the function of the parts was not adequately documented, and the various components making up a system were individually verified using a wide range of different and incompatible simulation languages and tools. The requirement was for a language with a wide range of descriptive capability that would work the same on any simulator and was independent of technology or design methodology.

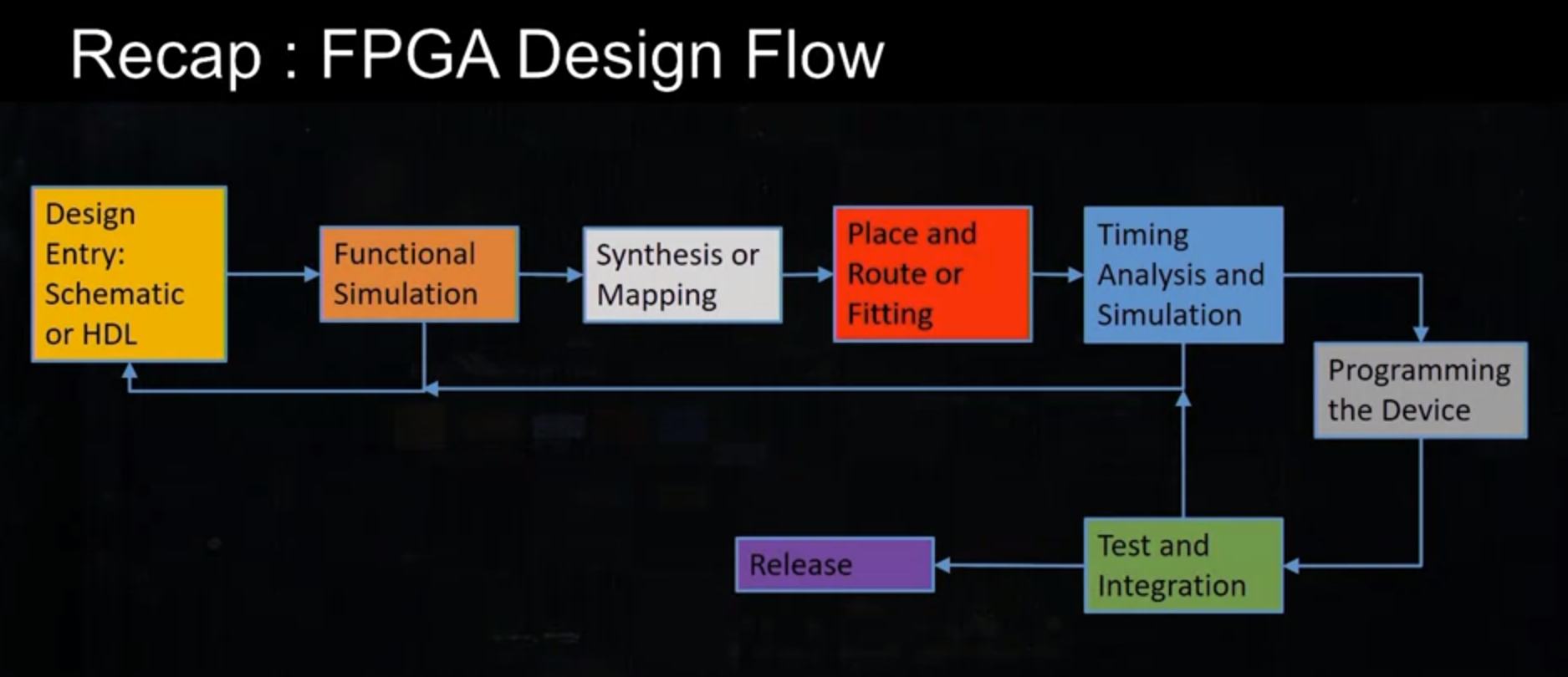
VHDL is a structured language meaning that it consists of a common structure for every script divided into boxes that are meaningful itself.

It is used to describe hardware such as the following 4-bit comparator.



## FPGA Design Flow

In FPGA, the design flow is the following:



**Which is the difference between Schematic and Hardware Desription Languages?**

Schematic is highly correlated to the physical implementation of the circuit since the implementation is based on physical components interconnected. However, for large designs it would be very difficult to transfer the design to FPGA or ASIC. So, using Schematic design entry is easily transferable to other FPGA or ASIC tool flows ? No, **Schematic entry** tools are specific to a vendor tool data entry flow, but **HDLs** are **text and are easily emailed or transferred to other projects**. Basically, an schematic would be difficult to be transferred and understood by any other person while HDL allows the professional to send the design through text.

HDL Key concepts:

* Describe a hardware circuit implementation that the tool chain will interpret and synthesize into FPGA logic cells, to everything is compact and condensed in a logic cell that can be programmable if required.
* HDL is concurrent, not sequential like other languages.
* FPGA logic cells are hardware, they inherently are executed in parallel.
* CPU code is software so it is inherently executed in sequence.
* VHDL is not key sensitive, meaning that variables DATA\_out and data\_Out are the same.

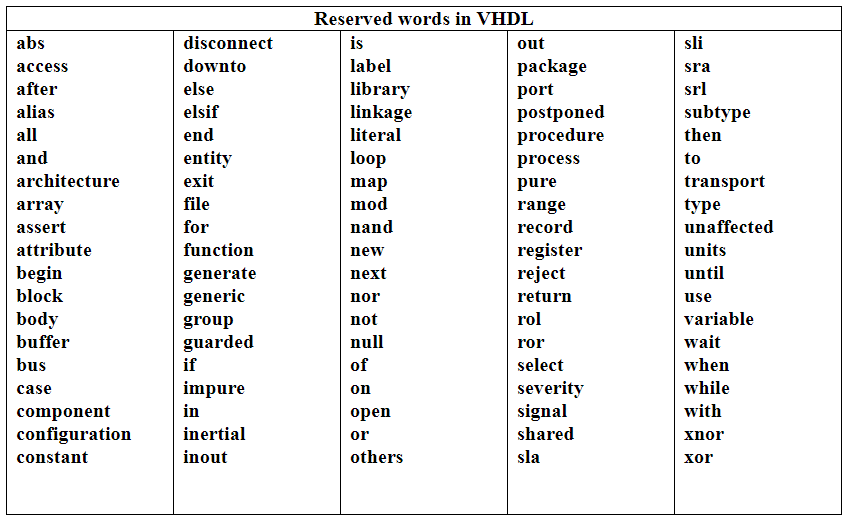
Structure of HDL

VHDL consists of:

Entity: the interface of the system, input and output ports.

Architecture: the logic design of our circuit.

**VHDL Keywords:**



## VHDL Modeling

There are 3 main VHDL modeling methodologies: Structural, Dataflow and Behavioural.

1. Structural Modeling

There are structural keywords that allow the usage of logic elements in the code like library defined primitive gates (and2/or2), Boolean, bitwise logical (and/or) and library user defined functions (and17).

1. Dataflow modeling

Use assignment and select statements for the code.

1. Behavioral modeling

Use assignments with a process, using the sensitivity list of A and B signals for example.

Components of VHDL

Signals are a wire to the entity denoted as std\_logic. The operator signal assignment (<=) initializes a signal. Variables are defined by (:=) operator and it assigns the variable in the process immediately and does not need to wait for an event like a clock edge in the process.

Datatypes in VHDL are divided into:

* Array:
  + String. “abc”
  + Bit\_vector: “1001”
  + Std\_logic\_vector: allows 9 values of 1,0,X,Z,W,L,H,U. i.e. “101Z”
* Scalar
  + Character: ‘a’
  + Bit: ‘1’, ‘0’
  + Std\_logic: ‘1’, ‘0’, ‘X’, ‘Z’
  + Boolean: true, false
  + Real, integer: 3.14 1E+0.27
  + Time: fs, ps, ns, us, ms

Comments are put with two dashes (--) and every sentence must end with semicolon (; ).

Example Codes

1. 2 Bit comparator

A 2 bit comparator take 2 signals of 2 bits and compares the values of the two bits of each signal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A1A2/B1B2 | 00 | 01 | 10 | 11 |
| 00 | 1 | 0 | 0 |  |
| 01 | 0 | 1 | 0 |  |
| 10 | 0 | 0 | 0 |  |
| 11 | 1 |  |  |  |