Introduction to HDL:- (Hardware description Languages)

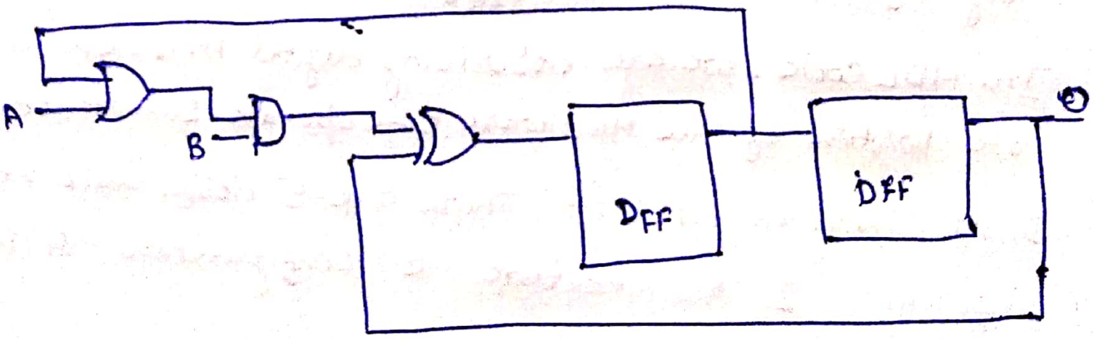
HDL is a language to describe Hardware. Digital circuits ~~as~~ consist primarily of interconnected transistors. we Structure and analyze these circuits with the help of hierarchical Structure. The Hierarchical Structure allows us to effectively represent a digital circuit by means of Interconnected diagrams. we can Say this is Schematic. This visual approach to describing a digital circuit is intuitive but it becomes impractical as complexity increases. Another way to describe a digital circuit is to use a textual language that is Specifically intended to clearly and concisely capture the defining feature of digital design. Such language are called Hardware Description Language (HDL) The most popular HDL are Vertilog and VHDL. They are widely used in conjunction with FPGAs(Field programmable Gate Array) An FPGA is an array of logic gates and this array can be programmed. HDL allow you to describe a circuit using words and Symbols.

Sumit Agarwal
9960132130

~~Vertilog~~

⇒ why HDL Required?

Let we have to design a circuit



Suppse we have to design above circuit which have 10times logic gates and FF. Now design of Such circuit Structure is nearly impossible due to Scalability. So we need HDL.

Scanned with CamScanner

Vertilog and VHDL are two languages that are commonly used to describle digital circuits. Both vertilog and VHDL are powsful tools that can be used to design a complex digital circuit

Vertilog Evolution:- The word vertilog is introduce ~~for~~ from words "verification" and "logic" because the language was first suggested as a Simulation and verification tool. vertilog was designed in early 1984 by Gateway design automation

VHDL Evolution:- VHDL Stands on VASIC HDL (Very High Speed integrated circuit) VHDL is developed by Department of ~~Defs~~ Defense (DOD)-US in 1980.

Programming Language VS. HDL

As we know that we already ~~famli~~ familiar with programming languages. HDLs resemble high level programming languages such as c or Python but there are fundamental difference: Statement in HDL code involve parallel operation, whereas the programming language represent Sequential operation.

When we write a computer program or firmware module, processo will execute lines of code on at a time, following top-to-bottom organization.

Sumit Agarwal
9960132130

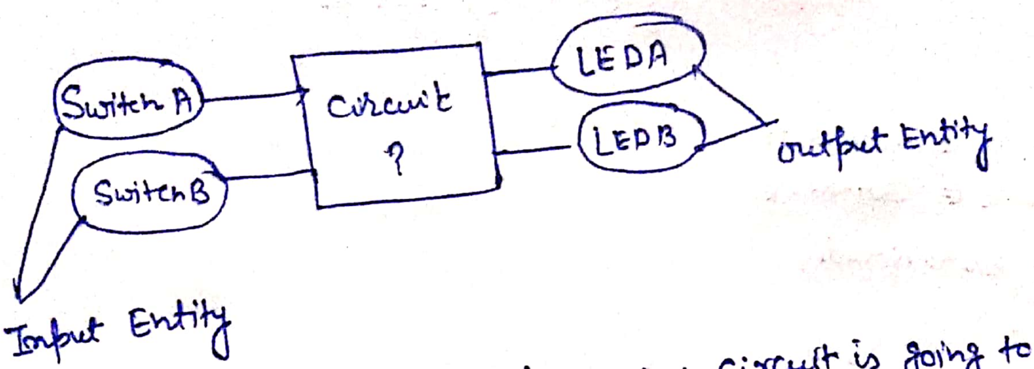
In HDL code, we are describing digital Hardware, and separate portion of this Hardware can operate Simultaneously.

Design units in VHDL:- There are 5 design units in VHDL

1. Entity 2. Architecture (3) Configuration (4) Package Declaration (5) Package Body

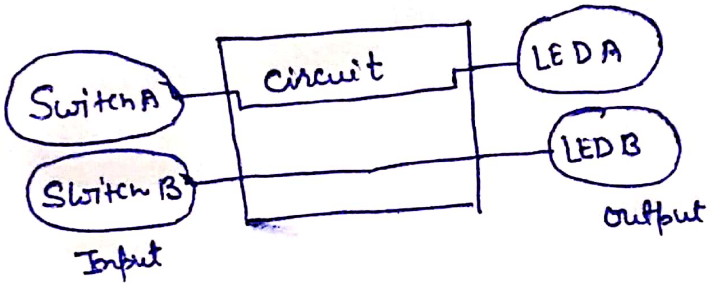
Entity:- It is name plate of the design. Entity define Input and output ports.

Scanned with CamScanner



Architecture:- What / How my circuit is going to behave or function.

(ऊपर fig मे हमे यह पता है कि Switch A व Switch B के On होने पर क्रमश: LEDA व LEDB On होती है अतः Switch A, Switch B, LEDA व LED B क्रमश: Input व output Entity है Architecture हमे यह बताता है कि हमारा circuit किस प्रकार का होना चाहिए कि हमे desired o/p मिले जैसे यहा Switch A → LEDA से व Switch B - LED B से connect है



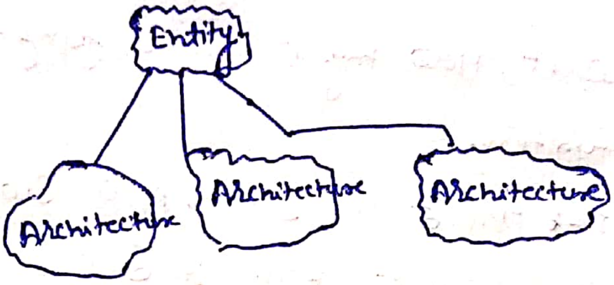
How to Design Architecture.

Blue → Keyword
Black → used defined

```
library IEEE                              } → library
use IEEE, STD_LOGIC_1164.ALL              }
                                  → Port Type (Input port)
entity Switches_LEDs
port {  Switch-A : in  STD-LOGIC;  ⟷ Data type : STD-Logic
        Switch-B : in  STD-LOGIC;                  ↑
        LED-A : out  STD-LOGIC;             Defined in Library
        LED-B : out  STD-LOGIC}; ← ~~end~~ last semicolon must
                                        outside the braces
end Switchs_LEDs → must have same name (entity start व
                     → Same Name        end same name से
architecture Behavioural of Switches_LEDs is    ही होंगे)
                     — Same Name
begin
end Behavioural
```

Scanned with CamScanner

```
architecture Behavioral of Switches_LEDs is
begin
  LED_A <= Switch_A;
  LED_B <= Switch_B;
end Behavioral;
```



In VHDL we can map to One Entity to multiple architecture. So configuration tells us How Entity mapped with architecture.

Package Declaration!- If I defined own data type such that STD_LOGIC etc such own data type is comes under package Declaration

Package Body!- In Package body we define the body/functionality of own datatype.

Sumit Agarwal
9960132130

VHDL Capabilities!- * It is case ~~Senstive~~. insenstive language.
* It is vendor independent

* It supports Simulation. In absence of Real System, we Simulate the function by taking the model of that function.

Types of Modeling Styles in VHDL!-

(i) Data Flow Modeling :- It Present design equation. In Dataflow Modeling we directly implement the design equation.

(ii) Behavioral Modeling: We design the program according to Behaviour of System.

(iii) Structural Modeling: Follows the Bottom-up approach. First we design ~~A~~ Sub module, ~~comp~~ combine the module to get the desired Solution

(iv) Hybrid Modeling:

VHDL operators:- VHDL Support the following operators

(i) Logical operators (ii) Relational operators (iii) Shift operators (iv) Adding operators (v) Multiplying operators (vi) Miscellaneous operators

Logical operators:- VHDL Support the following logical operators

(i) and (ii) or (iii) Nand (iv) NoR (v) XOR (vi) Xnor (vii) not

Sumit Agarwal
9968132130

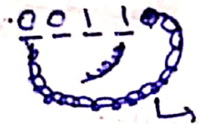
Relational operators:-

(i) equal to (ii) Not equal (iii) Less than (iv) & less than equal to (v) Greater than (vi) Greater than equal to

Shift operators:

(i) ~~[illegible]~~ Sll : Shift logical left
Srl : Shift logical right
Sla : Shift arithmetic left
Sra : Shift arithmetic right
Rol : Rotate left
Ror : Rotate Right

ex (i) 0011 sll 1 =



= 0110 (Shift 1 bit left and insert 0 at vacant place)

↳ 0 1 1 0
↑ Insert Zero at vacant Place

(ii) 1100 srl 2 = (Shift 2 bit right and Insert 0 at vacant Place)
= 0011

(iii) 1100 sla 1 = Shift 1 bit left and insert zero bcoz Right most is zero (1000)

1101 sla 1 = 1011

Sumit Agarwal
9968132130

(iv) 1100 sra 2 = Shift 2 bit right and Insert 1 bcoz left most is 1 (1111)

(v) 1010 rol 1 = ~~to~~ 0101

(vi) 0011 ror 1 = 1001

Adding operators

(i) ~~A~~ + ~~B~~ (Addition) (ii) – (Substraction) (iii) & (contatenation)
(EX B = '0' & '1' & '1' = 010)