2. INTRODUCTION TO VHDL

HDL :- (HardWare description Language):-

→ HDL's are used to verify the functionality of digital ckt's before its hardware implementation

→ HDL's are used to model a digital system., digital system contain a simple gate (or) complex electronic system.

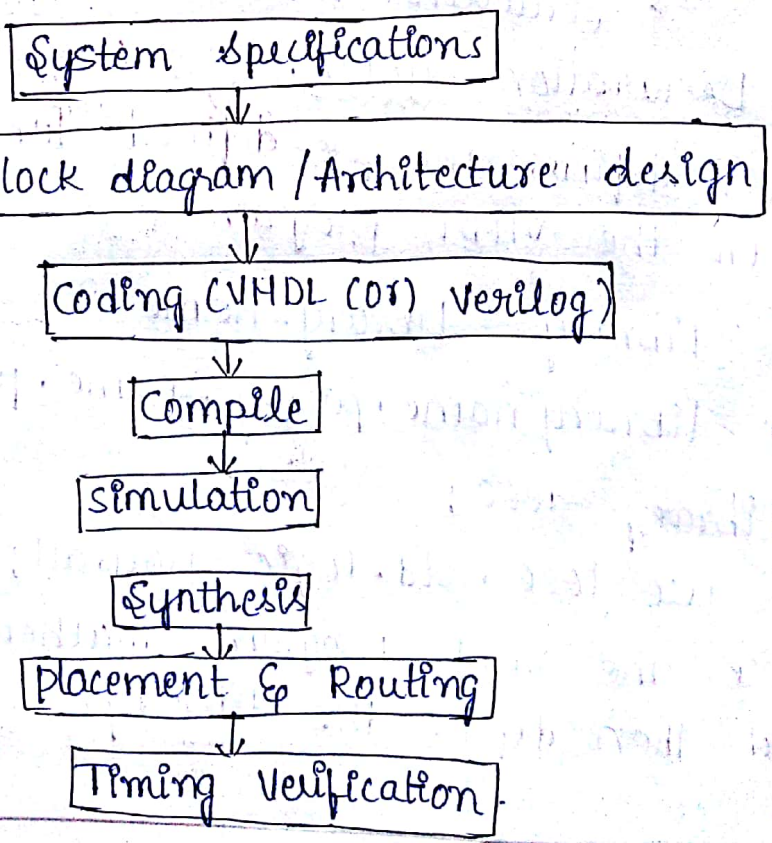
→ HDL's represents the logic diagrams and other digital information in terms of textual form.

→ There are 2 types of HDL's

* VHDL
* Verilog HDL

VHDL → [V H S I C H D L] → Very high speed integrated ckt hardware description language

VHDL Design Flow:-



www.Jntufastupdates.com Scanned by CamScanner

Simulation:- Verification and validation of a digital ckt is known as simulation.

Synthesis:- To generate the net list file i.e inter connections b/w i/p & o/p's of a digital ckt.

VHDL program structure:-

→ Library Declaration

→ Entity Declaration

→ Architecture Declaration

VHDL modelling styles:-

There are 3 different types of modelling styles in VHDL there are

→ Data Flow model

→ Behavioural model

→ Structural model

Library Declaration Syntax:-

It defines the pre-defined functions and key-words in the VHDL program.

Library Library-name

use Library name.package_name.package-parts;

Ex:- Library ieee;

use ieee.std-logic_1164.all;

→ Suppose the VHDL program mathematical operations are used then define the library.

www.Jntufastupdates.com Scanned by CamScanner

use ieee . std_logic_arith . all; (2)

Similarly for signed numbers and unsigned numbers the library declarations are

use ieee . std_logic_signed . all;

use ieee . std_logic_unsigned . all;

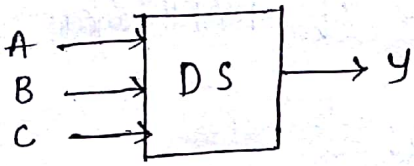
Entity Declaration Syntax:-

It defines the input and outputs of the digital system.

Syntax:-

Entity Entity_name is
port (Data objects : mode Datatype);
End Entity_name;

Example:-



Entity Digital is
port (A, B, C : In std_logic;
y : Out std_logic);
End Digital;

Architecture Declaration Syntax:-

It defines the actual logic of the digital system i.e relation b/w i/p's and o/p's.

Syntax:-

Architecture Architecturename of Entityname is
Declarations;
Begin
Statements;

www.Jntufastupdates.com Scanned by CamScanner

End Architecture_name;

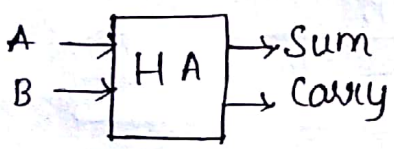
Eg:- Y = ABC

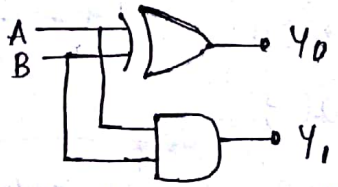
```
Architecture Dataflow of Digital is
Begin
    Y <= A and B and C;
End dataflow;
```

Write a VHDL program for a NAND Gate?

```
Library ieee;
use ieee.std_logic_1164.all;
Entity NAND Gate is
 port (A, B : In Bit;
        Y : Out Bit);
 End NAND Gate;
Architecture Dataflow of NAND Gate is
 Begin
     Y <= A nand B;
 End dataflow;
```

Write a VHDL program for a half Adder?






```
Library ieee;
use ieee.std_logic_1164.all;
Entity half adder is
 port (A, B : In std_logic;
       Sum, Carry: Out std_logic);
End half adder;
Architecture Dataflow of halfadder is
 Begin.
```

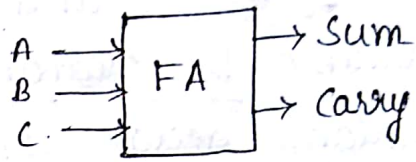

www.Jntufastupdates.com Scanned by CamScanner

Sum <= A ⊕ B ;

Carry <= A and B ;

End Dataflow;

Write a VHDL program for a full Adder?




```
Library ieee;
use ieee.std_logic_1164.all;
Entity full adder is
port (A, B, C : in std_logic;
   Sum, carry : out std_logic);
End full adder;
Architecture Dataflow of full adder is
Begin
   Sum <= A ⊕ B ⊕ C;
   carry <= AB + BC + CA;
End dataflow;
```

Data Objects in VHDL:-

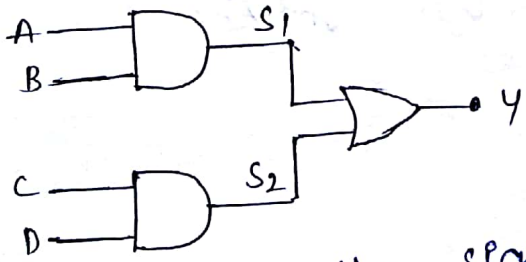
→ Signal Declaration
→ Variable Declaration
→ Constant Declaration
→ Component Declaration

Signal Declaration:-

Signal Declaration Syntax:

Signal signal_name : Data type;

Ex:-



Where S_1 & S_2 are the signals

Signal S_1, S_2 : std_logic;

www.Jntufastupdates.com Scanned by CamScanner

Signal Assignment Declaration:-

Signal <= Expression;

Ex:- S1 <= A and B;

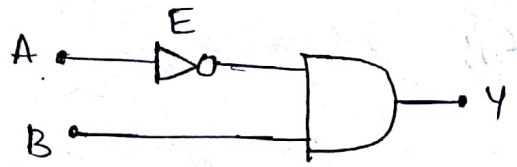
S2 <= C and D;

Variable Declaration:-

Variable declaration Syntax:-

Variable Variable_name : Data type.

Ex:-



Where 'E' represents the variable

Variable E : std_logic;

Variable Assignment Declaration:-

Variable_name := Expression

Ex:- E := NOT A

Constant Declaration:-

Constant Declaration Syntax:-

Constant const_name : datatype := value;

Ex:-

Constant pi : Real := 3.14;

Constant c : Real := 2;

Constant ee : Real := 2.17;

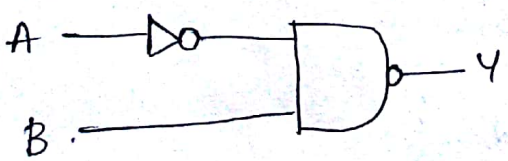
Component Declaration:-

Component component_name is

port (Data objects: mode Data type);

End component;

Ex:-



www.Jntufastupdates.com Scanned by CamScanner

```
Component NOT Gate is
 port (x: In  std_logic;
       Y: Out std_logic);
End component;
Component Nand Gate is
 port (x,Y: In  std_logic;
       z  : Out std_logic);
End component;
```

In component declaration define the size of each and every component present in the logic diagram.

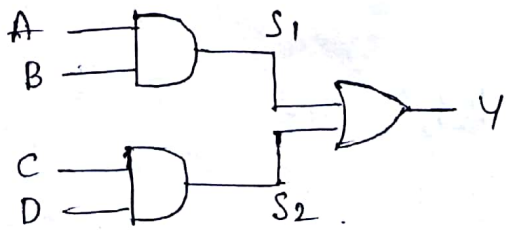
Write a VHDL program for the following boolean Expression?

$$Y = (\overline{AB+CD})$$

```
Library ieee;
 use ieee. std_logic_1164.all;
 Entity Boolean Expression is
 port (A,B,C,D : In Std_logic;
       Y       : Out std_logic);
 End Boolean Expression;
 Architecture Dataflow of Boolean Expression is
 Begin
      Y <= NOT (A and B) OR (C and D);
 End Dataflow;
```

Write a VHDL program for the following Boolean Expression using Signal Declaration?

$$Y = AB + CD$$

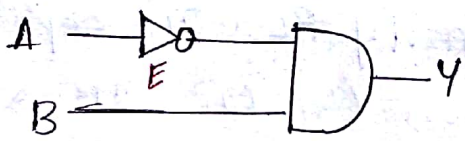


www.Jntufastupdates.com Scanned by CamScanner

```
Library ieee;
use ieee.std_logic_1164.all;
Entity Expression is
port (A,B,C,D : In  std_logic;
       Y : Out std_logic);
End Expression;
Architecture dataflow of Expression is
      Signal S1, S2 : std_logic;
Begin
      S1 <= A and B;
      S2 <= C and D;
      Y <= S1 or S2;
End Dataflow;
```

Write a VHDL program for the following Boolean Expression using variable declaration?

$Y = \bar{A}B$




```
Library ieee;
use ieee.std_logic_1164.all;
Entity Boolean Expression is
port (A,B : In  std_logic;
       Y : Out std_logic);
End Boolean Expression;
Architecture Dataflow of Boolean Expression is
  Variable E: std_logic;
Begin
      E := NOT A
      Y <= E And B;
End dataflow;
```


www.Jntufastupdates.com Scanned by CamScanner

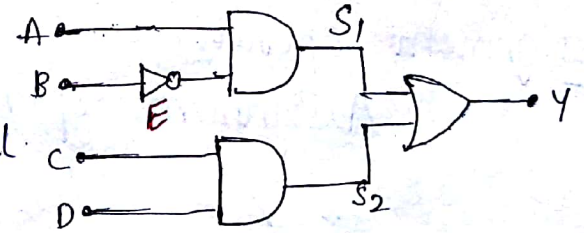
Write a VHDL program for the following Expression using constant Declaration. ⑤

$Y = \Pi + A$ [Where Π = constant]

```
Library ieee;
use ieee.std_logic_1164.all;
Entity Expression is.
 port (A: In (std_logic) Integer;
       Y: Out          Integer);
End Expression;
Architecture Dataflow of Expression is
 Constant pi : Real : = 3.14;
 Begin
        y <= pi + A;
End Dataflow;
```

Write a VHDL program for the following expression using variable and signal declaration

$Y = (A\bar{B} + CD)$




```
Library ieee;
use ieee.std_logic_1164.all
Entity Expression is
port (A,B,C,D: In   std_logic;
       Y: out   std_logic);
End Expression.
Architecture Dataflow of Expression is
  Signal S1,S2 : std_logic;
   Variable E : Std_logic;
Begin
        E : = NOT B;
        S1 <= E and A;
        S2 <= C and D;
        Y  <= S1 or S2;
End Dataflow;
```


www.Jntufastupdates.com Scanned by CamScanner

Operators in VHDL:-

* Logical Operators
* Assignment Operators
* Arithematic Operators
* Relational Operators
* Shift Operators
* Miscellaneous Operators

Logical Operators:-

There is a logical operation is performed b/w i/p's and resultant is assigned to the output

The logical operators in VHDL is given by

AND, OR, NAND, XOR, NOR, NOT, XNOR.

Ex:- y <= A and B;

y <= A or B;

y <= A xor B;

Assignment Operators:-

Assignment operators in VHDL is given by

→ <=

→ := =

y <= A and B;

For declaring the variables and constants in VHDL as follows.

Eg:- Variable E : std_logic;

E := NOT A;

→ constant α : Real := 4;

Arithematic Operators:-

Arithematic operators in VHDL is given by

→ Addition → +

→ Subraction → -

www.Jntufastupdates.com Scanned by CamScanner

→ Multiplication → *
→ Division → /
→ Modulus → MOD
→ Remainder → REM
→ Exponential → * *

Eg:- y <= A+B;
y <= A*B;

Relational Operators:-

Relational Operators in VHDL is given by

→ Equality → =
→ Inequality → /=
→ Less than → <
→ Greater than → >
→ less than or Equal to → <=
→ Greater than or Equal to → >=

Eg:- If (A < B) then
y <= '0';
Else y <= '1';
End if;

Shift Operators:-

Shift operators in VHDL is given by

* Shift left logical
* Shift Right logical
* Shift left arithematic
* Shift right arithematic
* Rotate left
* Rotate right

Eg:- A = 100101
SLL A

www.Jntufastupdates.com Scanned by CamScanner

Y = 001010

A = 100101

(SRLA)Y = 010010

* Miscellaneous Operators:-

Miscellaneous operators in VHDL is given by

→ Concatenation combining operator → &

→ Absolute Operator → ABS

Eg:-

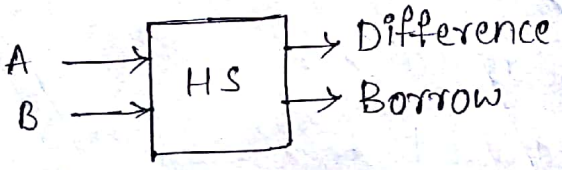
"1001" & "0011"

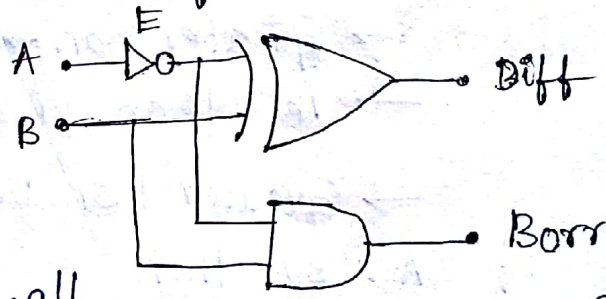
Y = 10010011

Temperature

ABS 27°C

Write a VHDL program for a half subractor



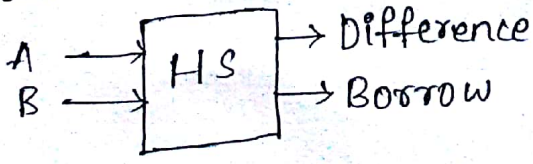


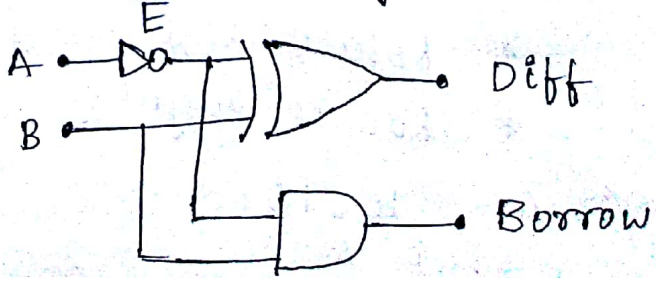
Difference = $A \oplus B$

Borrow = $\overline{A}B$

```
Library ieee;
 use ieee.std_logic_1164.all
Entity Half Subractor is
port (A,B: In   std_logic;
    Diff,Borrow: Out std_logic);
End Half Subractor
Architecture dataflow of half subractor is
Begin
                          XOR
      Difference  <= A ⊕ B;
      Borrow      <= (NOT (A and B));
End Dataflow;
```

Write a VHDL program for Subractor using variable declaration?





www.Jntufastupdates.com Scanned by CamScanner

```
Library ieee;
  use ieee.std_logic_1164.all
Entity Subractor is
port (A,B : In  std_logic;
    Diff, Borrow: Out std_logic);
End Subractor;
Architecture dataflow of Subractor is
    Variable E : = std_logic;
Begin
            E : = NOT A;
      Difference < = A XOR B;
       Borrow  < = A and B;
End  Dataflow;
```

Data Flow Model Syntax:-

→ Based on Output Expression of a digital ckt.

→ As a set of comment concurrent assignment statements. The outputs are specified in the form of boolean Expressions.

Syntax:-

```
Library  Library_name;
use library_name.package_name_package_
                                parts_all;
Entity  Entity_name is
port (Data Objects: Mode  datatype);
End  Entity_name;
Architecture Architecture name of Entity name is
Declarations;
Begin
        Assignment statements;
End
    Architecture_name;
```

Ex:- Y=AB+C

```
library ieee;
```

```
use ieee . std_logic_1164.all
Entity Boolean Expression is
port (A,B,C : In std_logic;
      Y : Out std_logic);
End Boolean Expression
Architecture Dataflow of Boolean Expression is
Begin
      Y <= ((A and B) OR C);
End dataflow;
```

Libraries:-

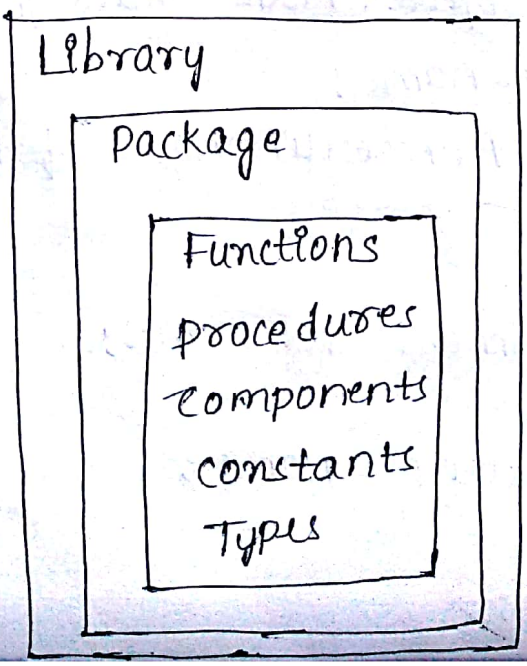
Functionality:- It defines the predefine functions and keywords that are used in the VHDL program

Syntax:-

```
Library Library name;
use library_name.package_name package-parts.all;
```

Ex:-

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arithematic.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_unsigned.all;
```



www.Jntufastupdates.com Scanned by CamScanner

Packages:-

Package declaration is used to store a common declarations such as procedures, functions, components constant, types, sub-types, sub-programmes etc ---

Syntax:-

```
Package  package-name is
package-item-Declarations
End package-name;
```

Eg:-

```
Package Logic 1 is
Constant α: Real := 5;
component XOR Gate is
port (A,B: In  std_logic;
       Y: Out  std_logic);
End Component;
End Logic 1;
```

→ package_item declaration consists of function declaration, procedure declaration, constant declaration, component declaration, Type declaration.

Structural Model:-

Based on Logic diagram of the digital System It is the interconnection of components.

Syntax:-

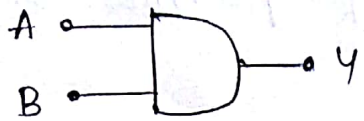
```
Library  Library-name;
use library-name.package-name-package-parts.all;
Entity  Entity-name is
port (Data Objects : Mode datatype);
End Entity-name;
Architecture Architecture_name of Entity is
Declarations;
        Component Declaration
Begin
        Component  component-name  is
      port (Data Objects : mode  datatype);
End component;
```


www.Jntufastupdates.com Scanned by CamScanner

Component Instantiation Declaration :-

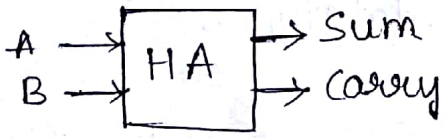
Component -label : Component-name portmap(Association List);

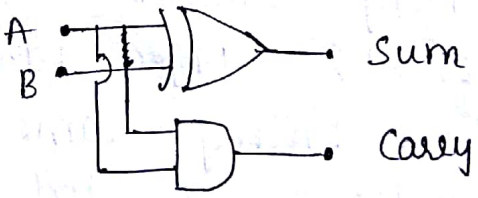
Ex:-




```
Library ieee;
use ieee.std_logic_1164.all
 Entity AND Gate is
port (A,B: In   std_logic;
      Y: Out   std_logic);
End AND Gate;
Architecture structural of AND Gate is
  Component AND Gate is
    port (C,D : In   std_logic;
          W: Out   std_logic);
    End Component;
    Begin
       X1: AND Gate  port map (A,B,Y);
    End structural;
```

Write a VHDL program for a half adder using structural model.





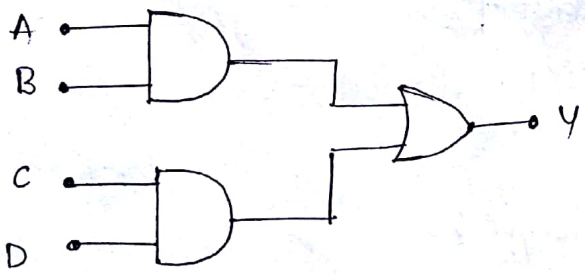

```
Library ieee;
 use ieee.std_logic_1164.all
Entity Half adder is
 port (A,B: In   std_logic;
       Sum,Carry : Out std_logic);
End Halfadder;
Architecture Structural of half adder is.
```


www.Jntufastupdates.com Scanned by CamScanner

```
Component AND Gate is
port (C,D : In   std-logic;
          S : Out  std-logic);
End component;
(Begin
    X1 : Half adder)
  Component XOR Gate is
  port ( E,F : In   std-logic;
             C : Out  std-logic);
  End component;
Begin
    X1 : XOR Gate port map (A,B, Sum);
    X2 : AND Gate port map (A,B, Carry);
End Structural;
```

Write a VHDL program for the following expression using structural model?

$Y = AB + CD$




```
Library ieee;
 use ieee. std-logic-1164.all;
Entity Expression is
port (A,B,C,D : In   std-logic;
              Y : Out  std-logic);
End Expression is
Architecture structural of Expression is.
Component AND Gate is.
```

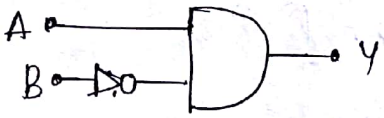
 Scanned by CamScanner

~~Component OR G~~

```
port (E,F : In  std_logic;
      W : Out  std_logic);
End Component;
Component OR Gate is
 port ( W,X : In  std_logic;
        Y : Out  std_logic);
 End Component;
 Signal S1,S2 : std_logic;
 Begin
    X1 : AND Gate  port map (A,B,S1);
    X2 : AND Gate  port map (C,D,S2);
    X3 : OR  Gate  port map (S1,S2,Y);
 End Structural;
```

Write VHDL program for the following boolean expression

$Y = A\bar{B}$



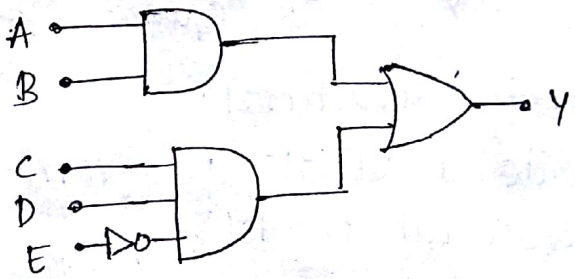

```
Library ieee;
use ieee.std_logic_1164.all;
Entity Expression is
port (A,B : In  std_logic;
      Y : Out  std_logic);
End Expression
Architecture structural of expression is
 Component AND Gate is
  port (A,B : In  std_logic;
        Y : Out  std_logic);
 End component;
    Variable E: std_logic;
```


www.Jntufastupdates.com Scanned by CamScanner

```
    E : = NOT B;
Begin
    X1 : AND Gate  port map (A, B̄, Y);
    X2 : NOT Gate  port map (B, E);
End structural;
```

Write a VHDL program for Boolean Expression using structural model.

$Y = AB + CDE$

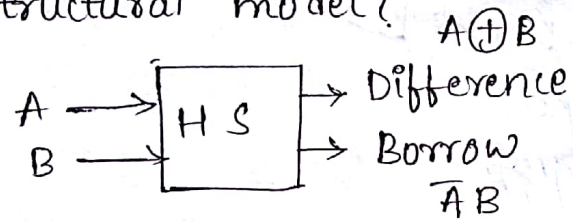


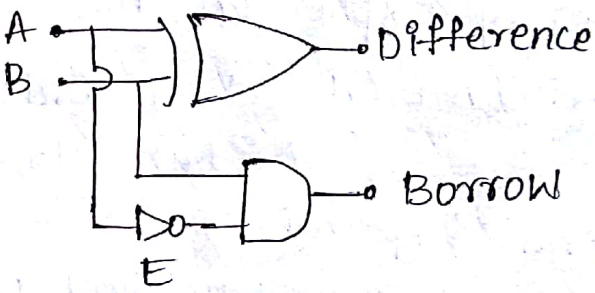

```
Library ieee;
use ieee.std_logic_1164.all;
Entity Expression is
port (A,B,C,D,E : In  std_logic;
      Y : Out  std_logic);
End Expression;
Architecture structural of Expression is
    Component AND Gate is
    port (A,B: In  std_logic;
          W: Out  std_logic);
    End component;
    Component AND Gate is
     port (C,D,E: In  std_logic;
           X : Out  std_logic);
    End component;
     Component NOT Gate is
      port (F : In  std_logic;
            E : Out  std_logic);
     End component;
     Component OR Gate is
        port (W,X : In  std_logic;
              Y : Out  std_logic);
```


www.Jntufastupdates.com Scanned by CamScanner

```
End Component;
    Signal S1, S2 : std_logic;
    Variable E : std_logic;
Begin
    X1 : AND Gate  port map (A, B, S1);
    X2 : AND Gate  port map (C, D, F, S2);
    X3 : NOT Gate  port map (F, E);
    X4 : OR Gate   port map (S1, S2, Y);
End Structural.
```

Write a VHDL program for half subractor using structural model?






```
Library ieee;
use ieee.std_logic_1164.all;
Entity subractor is
port (A, B : In  std_logic;
      Diff, Borrow : Out std_logic);
End subractor;
Architecture structural of subractor is
Component XOR Gate is
    port (A, B : In  std_logic;
          R : Out  std_logic);
End component;
Component NOT Gate is
    port (A : In  std_logic;
          Y : Out  std_logic);
End component
Component XOR AND Gate is
    port (A, B : In  std_logic;
```


www.Jntufastupdates.com Scanned by CamScanner

```
        z: Out std-logic );
    End component;
  Variable E : std-logic ;
Begin
      E := NOT A
  X1 : XOR Gate  port map (A,B, Difference );
  X2 : AND Gate  port map (B,E, Borrow);
  X3 : NOT Gate  port map (A,E);
End structural;
```

Levels of Abstractions in VHDL:-

There are 5 levels of abstractions in VHDL

→ Behavioural ~~model~~ level
→ RTL Level [Register transfer Level]
→ Gate Level
→ Transistor Level
→ Layout Level

Behavioural Level :-

It is the functional description of the digital system

→ It is used at the very begining stage of the design. In order to run a simulation also used to describe the text benches.

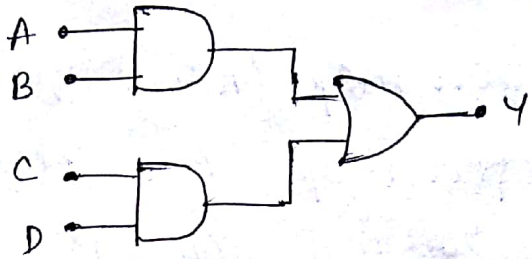
RTL Level :- [Registed transfer Level]:-

The description is divided into combinational logic and storage elements. The storage elements are controlled by the clock signal.

www.Jntufastupdates.com Scanned by CamScanner

Gate level:-

The design is represented as a net list (interconnection b/w i/p's and o/p's) with gates and some storage elements

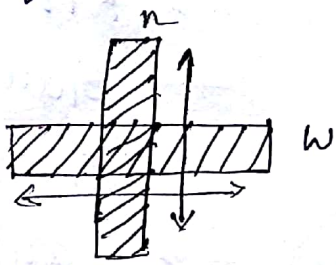


Transistor level:-

Now the gate level design can be implemented. interms of transistors; storage notes and interconnection between them.

Lay Out Level:-

Geometrical representation of transistors with specific shape.



Identifiers:-

Identifiers are used for the names of VHDL objects, procedures, functions and Entity names

There are 2 types of Identifiers

→ Basic Identifiers

→ Extended Identifiers.

Basic Identifiers:-

A basic Identifier in VHDL is composed of a sequence of (or) more characters.

A character is an upper case letter (A, B - - - Z), a lower case letter (a, b - - - z) and a digit (0, 1, 2 - - - 9). The first character in a basic

www.Jntufastupdates.com Scanned by CamScanner

Identifier must be a letter and the last character may not be an (-) underscroll

Eg :- a, A, sum, SUM, SUM sum, Half-Adder, Subractor

Extended Identifiers:-

An extended Identifier is a sequence of characters written between two (\\) backslashes.

A character is an upper case letter and lower case letter, digits and including following signals (., !, @, $, ')

Eg:- \A @ 5\, \carry $\, \MG @ $\

Datatypes in VHDL:-

Every Data objects in VHDL can hold a value that belongs to a set of values, These set of values is specified by using type declaration.

Different types of datatypes of VHDL is given by

- → Bit
- → std_logic
- → Bit type vectors
- → Real
- → Integer
- → Floating point
- → character.

Bit:- It represents the two values 0 (or) 1

Ex:- Entity Adder is
port (A, B, C : In Bit;
S, C : Out Bit);
End Adder;

www.Jntufastupdates.com Scanned by CamScanner

Std_logic :- It represents the '9' values

U → Uninitialisation
X → Unknown
0 → Logic '0'
1 → Logic '1'
Z → High Impedance
W → Weak Unknown
L → Weak zero
H → Weak One
- → Dont Care

Ex:- Entity Adder is
port (A, B, C : In std_logic;
S, C : Out std_logic);
End Adder;

Bit type Vectors:-

Vector type :-

→ Bit_Vector
→ Std_logic_Vector.

Bit Vector:- It represents the multiple inputs and multiple outputs of a digital system.

Eg:- Entity logic is
port (A: In Bit_Vector (7 down to 0);
Y: out Bit_Vector (2 down to 0));
End logic;

Std_logic_Vector:- It supresents the multiple inputs and multiple outputs of a digital system.

Eg:- Entity logic is.

 Scanned by CamScanner

```
port (A: In  std_logic_vector (7 down to 0);
      Y: out std_logic_vector (2 down to 0));
End Logic;
```

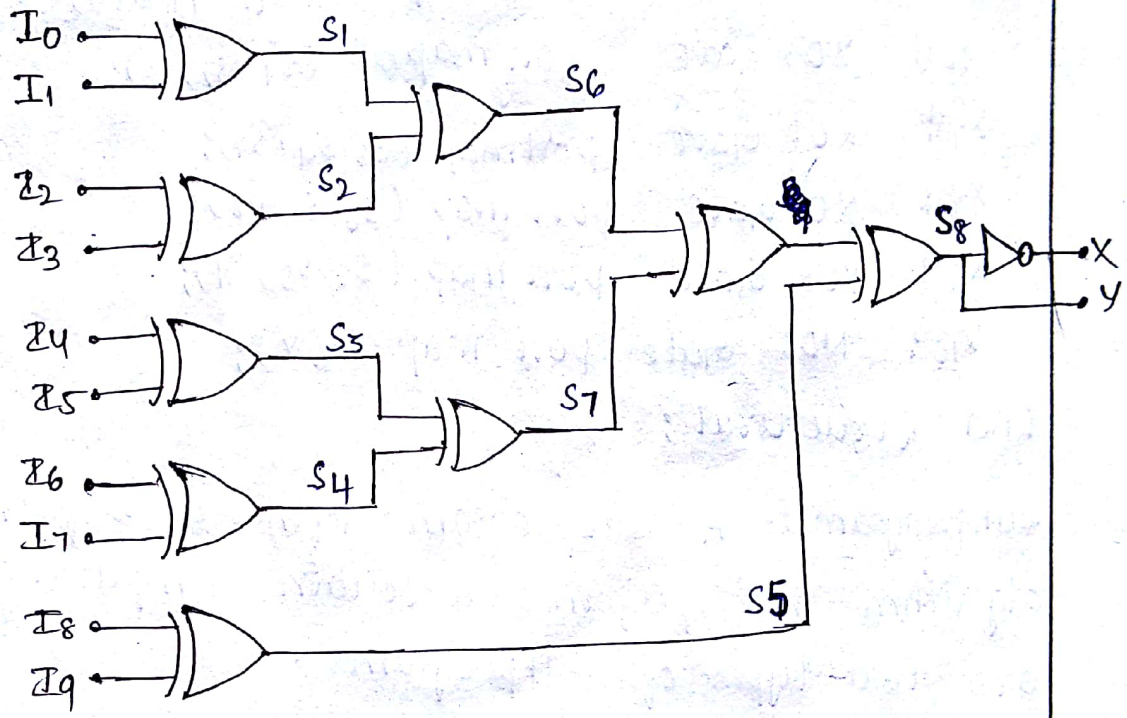
Real type:-

Eg:- Constant Pi : Real := 3.14;

Integer Data type:-

Eg:- Constant c: Integer := 3;

Write a VHDL program for the following logic diagram using structural model?




```
Library ieee;
  use ieee.std_logic_1164.all;
 Entity logic is
  port (I: In  std_logic_vector(9 down to 0);
        X: out std_logic;
        Y: Buffer std_logic);
  End Logic;
Architecture Structural of Logic 1 is.
   Component XOR Gate is.
 port (A,B: In std_logic;
       Z: Out std_logic);
  End Component;
   Component NOT Gate is
 port (C,D: In std_logic;
```

 Scanned by CamScanner

```
    W: Out  std_logic );
  End component;
Signal S : std_logic_vector (8 down to 1)
Begin
  X1: XORGate  port map (I0, I1, S(1));
  X2: XORGate  port map (I(2), I(3), S2);
  X3: XORGate  portmap  (I4, I5, S3);
  X4: XORGate  portmap  (I6, I7, S4);
  X5: XORGate  port map (I8, I9, S5);
  X6: XORGate  portmap  (S1, S2, S6);
  X7: XOR Gate portmap (S3, S4, S7);
  X8: XORGate  port map (S6, S7, S8);
  X9: XOR Gate port map (S8, S5, Y);
  X10: NOT Gate port map (Y, X);
End Structural;
```

Subprograms:- A subprogram defines a sequential algorithm that performs a certain computation. There are two types of subprograms.

1. Functions
2. procedures.

1. Functions:- These are usually used for computing a single value. A function executes in zero simulation time.

→ the general syntax of a subprogram specification for a function body is.

Syntax:- function function_name (parameter_list)
return return_type

Eg:-
```
function logic (clk: Bit)
  Return Boolean is
Begin.
  Return clk='1' and clk'event;
```


www.Jntufastupdates.com Scanned by CamScanner

End logic;

2. procedures:- These may be used to partition large behavioral descriptions. procedures can return zero (or) more values.

→ A procedure may (or) may not execute in zero simulation time, depending on whether it has a wait statement (or) not.

→ The syntax for the subprogram specification for procedure is

Syntax:- procedure procedure_name (parameterlist)

Eg:- procedure logic (A,B: In std_logic;
X,Y,Z: out std_logic);
Begin
X <= A and B;
Y <= A OR B;
Z <= A NAND B;
End logic;

Subprogram:-

Syntax:- Subprogram specification is.
Subprogram item declaration;
Begin
Subprogram statements;
End subprogram_name;

→ Subprograms can be used for explicit type conversions, logic-unit definitions, operator re-definitions, new operation definitions and other applications commonly used in programming language.

www.Jntufastupdates.com Scanned by CamScanner