Navbodh
PAGE NO.:
DATE: / /

Q2→ Discuss all the modelling supported by VHDL.

Ans) 1) Data flow modelling :- (A) It describes how data flows through the system. Data dependencies in the description match those in a typical hardware implementation.
(B) A data flow description directly implies a corresponding gate level implementation.

2) Behavioral Mod :- (A) It describes a system's behavior or function in an algorithm fashion.
(B) It is the most abstract style. The description is abstract in the sense that it does not directly imply a particular gate level implementation.

3) Structural Mod :- It is most useful & efficient when a complex system is described as an interconnection of moderately complex design entities. This approach allows each design entity to be independently designed & Verified before being used in the higher level description.

4) Mixed Mod :- This level describes the logic in terms of registers & the RTL logic consists only of boolean eq for the combinational logic bet the registers.

Structural Mod.:- For structure modelling, we deal with components & their interconnection.

→ Library IEEE library
use std_logic_1164.all
Entity

Component XOR21
A, B = in_std_logic.
C = out_std_logic

Component AND21
A, B : in std_logic
C : out std_logic

Component OR21
A, B, C : in std_logic
D : out std_logic

Signal S1, C1, C2, C3 std_logic

U1: XOR21 portmap (A, B, S1)
U2: XOR21 portmap (S1, Cin, Sum)
U3: AND21 portmap (A, B, C1)
U4: AND21 portmap (A, Cin, C2)
U5: AND21 portmap (B, Cin, C3)
U6: OR21 portmap (C1, C2, C3, Carry)

Rajshree
PAGE NO.:
DATE:

Q → What is VHDL?

→ Stands for Very high speed integration circuit HDL. It is an IEEE standard hardware description language that is used to describe & simulate the behaviour of complex digital circuits. ex:- Pulse generator, Priority encoder for 16 words, 8 bit RAM, etc. It supports following features:- Design exchange, Readability, Documentation, Standardization.

Capabilities:-

1] The language supports flexible design methodologies: top down, bottom-up, or mixed.

2] It supports both synchronous & asynchronous timing models.

3] There is no need to learn a diff. language for simulation control. Test benches can be written using the same language to test other VHDL models.

4] The language has elements that make large scale design modeling easier, for ex, components, functions, procedures & packages.

5] Models written in this language can be verified by simulation since precise simulation semantics are defined for each language construct.

6] The language is publicly available, human readable, & above all, it is not proprietary.

7] Various digital modeling techniques such as finite state machine ~~readable, and above all, it~~ descriptions, algorithmic descriptions and boolean eq's can be modeled using the language.

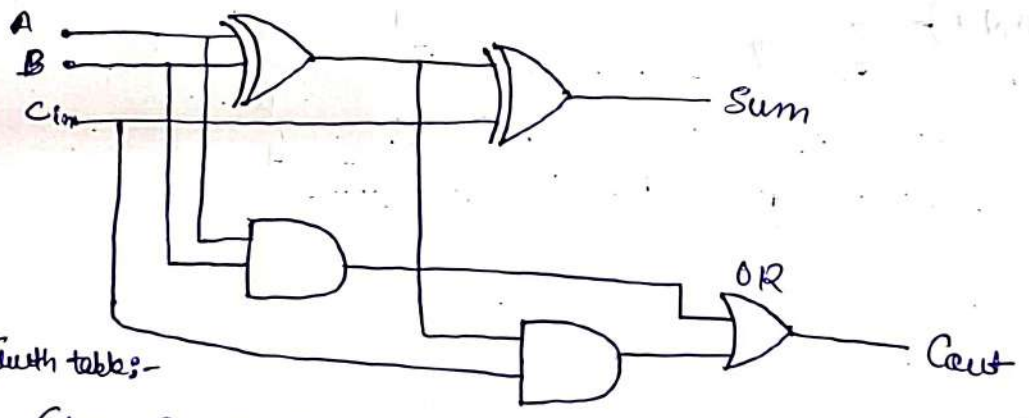
प्रयोग क्र.: _____ दिनांक : _____

Q 5→ Write the VHDL code for the full adder using structural modelling.

Ans]

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity HA is
Port (A,B : in STD_LOGIC;
      S,C : out STD_LOGIC);
end HA;
architecture dataflow of HA is
begin

S <= A XOR B;
C <= A AND B;
end dataflow;
```



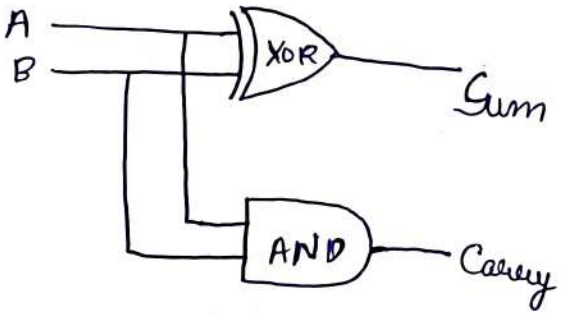
Truth table:-

Cin	B	A	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

शिक्षक के हस्ताक्षर : _____

Q4 VHDL Code for half-adder using Structural mod.
(2).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity HA is
Port (A, B : in STD_LOGIC;
      S, C : out STD_LOGIC);
end HA;
S <= A XOR B;
C <= A AND B;
end dataflow;
```



Truth Table :-

A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Rajshree
PAGE NO.:
DATE:

Q → Diff bet HDL & High level language.

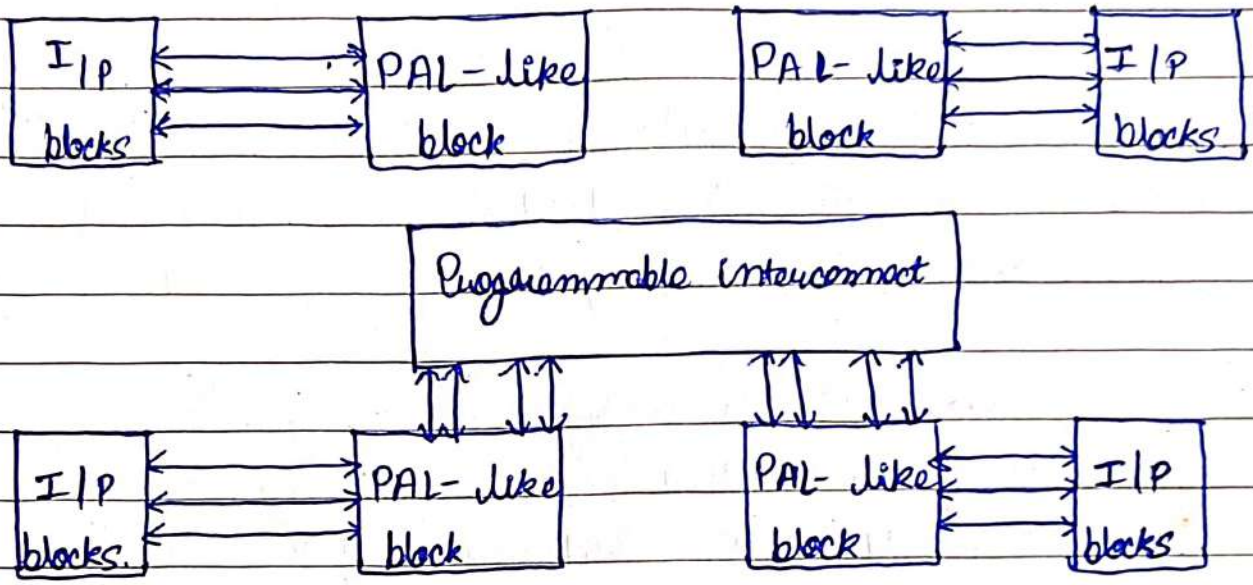
	HDL	HLL
1]	A Specialized computer language used to describe the Structure & behavior of electronic circuits, most commonly, digital logic circuits.	A computer language used to write a set of inst. to allow the CPU to perform a specific task.
2]	More Complex	Not as complex
3]	Verilog & VHDL are common examples.	Java, C, C++, Python, PHP, etc.
4]	Describe the behavior of digital circuits	Helps to develop various applications.

Q → Diff bet CPLD & FPGA.

	CPLD	FPGA
1]	Stands for complex prog. logic device	→ Stands for field prog. gate array.
2]	CPLD provide highest performance, but they also contain fewer registers than FPGA's	→ FPGA's are offered in a wide density range, from few thousand gates to several million gates.
3]	CPLD consume more power than FPGA devices.	→ FPGA consumes less power than CPLD.
4]	Process tech. are EPROM, EEPROM and FLASH.	Process tech. are SRAM, anti-fuse and EEPROM.
5]	CPLD doesn't require any Memory store configuration program	→ FPGA invariably require one or more configuration PROM, depending on the size of FPGA used.
6]	CPLD's can implement large functions in one pass through logic array.	→ FPGA's contain arrays of logic cells & are less partitioned than CPLD's.
7]	The no. of I/P-O/P pins offered by CPLD is significantly higher.	→ A no. of I/P - O/P pins offered on the FPGA are less than CPLD.

• Structure of CPLD :-

A CPLD comprises multiple PAL-like blocks on a single chip with prog. interconnect to connect the blocks.



Q → What are prog. logic devices.

Ans → A PLD is an individual programmable electronic chip which can be used as an element to build digital circuits that can be reconfigured.

Types:-

(A) Prog. Read only memory (PROM):- The I/P is fed into a fixed AND array which acts as the decoder, & then is processed through a programmable OR array before giving the O/P.

(B) Prog. Array Logic (PAL):- It comprises of a prog. AND array & then fixed OR array in that sequence. As such, the O/P of these devices would be the combination of the I/P in the form of Sum of products.

(C) Prog. Logic Array (PLA):- It comprises of two prog. AND & OR arrays one after the other, sandwiched between the I/P & O/P ports.

(D) CPLD

(E) FPGA

Date : Page :

- Concurrent Statements.

These are used to define interconnected blocks & processes that jointly describe the overall behaviour or structure of a design. Concurrent statements execute asynchronously with respect to each other.

Concurrent statements:-

→ Block Statement
→ Process Statement
→ concurrent procedure call statement
→ Concurrent assertion statement.
→ Concurrent signal assignment statement
→ component instantiation statement
→ generate statement.

→ Block statement groups together other concurrent statements.
→ Process statement represents a single independent sequential process.
→ Additional concurrent statements provide convenient syntax for representing simple, commonly occuring forms of processes, as well as for representing structural decomposition and regular descriptions.

Date : Page :

- Sequential Statement

Date : Page :

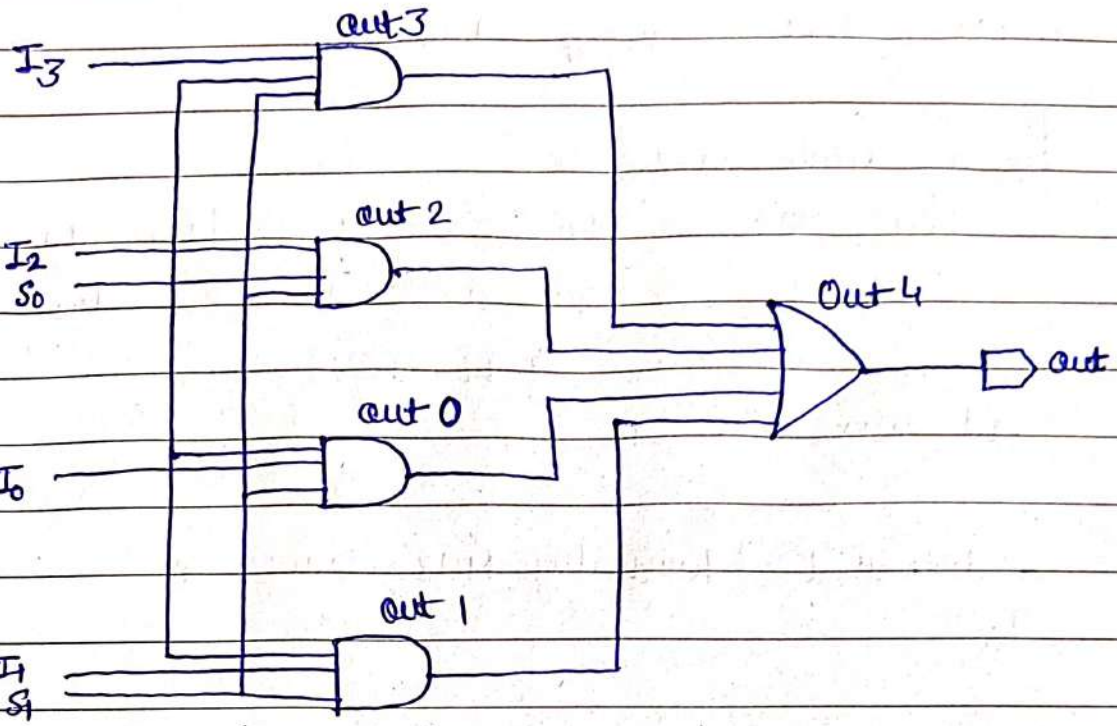
- 4:1 multiplexer using behavioural.

```
entity MUX_SOURCE is
  Port (S: in STD_LOGIC_VECTOR (1 downto 0);
        I: in STD_LOGIC_VECTOR (3 downto 0);
        Y: out STD_LOGIC);
end MUX_SOURCE

architectural behavioural of MUX_SOURCE is
begin

Process (S, I)
begin
if (S<= "00") then
Y<= I(0);
elsei (S<= "01") then
Y<= I(1);
elsei (S<= "10") then
Y<= I(2);
else
Y<= I(3);
end if;
end process;
end behavioural;
```

Date : Page :



TT:-

I_0	I_1	I_2	I_3	S_0	S_1	Y
I_0	X	X	X	0	0	I0
X	I_1	X	X	0	1	I_1
X	X	I_2	X	1	0	I_2
X	X	X	I_3	1	1	I_3