- For every input vector. the code is repe executed until all variables have attained steady value.
- Compiled code simulator is efficient when used for high-level design verification. Inefficiency is incurred by the evaluation of the design when only few inputs are changing.

Q.4(a) (iii) Explain sensitivity list and zero modeling. [4]

(A) Sensitivity List:

- The sensitivity list is a compact way of specifying the set of signals, events on which may resume a process. A sensitivity list is specified right after the keyword process.
- The sensitivity list is equivalent to the wait on statement, which is the last statement of the process statement section.
- Only static signal names, for which reading is permitted, may appear in the sensitivity list of a process, i.e. no function calls are allowed in the list.

Example 1:

```
DFF : PROCESS (CLK,RST)
BEGIN
IF RST = '1'
THEN Q <= '0';
ELSIF (CLK'EVENT) AND (CLK = '1')
THEN Q <= D;
END IF;
END PROCESS DFF;
DFF : PROCESS
BEGIN
IF RST = '1'
THEN Q <= '0';
ELSIF (CLK'EVENT) AND (CLK = '1')
THEN Q <= D;
END IF;
WAIT ON RST, CLK;
END PROCESS DFF;
```

Here, the process is sensitive to the RST and CLK signals, i.e. an event on any of these signals will cause the process to resume. This process is equivalent to the one described in the comment section.

A process with a sensitivity list may not contain any explicit wait statements. Also, if such a process statement is a parent of a procedure, then that procedure may not contain a wait statement as well.

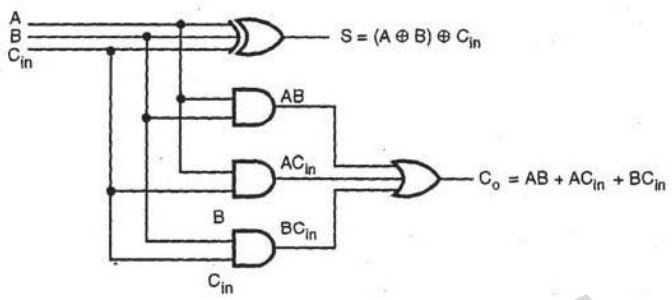
Zero Modelling:

- While describing any system for synthesis circuit delays are determined by the target technology. While writing VHDL for synthesis signal, assignment statements never included a delay assignment.
- All digital circuit elements have a delay [propagation delay] which is very small in terms of nano sec. This nano sec delta delay will have little impact while writing the VHDL code. But for circuit realization this delay must be incorporated. The physical circuits always have finite delays.
- In VHDL zero delay circuits and designs that depends on zero delay components can never be built.
- Simulation deltas are used to order some types of events during simulation. Specifically zero delay events must be ordered to produce consistent results. If they are not properly ordered results can vary between different simulation runs.

Q.4(a) (iv) Draw full Adder using gates and write VHDL code for it. [4]

(A)

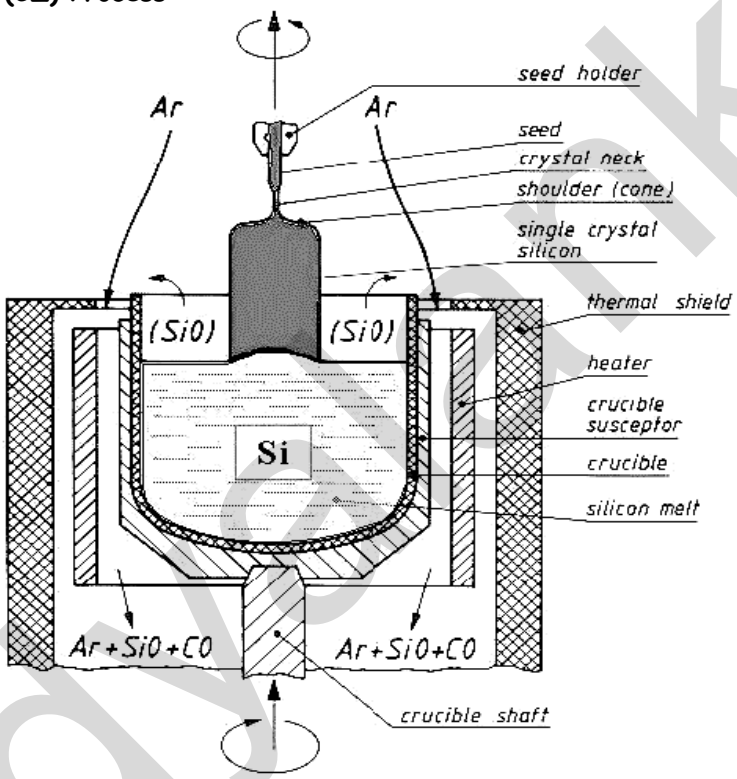
```
library IEEE,
use IEEE.std_logic_1164.all,
entity f_Adder is
port ( a, b,cin: in BIT;
Sum,Carry : out BIT  );
end f_Adder;
architecture dataflow of f_adder is
begin
sum <= A xor B xor Cin;
carry <= (A and B) or (A and Cin) or (Cin and B); end dataflow;
```



Q.4(b) Attempt any ONE questions. [6]

Q.4(b) (i) Explain CZ process for wafer fabrication, with neat diagram. [6]

(A) Czochralski (CZ) Process:



It consists of Quartz crucible, which is surrounded by a graphite radiator. The graphite is heated by radio frequency induction heating and temperature maintained a few degrees above the melting point of silicon (approx. 14250C), the atmosphere just above the polysilicon melt is typically helium or orgon for freezing.

A polycrystalline Si is melted in the crucible and controlled amount of impurities (p type or n type) are added to the melt to provide the crystal with required electrical properties.

After the seed (single crystal silicon piece) is dipped into the melt, the seed is gradually withdrawn vertically from the melt while simultaneously being rotated. The molten polycrystalline silicon melts the tip of the seed and it is withdrawn, refreezing occurs. As the melt freezes, it assumes the single crystal form of the seed. This process is continued until the melt is consumed. The diameter of the ingot (rod of silicon) is determined by the seed withdrawn rate and seed rotation rate.

The produced crystalline silicon rod is then slicing into wafers using cutting tools like diamond blades. Following slicing at least one face of the wafer is polished to flat scratch free mirror finish surface.

Q.4(b) (ii) List and explain the main steps carried in typical n-well, CMOS fabrication process with neat sketches. **[6]**

(A) n-well, CMOS fabrication process : The fabrication steps are as follows :

Step 1 : Formation of n-well regions.
Step 2 : Define nMOS and pMOS active areas.
Step 3 : Field and gate oxidation (thinox)
Step 4 : Form and pattern polysilicon.
Step 5 : p^+ diffusion.
Step 6 : n^+ diffusion.
Step 7 : Contact cuts.
Step 8 : Deposit and pattern metallization.
Step 9 : Over glass with cuts for bonding pads.

The first mask defines the n-well regions. This is followed by a low dose phosphorous implant driver in by a high temperature diffusion step to form the n-well. The well-depth is optimized to ensure against p-substrate to p^+ diffusion breakdown without compromising the n-well to n^+ mask separation. The next steps are to define the devices and diffusion paths, grow field oxide, deposit and pattern the polysilicon, carry out the diffusion, make contact cuts and finally metallize.

Q.5 Attempt any FOUR questions. **[16]**

Q.5(a) Define following terms related to fabrication process. **[4]**

(i) Oxidation **(ii) Diffusion**
(iii) Ion-implantation **(iv) Deposition**

(A) (i) Oxidation:

Oxidation is a process by which a layer of silicon dioxide is grown on the surface of a silicon wafer.

The oxidation of silicon is necessary throughout the modern integrated circuit fabrication process.

(ii) Diffusion:

It is the process by which impurities may be introduced into selected region of a semiconductor. In the silicon technology diffusion allows formation of sources and drains for metal-oxidesemiconductor devices. It is extensively used because it is ideally adopted to batch processes where many slices are handled in single operation. It does not produce crystal damage, thus high quality junctions with minimum leakage current can be made easily by this method. It is achieved by heating the wafer to a high temperature and passing a gas containing the desired impurity.

(iii) Ion- Implantation:

Ion implantation can be defined as the process by which impurity ions are accelerated to high velocity and physically lodged into the target material.

(iv) Deposition:

Deposition is a process followed by an implantation step to reduce poly resistance.

Q.5(b) Define the following terms. **[4]**

(1) Metastability **(2) Noise Margin**

(A) Metastability:

Metastability in electronics is the ability of a non- equilibrium electronic state to persist for a long (and theoretically unboundable) period of time.

(OR)

A metastable state is half way between logic '0' and logic '1'. It is undefined state.

Noise Margin:

It is a measure of noise immunity of a gate or circuit (noise immunity is the ability of a gate or circuit to tolerate any noise present in a signal without performing a wrong operation).

Q.5(c) Compare BJT and CMOS. **[4]**

(A)

Sr. No.	BJT	CMOS
1.	High power dissipation	Low static power dissipation
2.	Low input impedance	High input impedance
5.	Low packing density	High packing density
4.	Low delay sensitive to load	High delay sensitive to load
5.	High output drive current	Low output drive current
6.	Essentially unidirectional	Bidirectional capability
7.	It is not an ideal switching device	It is an ideal switching device
8.	Current driven	Voltage driven

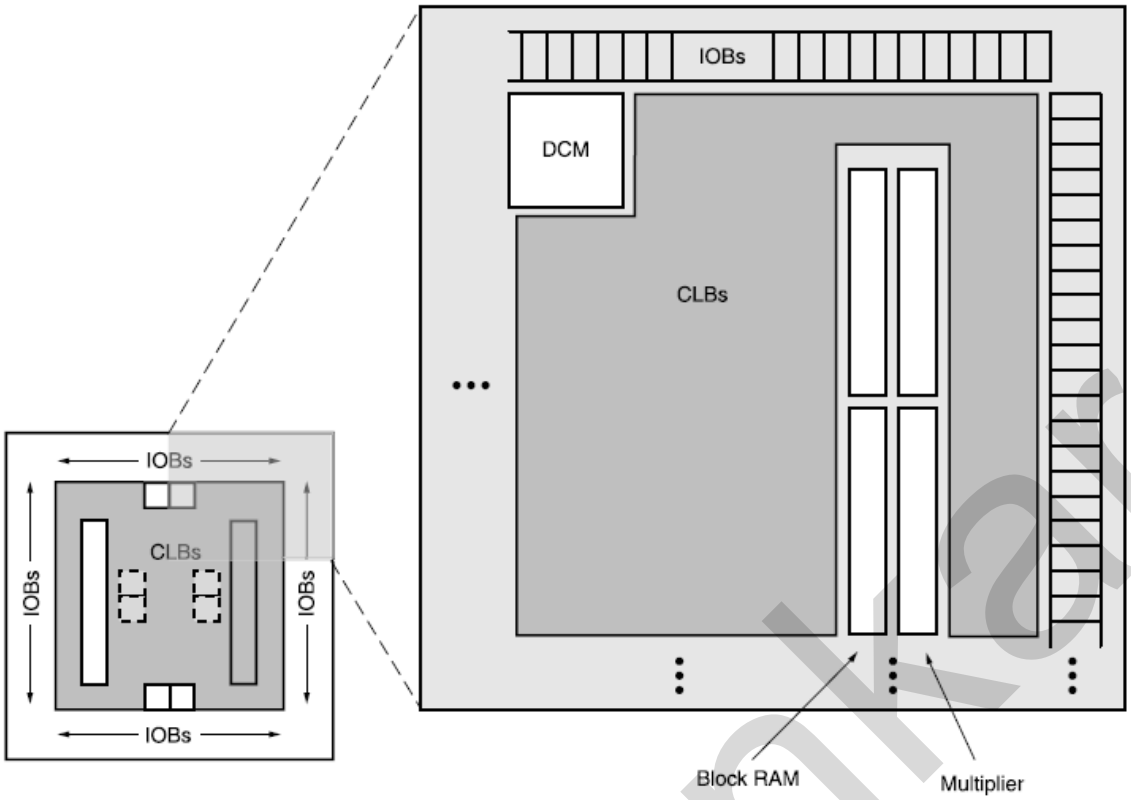
Q.5(d) Write VHDL program to Implement JK flip-flop with + ve edge trigger. **[4]**

(A) **Note:** Program in any modeling should be considered and marks to be given

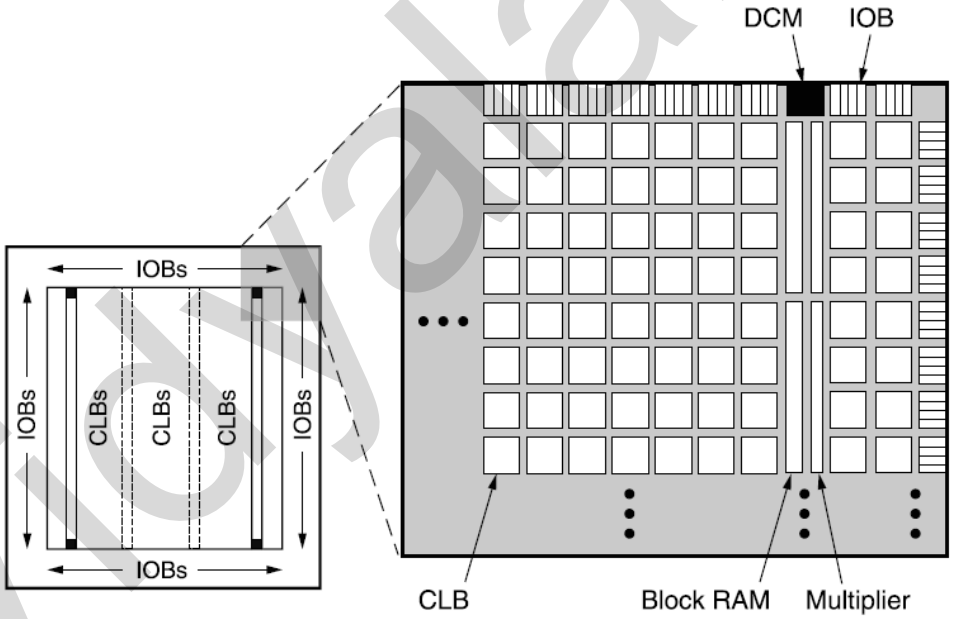
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY JKff IS
PORT (clk: IN STD_LOGIC;

        J, K: IN STD_LOGIC;
        Q, Qbar: OUT STD_LOGIC);
END JKff;
ARCHITECTURE BEHAVIORAL OF JKff_arch IS
BEGIN
PROCESS (clk)
BEGIN
        if (clk'event and clk = '1') then
        if (J = '0' and K = '0') then
            Q < = Q;
        elsif (J = '0' and K = '1') then
            Q < = '0';
        elsif (J = '1' and K = '0') then
            Q < = '1';
        elsif (J = '1'and K = '1') then
            Q < = not Q;
        End if
    End if
End process
Qbar < = not Q;
End behavioural JKff_arch
```

Q.5(e) Explain the basic architecture of SPARTAN – 3 FPGA series. **[4]**

(A)



OR



The Spartan-3 family architecture consists of five fundamental programmable functional elements:

Configurable Logic Blocks (CLBs): Contain flexible Look-Up Tables (LUTs) that implement logic plus storage elements used as flip-flops or latches. CLBs perform a wide variety of logical functions as well as store data.

Input/ Output Blocks (IOBs): Control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Double Data-Rate (DDR) registers are included.

Block RAM : Provides data storage in the form of 18-Kbit dual-port blocks.

Multiplier Blocks : Accept two 18-bit binary numbers as inputs and calculate the product.

Digital Clock Manager (DCM): Blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase-shifting clock signals.

Q.5(f) Compare FPGA and CPLD. **[4]**

(A)

Sr. No.	FPGA	CPLD
1	It is field programmable gate arrays.	It is complex programmable logic device.
2	Capacity is defined in terms of number of gates available.	Capacity is defined in terms of number of macro-cells available.
3	FPGA consumes less power than CPLD	CPLD consumes more power than FPGA devices.
4	Numbers of input and output pins on FPGA are less than CPLD.	Numbers of input and output pins on CPLD are high.
5	FPGA is suitable for designs with large number of simple blocks with few numbers of inputs.	CPLD are ideal for complex blocks with large number of inputs.
6	FPGA based designs require more board space and layout complexity is more.	CPLD based designs need less board space and less board layout complexity.
7	It is difficult to predict the speed performance of design.	It is easier to predict speed performance of design.
8.	FPGA are available in wide density range.	CPLDs contain fewer registers but have better performance.

Q.6 Attempt any FOUR questions. **[16]**

Q.6(a) Explain with syntax. **[4]**

(1) Entity **(2) Architecture**

(A) Entity:

- A VHDL Entity specifies the name of the entity, the ports of the entity, and other entity related information.
- All designs are created using one or more entities.
- **Syntax:**

  Entity <entity_name> is
  Generic (<generic_list>);
  Port (<port_list>);
  End <entity_name>

- The keyword entity signifies that this is the start of the entity statement. The standard type provided is BIT.
- **Example**

```
ENTITY mux is
PORT (a, b, c, d: IN_BIT;
      s0, s1: IN_BIT;
      Y: OUT_BIT);
END mux;
```

- Name of the user created object is mux. The name of the entity is mux.
- The entity has seven ports in the PORT Clause.
- Six of them are input ports and one is output port which is notified as IN and OUT respectively.
- The four data input ports (a, b, c, d) and two select input ports (s0, s1) and one output port (y) are of type BIT.
- The entity describes the interface to the outside world. It specifies the number of ports, direction of the ports, type of ports, etc.

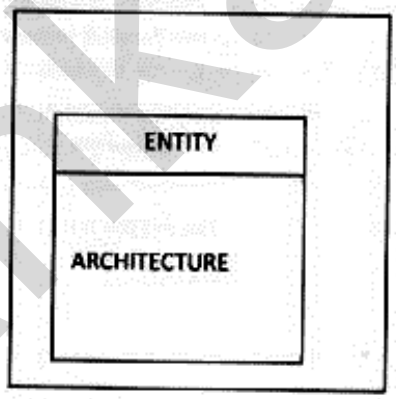
Architecture:

- The entity describes the interface to the VHDL model.
- The architecture specifies behaviour, function, interconnection or relation between input and output of an entity.
- Syntax:

```
Architecture architecture_name of entity_name is
architecture_declarations
Begin
concurrent_statements
End [architecture] [architecture_name];
```

- It describes the contents of an entity. The reason for connection between the architecture and the entity is that an entity can have multiple architectures describing the behaviour of the entity.
- The keyword ARCHITECTURE signifies that this statement describes architecture for an entity.
- The statement of the architecture starts with the keyword BEGIN and ends with the END netlist statement.
- **Example:**

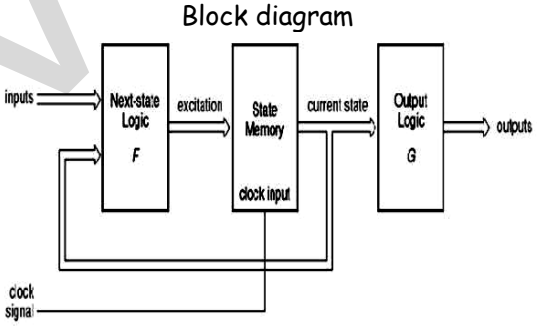
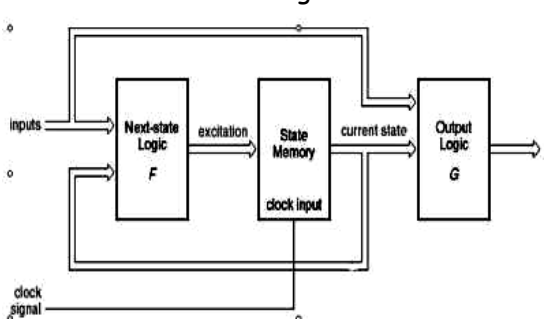
```
Architecture AND1 of ANDGATE is
--declarations
Begin
--statements
Y <= A AND B;
End architecture AND1;
```



VHDL program file structure

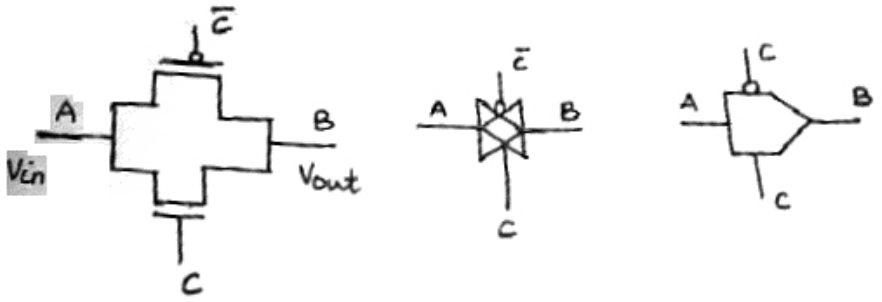
Q.6(b) Compare Mealy Machine with Moore Machine. [4]

(A)

Moore machine	Mealy Machine
Definition :output is function of state of machine	Definition :output is function of state of machine and present input condition
Requires more number of states	Requires less number of states
Faster	slower
Design simple	Design complex
Output in state	Output is at the time of state transition
Block diagram 	Block diagram 

Q.6(c) Draw and explain working of CMOS Transmission gates. [4]

(A)

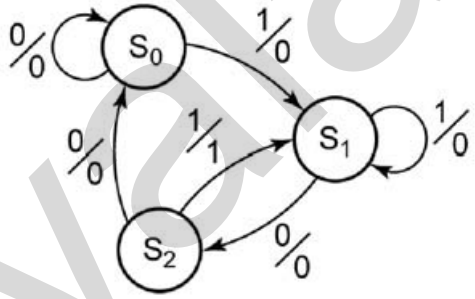


- It consists of one nMOS and one pMOS transistor in parallel.
- The gate voltages, applied to these two transistors are also set to be complementary signals.
- The CMOS Transmission gate operates as a bidirectional switch between the nodes A & B which is controlled by C.
- If the control signal C is logic high, VDD, then both the transistors are turned ON and provides a low resistance current path between the nodes A & B.
- If C is low, then both the transistors are off & path between A & B is open circuit.
- This condition is called high impedance state.

Q.6(d) Draw and Implement the T flip-flop using Moore machine. [4]

(A) Let's consider a sequence detector 101 which can be implemented with T flip flop using Moore machine.

The state diagram will be



Transition table will be

Output Table

Present State	Next State $X=0$	Next State $X=1$	Present Output $X=0$	Present Output $X=1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_2	S_0	S_1	0	1

Transition Table

AB	A^+B^+ $X=0$	A^+B^+ $X=1$	Z $X=0$	Z $X=1$
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

State Mapping
S0 = 00
S1 = 01
S2 = 10

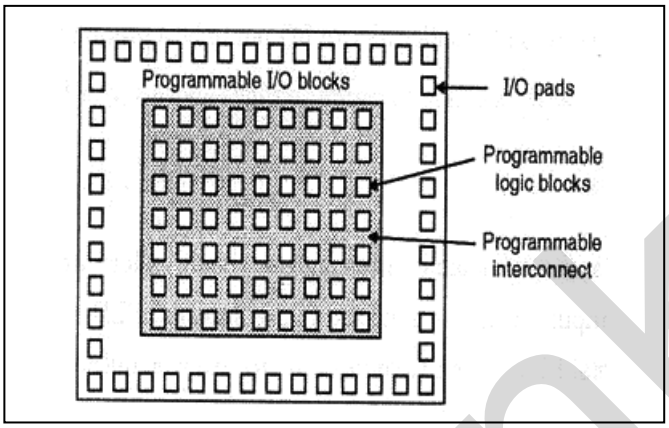
K – maps for implementing design using T flip-flop (1 mark)

X \ AB	00	01	11	10
0	0	1	–	1
1	0	0	–	1

X \ AB	00	01	11	10
0	0	1	–	0
1	1	0	–	1

Q.6(e) Draw the general FPGA chip architecture and explain the same. **[4]**

(A)



Field programmable Gate Arrays (FPGA):

- A field programmable gate array (FPGA) ha large number of programmable logic blocks that are individually smaller than a PLD. The basic structure of a FPGA is sown in figure.
- The programmable logic blocks are arrange in the matrix form with programmable interconnections and the entire array is surrounded by programmable I/O blocks. Each logic block is less capable than a typical PLD, but it has lot more logic blocks than a CPLD of the same size.

Q.6(f) Compare software and hardware description languages. **[4]**

(A)

Sr. No.	Software language	Hardware Description Language
1.	In a software language, all assignments are sequential. That means the order in which the statements appear is significant because they are executed in that way.	The events (change in value) in hardware are concurrent, and they must be represented in that way.
2.	A software language cannot be used to describe hardware and so a hardware language is required.	A hardware language is used to describe the hardware.
3.	In software language, the statements are evaluated sequentially.	In VHDL, concurrent statements are defined to take care of concurrency hardware.
4.	We get different results when the order is changed.	The HDL is always concurrent.

❑ ❑ ❑ ❑ ❑