Explain the data objects used in VHDL.

Data Objects

- A data object holds a value of a specified type. It is created by means of an object declaration. An example is

  variable COUNT: INTEGER;

- This results in the creation of a data object called COUNT which can hold integer values. The object COUNT is also declared to be of *variable class.*

DATA OBJECTS & CLASSES

- Every data object belongs to one of the following three classes:

1. *Constant: An object of constant class can hold a single value of a given type. This value is assigned to the object before simulation starts and the value cannot be changed during the course of the simulation.*

DATA OBJECTS & CLASSES Cont..

2. *Variable: An object of variable class can also hold a single value of a given type. However in this case, different values can be assigned to the object at different times using a variable assignment statement.*

3. *Signal: An object belonging to the signal class has a past history of values, a current value, and a set of future values. Future values can be assigned to a signal object using a signal assignment statement.*

DATA OBJECTS & CLASSES Cont..

- An *object declaration is used to declare an object, its type, and its class, and optionally assign it a value. Some examples of object declarations of various types and classes follow.*

Constant Declarations

Examples of constant declarations are-

constant RISE_TIME: TIME := 10ns;

constant BUS_WIDTH: INTEGER := 8:

DATA OBJECTS & CLASSES Cont..

- Variable Declarations Examples of variable declarations are-

variable CTRL_STATUS: BIT_VECTOR(10 downto 0);

variable SUM: INTEGER range 0 to 100 := 10;

variable FOUND, DONE: BOOLEAN;

DATA OBJECTS & CLASSES Cont..

Signal Declarations

Here are some examples of signal declarations.

signal CLOCK: BIT;

signal DATA_BUS: BIT_VECTOR(0 to 7);

signal GATE_DELAY: TIME := 10 ns;

Compare signals and variables.

	Signal	Variable
1.	Signals are declared using signal declaration statements.	A variable is declared within a block, process, procedure or function.
2.	The value is assign to signal using "< =" assignment operator.	Variable is assign value using ": =" assignment operator.
3.	Signals may be of any data type.	Variable can be of any scalar data type.
4.	Signals are used to describe the hardware.	Variables are used to describe the behaviour of the digital system.
5.	Signals may be slower.	Variables allow quick simulation.

Explain Package declaration with example.

Package Declaration

- A package declaration is used to store a set of common declarations like components, types, procedures, and functions.

- These declarations can then be imported into other design units using a use clause.

Example of Package Declaration

```
package EXAMPLE_PACK is
type SUMMER is (MAY, JUN, JUL, AUG, SEP);

component D_FLIP_FLOP
    port (D, CK: in BIT; Q, QBAR: out BIT);
end component;

constant PIN2PIN_DELAY: TIME := 125 ns;

function INT2BIT_VEC (INT_VALUE: INTEGER)
    return BIT_VECTOR;
end EXAMPLE_PACK;
```

Package Declaration cont..

- Assume that this package has been compiled into a design library called DESIGN_LIB. Consider the following clauses associated with an entity declaration.

library DESIGN_LIB;

use DESIGN_LIB.EXAMPLE_PACK.all;

entity RX is . . .

Package Declaration cont..

- It is also possible to selectively import declarations from a package declaration into other design units. For example:

```
library DESIGN_LIB;
use DES[GN_LIB.EXAMPLE_PACK.D_FLIP_FLOP;
use DESIGN_LIB.EXAMPLE_PACK.PIN2PIN_DELAY;
 architecture RX_STRUCTURE of RX is . . .
```

Determine any five capabilities of VHDL.

VHDL's Capabilities & features

- The language can be *used as an exchange medium between chip vendors and CAD tool users.*
- The language supports *hierarchy.*
- The language *supports flexible design methodologies: top-down, bottom-up, or mixed.*
- The language is *not technology-specific.*
- It supports both *synchronous and asynchronous timing models.*

VHDL's Capabilities & features cont..

- The language is publicly available, human readable, machine readable.
- Various digital modeling techniques such as finite-state machine descriptions, algorithmic descriptions, and boolean equations can be modeled using the language.
- It is an *IEEE and ANSI standard, and therefore, models described using this language are portable.*

VHDL's Capabilities & features cont..

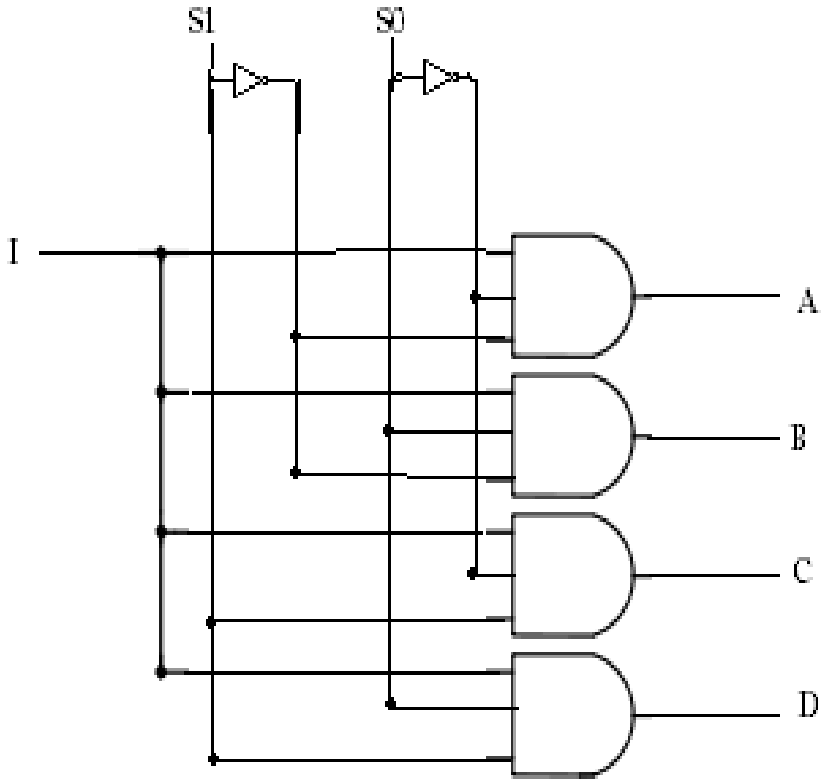
- The language supports *three basic different description styles: structural, dataflow, and behavioral.*
- It supports a wide *range, of abstraction levels ranging from abstract behavioral descriptions to very precise gate-level descriptions.*
- *Arbitrarily large designs can be modeled using the language.*

Write down the VHDL coding of 1X4 Demultiplexer using behavioral style of modeling.

Demultiplexer (1X4)

- take one data input and a number of selection inputs, and they have several outputs

I	S1	S0	ABCD
0	X	X	0000
1	0	0	1000
1	0	1	0100
1	1	0	0010
1	1	1	0001



Behavioral using IF statement Program:

```
entity abcd is
    Port ( i , s0, s1 : in  bit;
                 y : out  bit_vector (0 to 3));
 end abcd;
architecture beh_f  of  abcd  is
Process (i, s0, s1)
Begin
   if (s1 = '0' and s0 ='0' ) then
      y<= "1000";
   elsif (s1 = '0' and s0 ='1' ) then
      y<= "0100";
   elsif (s1 = '1' and s0 ='0' ) then
      y<= "0010";
   elsif (s1 = '1' and s0 ='1' ) then
      y<= "0001";
  end if;
End process;
End beh_f;
```

Behavioral using CASE statement Program:

```
entity abcd is
    Port ( I : in  bit;
             s: in bit_vector (0 to 1);
                y : out  bit_vector (0 to 3));
 end abcd;
architecture beh_f  of  abcd  is
Process (I, s)
Begin
   case s is
       when “00” => y <= “1000”;
       when “01” => y <= “0100”;
       when “10” => y <= “0010”;
       when “11” => y <= “0001”;
    end case;
End process;
End beh_f;
```

Define configuration declaration in VHDL.

Example of Configuration Declaration

- Consider the following configuration declaration for the HALF_ADDER entity.

library CMOS_LIB, MY_LIB;

configuration HA_BINDING of HALF_ADDER is

for HA-STRUCTURE

for X1:XOR2

Example of Configuration Declaration cont..

```
        use entity CMOS_LIB.XOR_GATE(DATAFLOW);
    end for;
    for A1:AND2
        use configuration MY_LIB.AND_CONFIG;
    end for;
end for;
end HA_BINDING;
```

What do you mean by sensitivity-list?

- Sensitivity list: It contains a set of signals to which the process is sensitive.
- Syntax:
  - (signal name a, signal name b)
- The sensitivity list is a way of specifying signals, each time an event occurs on any of the signals in the sensitivity list, the statements within the process are executed in a sequential order.

What is VHDL and how do we can use it to describe a model for a digital H/W device?

Hardware Abstraction

- VHDL is used to describe a model for a digital hardware device.
- This model specifies the external view of the device and one or more internal views.
- The internal view of the device specifies the functionality or structure, while the external view specifies the interface of the device through which it communicates with the other models in its environment.

Hardware Abstraction cont..

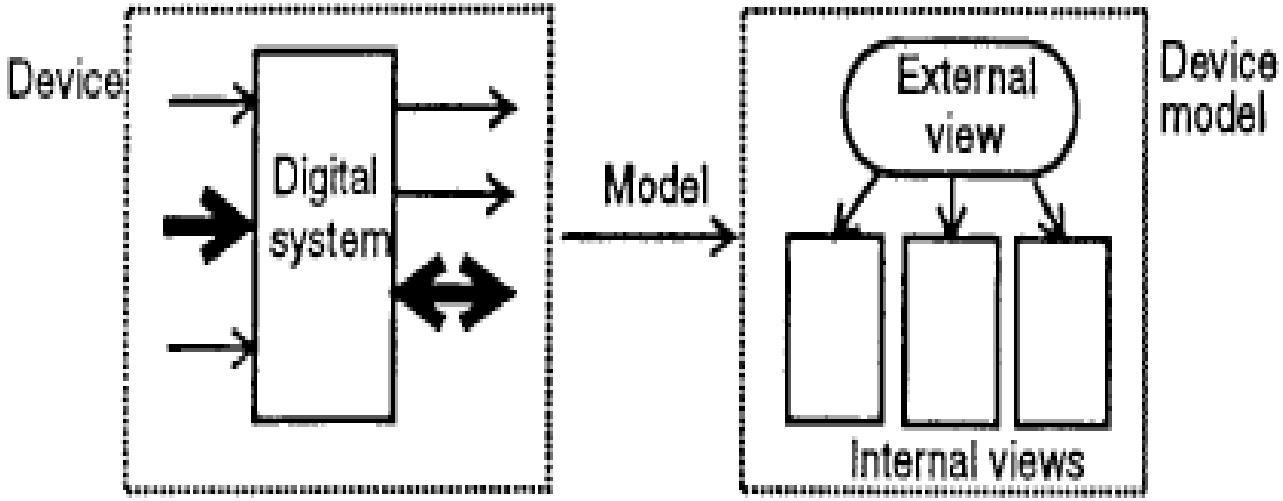


Figure 1.1 Device versus device model.

Hardware Abstraction cont..

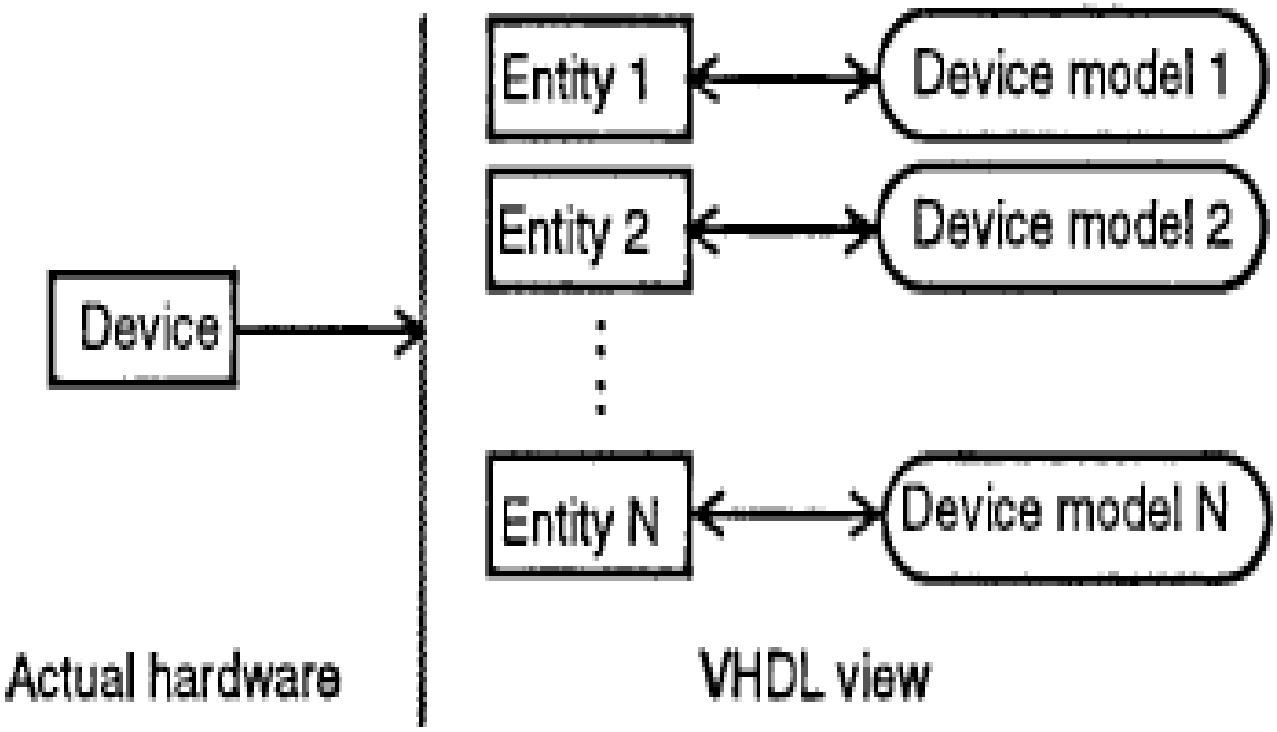


Figure 1.2 A VHDL view of a device.

Hardware Abstraction cont..

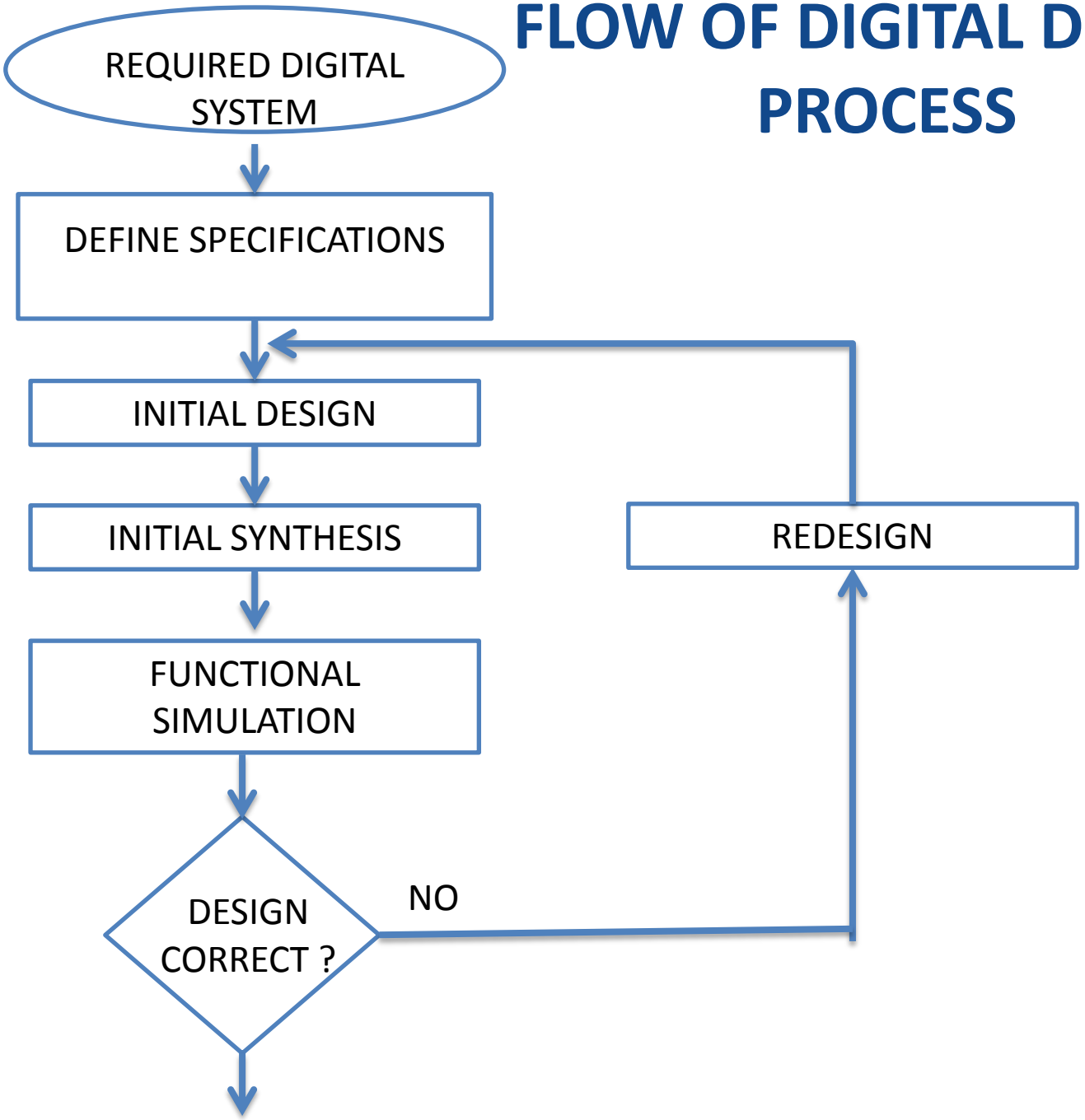
- The entity is thus a hardware abstraction of the actual hardware device. Each entity is described using one model that contains one external view and one or more internal views. At the same time, a hardware device may be represented by one or more entities.

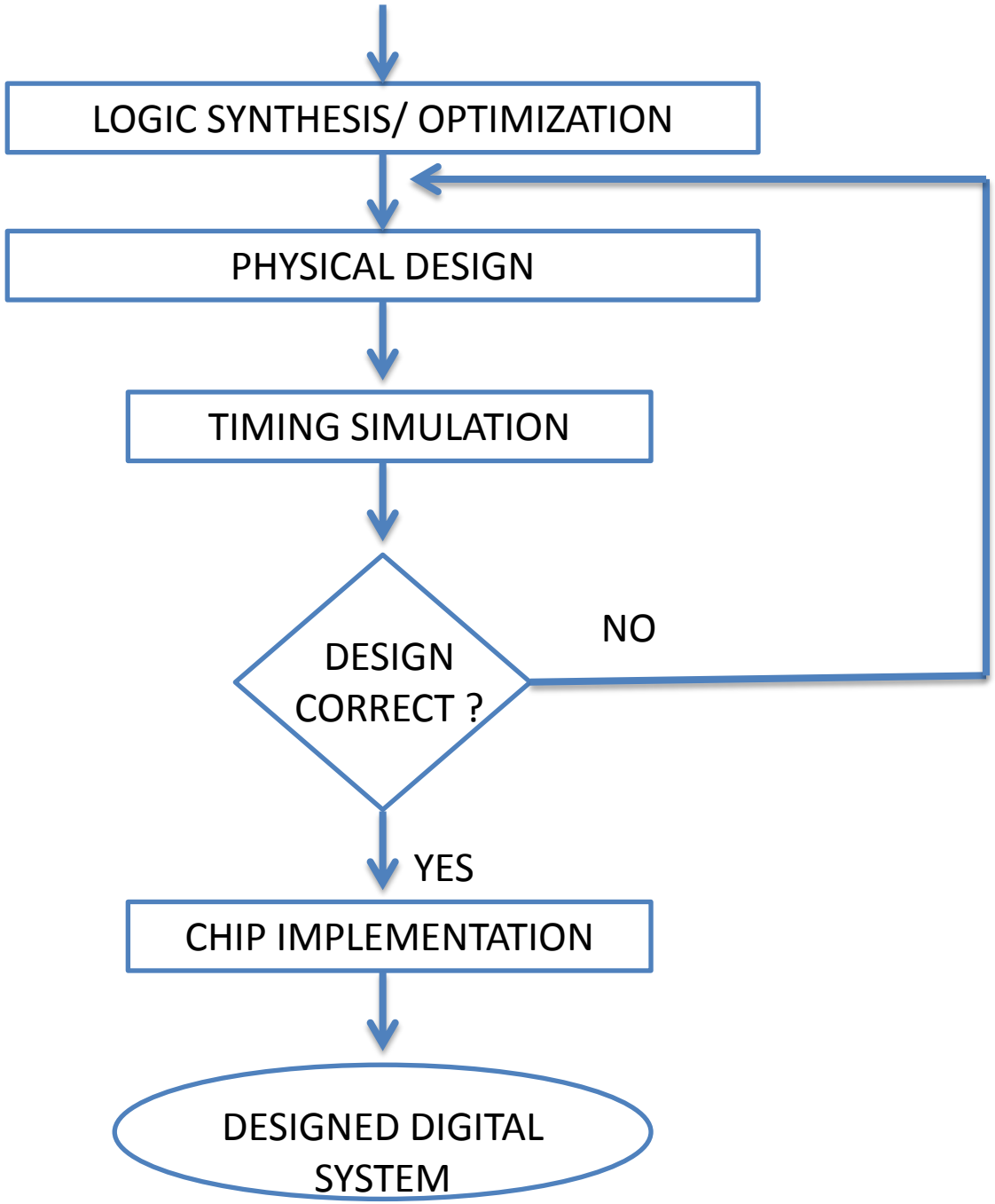
Write a short note on Computer-aided design tools for digital systems.

Introduction to Computer-aided design tools for digital systems

- The design methods which make the use of computer are known as Computer Aided Design methods.
- CAD tools refer to software tools that aid the development of circuits, systems and other things.
- Different CAD Tools for digital systems are:
  - Schematic entry tools
  - HDLS
  - HDL compilers, simulators and synthesis tools
  - Simulators
  - Test benches
  - Timing Analyzers and verifiers.

FLOW OF DIGITAL DESIGN PROCESS




LOGIC SYNTHESIS/ OPTIMIZATION
PHYSICAL DESIGN
TIMING SIMULATION
DESIGN CORRECT ?
NO
YES
CHIP IMPLEMENTATION
DESIGNED DIGITAL SYSTEM

Define Data-Types used in VHDL. Explain the Scalar type and Composite type in detail with example.

Data Types

- Every data object in VHDL can hold a value that belongs to a set of values.
- This set of values is specified by using a *type declaration.*
- *A type is a name that has associated with it a set of values and a set of operations.*

Types of Data Types in VHDL

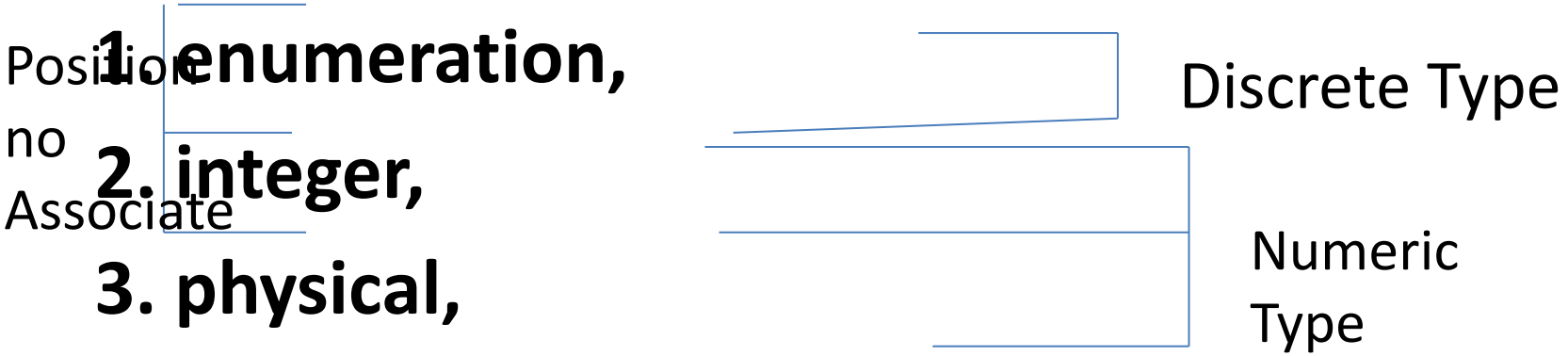
1. ***Scalar types: Values belonging to these types appear in a sequential order.***
2. ***Composite types: These are composed of elements of a single type (an array type) or elements of different types (a record type).***
3. ***Access types: These provide access to objects of a given type (via pointers).***
4. ***File types: These provides access to objects that contain a sequence of values of a given type.***

It is possible to derive restricted types, called subtypes, from other predefined or user-defined types.

1. Scalar Types

- There are four different kinds of scalar types. These types are

1. **enumeration,**
2. **integer,**
3. **physical,**
4. **floating point.**



- BIT- 0 and 1
- BOOLEAN – True and False

- SEVERITY_LEVEL- NOTE, WARNING, ERROR, FAILURE

- FILE_OPEN_KIND- READ_MODE, WRITE_MODE, APPEND_MODE

- FILE_OPEN_STATUS- OPEN_OK, STATTUS_ERROR, NAME_ERROR, MODE_ERROR

Enumeration Types

- An enumeration type declaration defines a type that has a set of user-defined values consisting of identifiers and character literals. Examples are –

- **Type MVL is ('U','0','1','Z);**
- **type MICRO_OP is (LOAD, STORE, ADD, SUB, MUL, DIV);**
- **subtype ARITH_OP is MICRO_OP range ADD to DIV;**

Enumeration Types cont..

- Examples of objects defined for these types are
- **signal CONTROL_A: MVL;**

- MVL is an enumeration type that has the set of ordered values, 'U', '0', '1', and 'Z'.

- The values of an enumeration type are called enumeration literals. For example, consider the following enumeration type declaration.
- **type CAR_STATE is (STOP, SLOW, MEDIUM, FAST);**

Integer Types

- An integer type defines a type whose set of values fall within a specified integer range. Examples of integer type declarations are
- **type INDEX is range 0 to 15;**
- **type WORD_LENGTH is range 31 downto 0;**
- **subtype DATA_WORD is WORD_LENGTH range 15 downto 0;**

Integer Types cont..

- Some object declarations using these types are
- **constant MUX_ADDRESS: INDEX := 5;**
- **signal DATA_BUS: DATA_WORD;**
- Values belonging to an integer type are called *integer literals. Examples of integer literals are*
- 56349 6E2 0 98_71_28

- INTEGER is the only predefined integer type of the language. The range of the INTEGER type cover the range -(2^ 31 - 1) to +(2^31 - 1).

Floating Point Types

- A floating point type has a set of values in a given range of real numbers. Examples of floating point type declarations are
- **type TTL_VOLTAGE is range -5.5 to -1.4;**
- **type REAL_DATA is range 0.0 to 31.9;**

- An example of an object declaration is
- **variable LENGTH: REAL_DATA range 0.0 to 15.9;**

Floating Point Types cont..

- *Floating -point literals are values of a floating point type. Examples of floating point literals are*
- 16.26 0.0 0.002 3_1.4_2

- The only predefined floating point type is REAL.

Physical Types

- A physical type contains values that represent measurement of some physical quantity, like time, length, voltage, and current. Values of this type are expressed as integer multiples of a base unit. An example of a physical type declaration is

```
type CURRENT is range 0 to 1 E9
    units
        nA;                     -- (base unit) nano-ampere
        uA = 1000 nA;           -- micro-ampere
        mA = 1000 μA;           --milli-ampere
        Amp = 1000 mA;          -- ampere
    end units;
```

Composite Types

- A composite type represents a collection of values. There are two composite types: an array type and a record type.
- An array type represents a collection of values all belonging to a single type.
- Record type represents a collection of values that belong to different types.

Array Types

- Examples of array type declarations are-

- type ADDRESS_WORD is array (0 to 63) of BIT;
- type DATA_WORD is array (7 downto 0) of MVL;

- There are two predefined array types in the language, STRING and BIT_VECTOR. STRING is an array of characters while BIT_VECTOR is an array of bits.

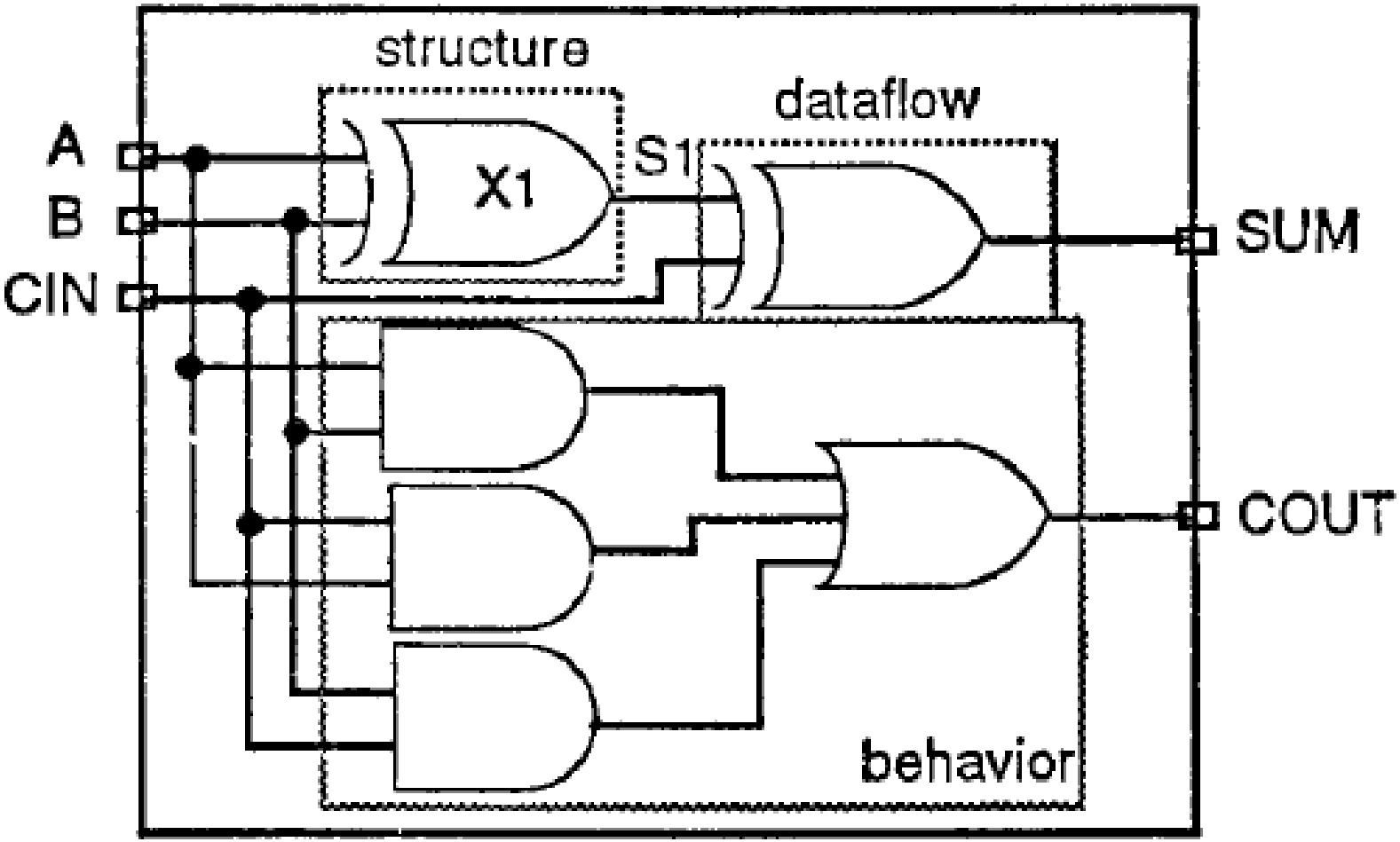
Record Types

- An example of a record type declaration is

```
type MODULE is
    record
        SIZE: INTEGER range 20 to 200;
        CRITICAL_DLY: TIME;
        NO_INPUTS: PIN_TYPE:
        NO_OUTPUTS: PIN_TYPE;
end record;
```

Write down the VHDL code for 1-bit full adder with diagram using mixed style of modeling.

Mixed Style of Modeling



```
entity FULL_ADDER is
port (A, B, CIN: in BIT; SUM, COUT: out BIT);
end FULL_ADDER;
architecture FA_MIXED of FULL_ADDER is
component XOR2
        port (A, B: in BIT; Z: out BIT);
end component;
signal S1: BIT;
begin
X1: XOR2 port map (A, B, S1 );                          - structure.
process (A, B, CIN)                                  - behavior.
        variable T1, T2, T3: BIT;
begin
        T1 :=A and B;
        T2 := B and CIN;
        T3:=A and CIN;
        COUT <= T1 or T2 or T3;
end process;
SUM <= S1 xor CIN;                                      - dataflow.
end FA_M!XED;
```

What do you mean by Delay? Explain different types of Delay used in VHDL.

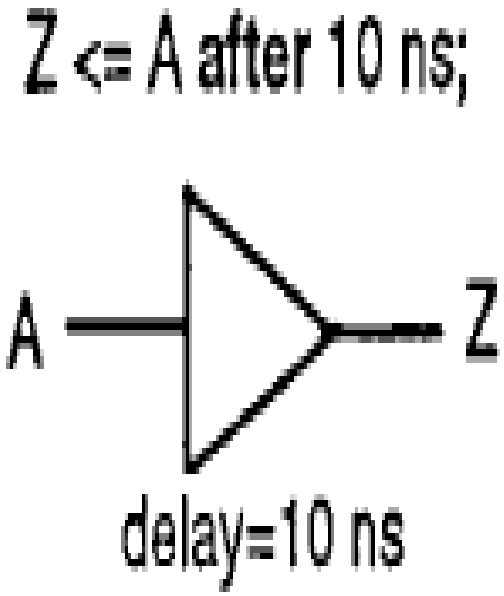
Types of Delays or Delay Models

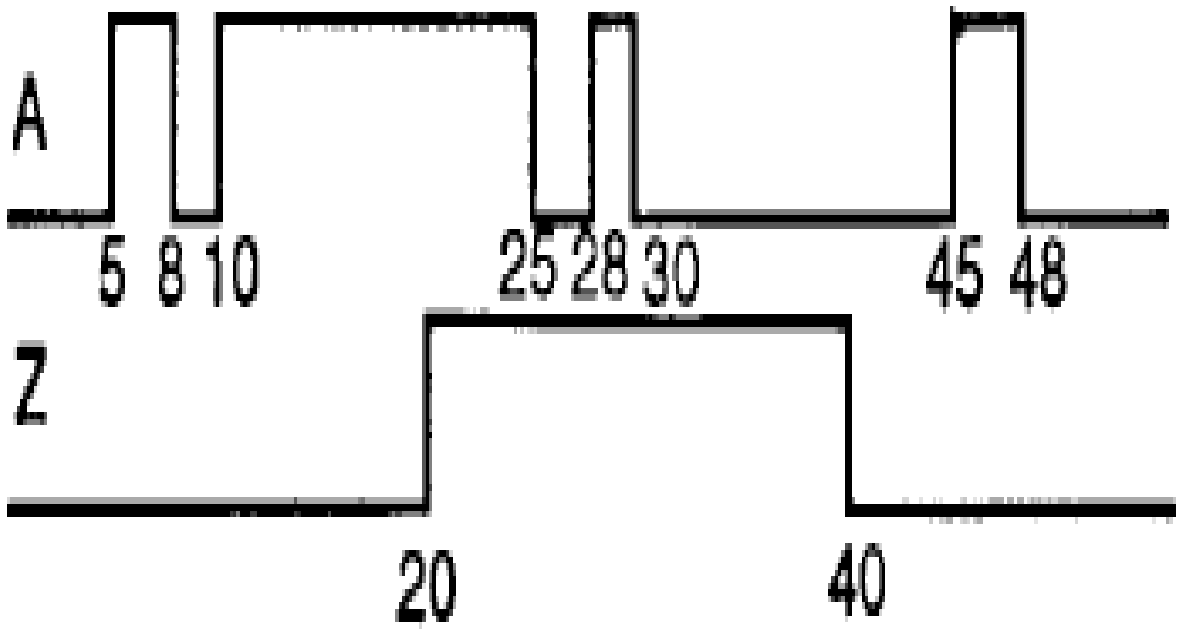
- There are two types of delay that can be applied when assigning a time/value of a signal.
  - Inertial Delay
  - Transport Delay

Inertial Delay Model

- *Inertial delay models the delays often found in switching circuits.*
- *It represents the time for which an input value must be stable before the value is allowed to propagate to the output.*
- *In addition, the value appears at the output after the specified delay. If the input is not stable for the specified time, no output change occurs.*

Inertial Delay Model cont..





Inertial Delay Model cont..

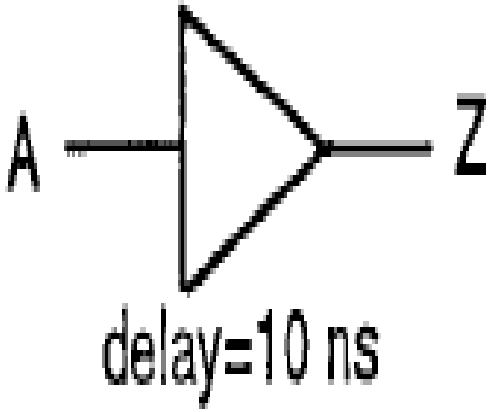
- Since inertial delay is most commonly found in digital circuits, it is the default delay model.
- This delay model is often used to filter out unwanted spikes on signals.

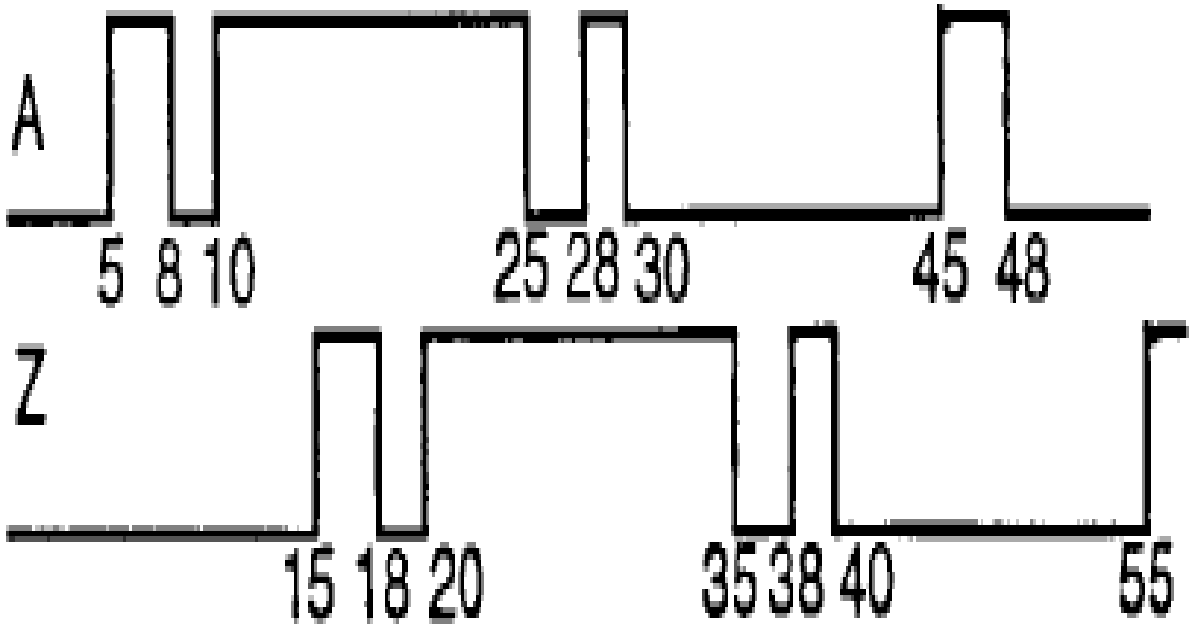
Transport Delay Model

- *Transport delay models the delays in hardware that do not exhibit any inertial delay.*
- *This delay represents pure propagation delay, that is, any changes on an input is transported to the output, no matter how small, after the specified delay.*
- *To use a transport delay model, the keyword transport must be used in a signal assignment statement.*

Transport Delay Model cont..

Z <= transport A after 10 ns;





Transport Delay Model cont..

- Ideal delay modeling can be obtained by using this delay model.
- In this case, spikes would be propagated through instead of being ignored as in the inertial delay case.

What do you mean by overloading? Explain different types of overloading with example

OVERLOADING

Overloading in VHDL is of two types.

- Subprogram Overloading
- Operator Overloading

Subprogram Overloading

- Sometimes it is convenient to have two or more subprograms with the same name. In such a case, the subprogram name is said to be *overloaded. For example, consider the following two declarations.*
- **function COUNT (ORANGES: INTEGER) return INTEGER;**
- **function COUNT (APPLES: BIT) return BIT;**

Subprogram Overloading Cont..

- Both functions are overloaded since they have the same name, COUNT, and have different parameter types. When a call to either function is made, it is easily possible to identify the exact function to which the call was made from the type of the actual parameters passed. For example, the function call

COUNT(20)

refers to the first function since 20 is of type INTEGER, while the function call

COUNT('1')

refers to the second function, since the type of actual parameter is BIT.

Subprogram Overloading Cont..

- It is also possible for two subprograms to have the same parameter types and result types but have a different number of parameters.
- Here is an example of such a set of functions that determine the smallest value from a set of 2,4, or 8 integers.

function SMALLEST (A1, A2: INTEGER) return INTEGER;

function SMALLEST (A1, A2, A3, A4: INTEGER) return INTEGER;

function SMALLEST (A1, A2, A3, A4, A5, A6, A7, A8: INTEGER)

return INTEGER;

Subprogram Overloading Cont..

A call such as

. . . SMALLEST (4, 5) . . .

refers to the first function, while the function call

. . . SMALLEST (20, 45,52,1,89,67,91,22)...

refers to the third function

Operator Overloading

- Operator overloading is one of the most useful features in the language.
- When a standard operator symbol is made to behave differently based on the type of its operands, the operator is said to be overloaded.
- The need for operator overloading arises from the fact that the predefined operators in the language are defined for operands of certain predefined types.

Operator Overloading cont..

The operator in the expression

S1 and S2

where S1 and S2 are of type MVL, would then refer to the and operation that was defined by the model writer as a function. The operator in the expression

CLK1 **and CLK2**

where CLK1 and CLK2 are of type BIT, would refer to the predefined and operator.

Operator Overloading cont..

- Function bodies are written to define the behaviour of overloaded operators. Such a function has, at most, two parameters; the first one refers to the left operand of the operator and the second parameter, if present, refers to the second operand. Here are some examples of function declarations for such function bodies.
- **type MVL is ('U', '0', '1', 'Z');**
- **function "and" (L, R: MVL) return MVL;**
- **function "or" (L, R: MVL) return MVL;**
- **function "not" (R: MVL) return MVL;**