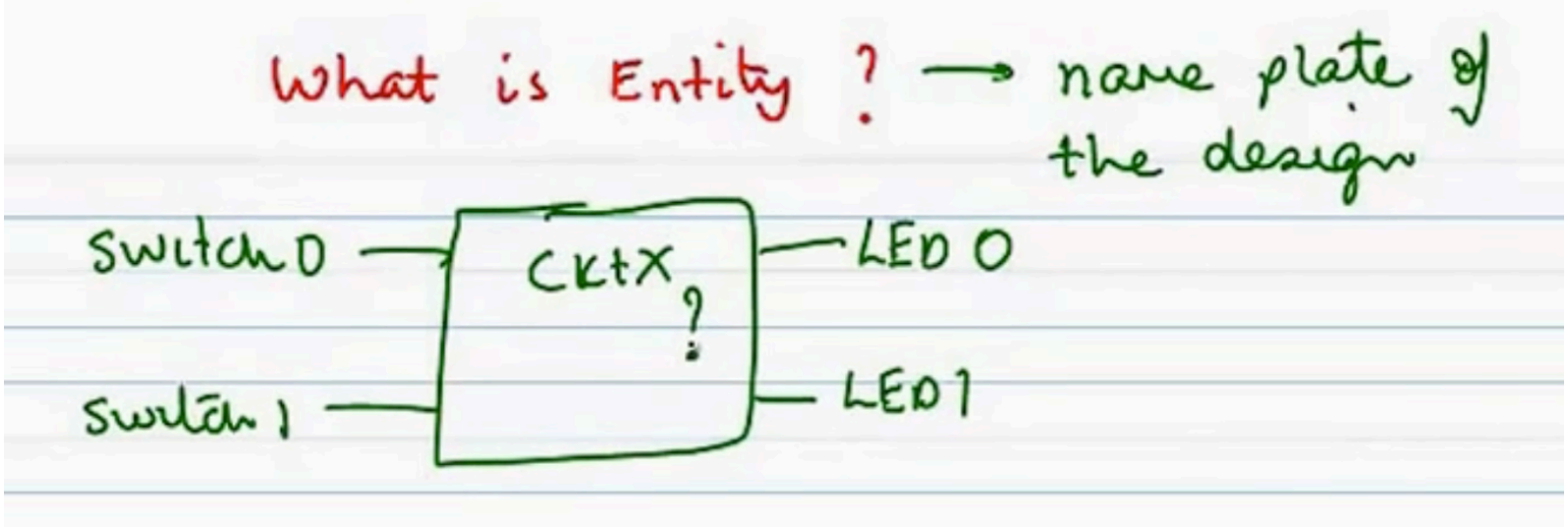
Lecture 4 (Chapter 2)

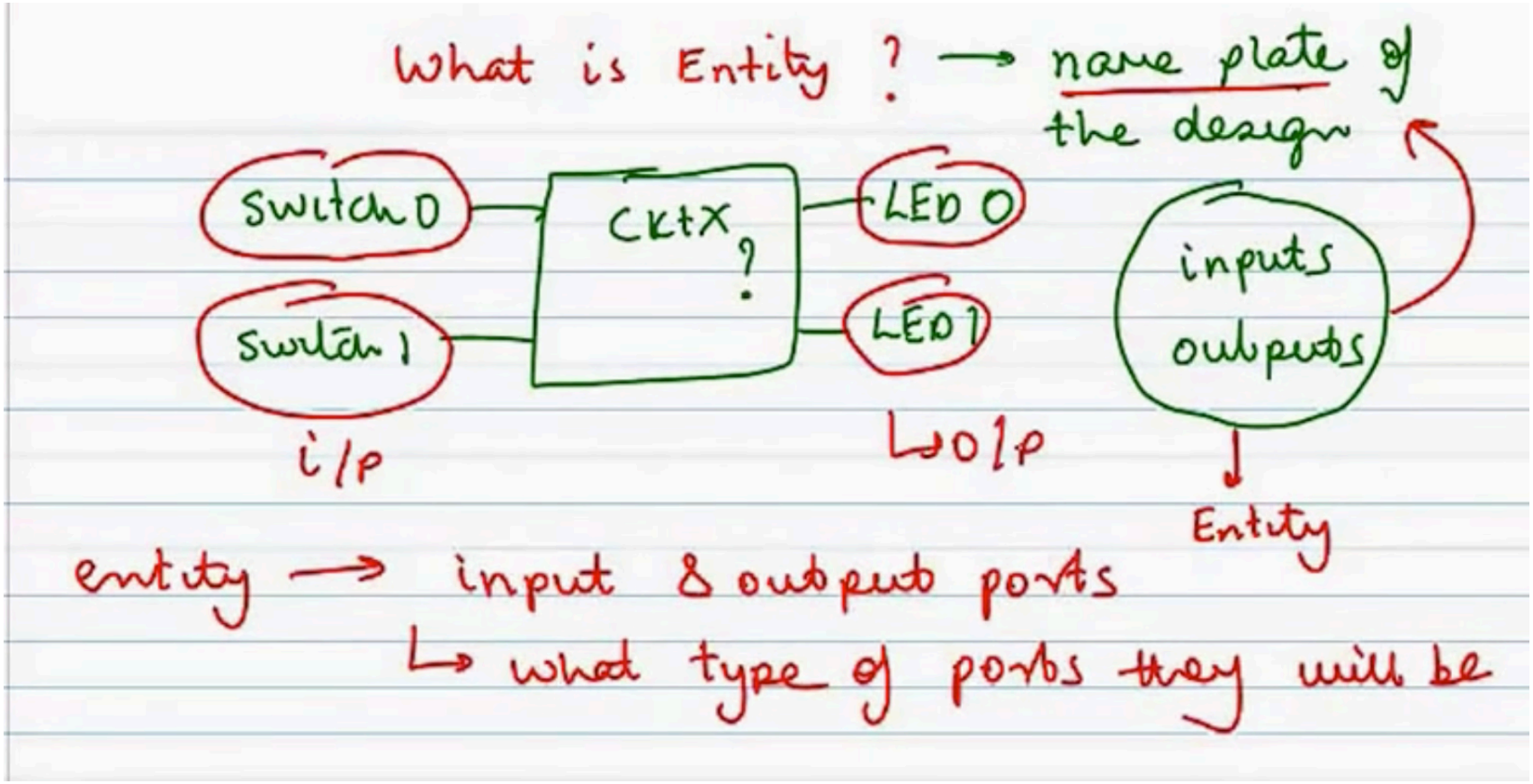
Introduction to VHDL (cont.)

Khaza Anuarul Hoque
ECE 4250/7250

Entity


What is Entity ? → name plate of the design
switch 0
switch 1
CktX
?
LED 0
LED 1

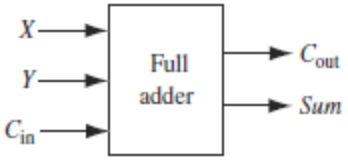
Entity


What is Entity ? → name plate of the design
switch 0
switch 1
CKtX ?
LED 0
LED 1
inputs
outputs
i/p
o/p
Entity
entity → input & output ports
↳ what type of ports they will be

Entity

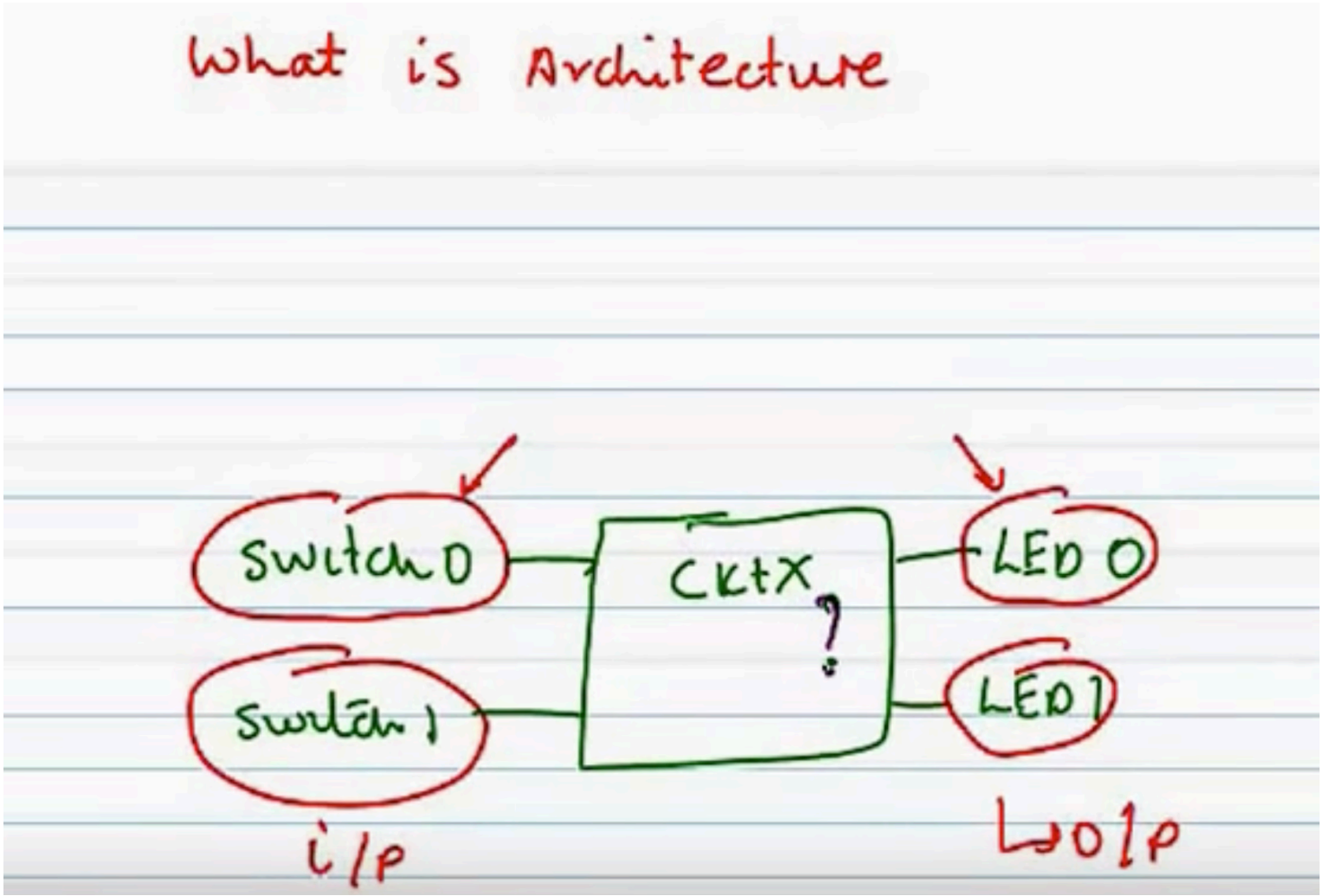
- Example for a Full Adder:

FIGURE 2-10: Entity Declaration for a Full Adder Module

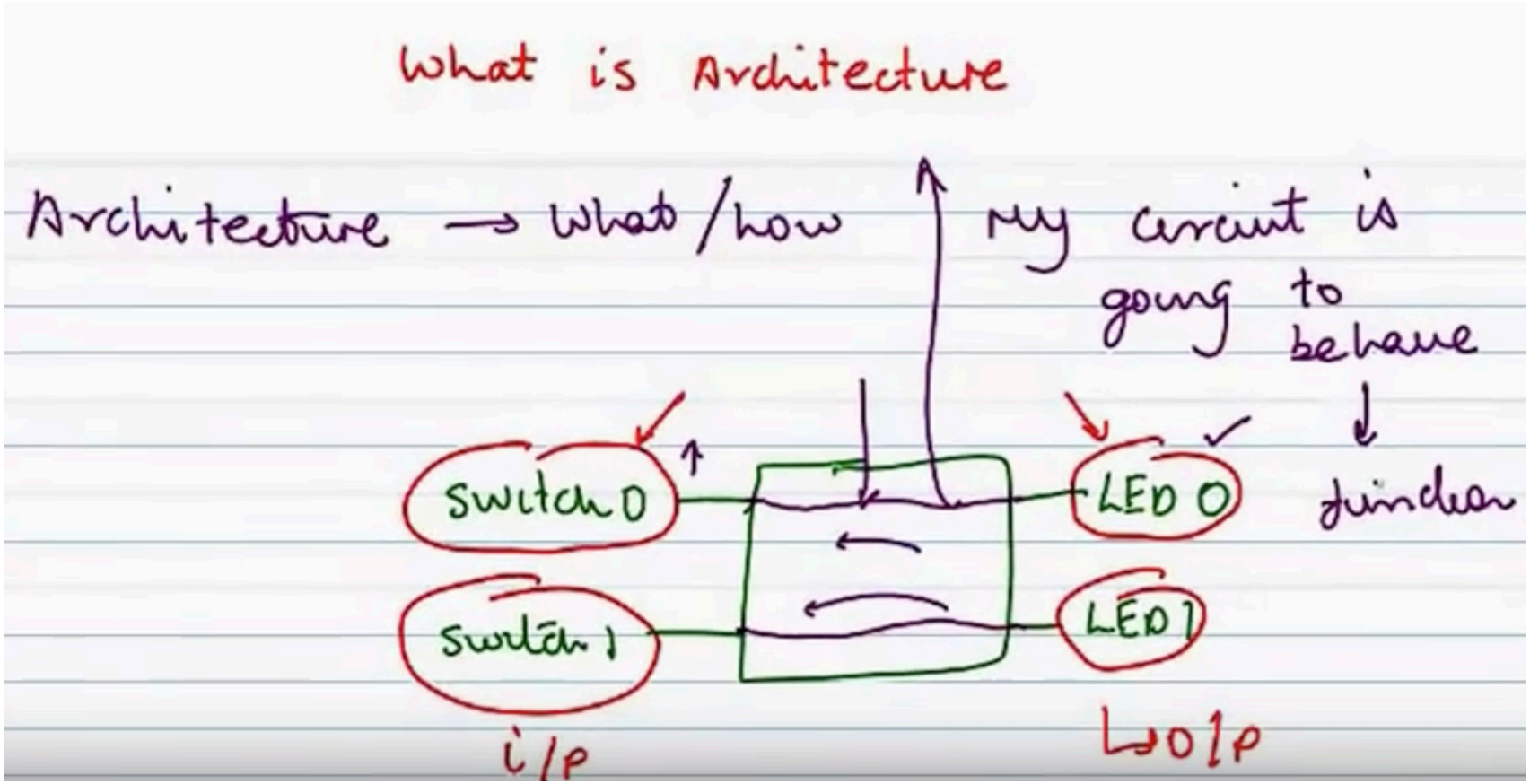



```
entity FullAdder is
  port(X, Y, Cin: in bit;     --Inputs
       Cout, Sum: out bit);   --Outputs
end FullAdder;
```

Architecture



Architecture



Architecture

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Switches_LEDs is
  Port ( switch_0 : in  STD_LOGIC;
         switch_1 : in  STD_LOGIC;
         LED_0 : out  STD_LOGIC;
         LED_1 : out  STD_LOGIC);
end Switches_LEDs;

architecture Behavioral of Switches_LEDs is
begin

end Behavioral;
```

```
architecture Behavioral of Switches_LEDs is
begin
  LED_0 <= switch_0;
  LED_1 <= switch_1;
end Behavioral;
```

Modes

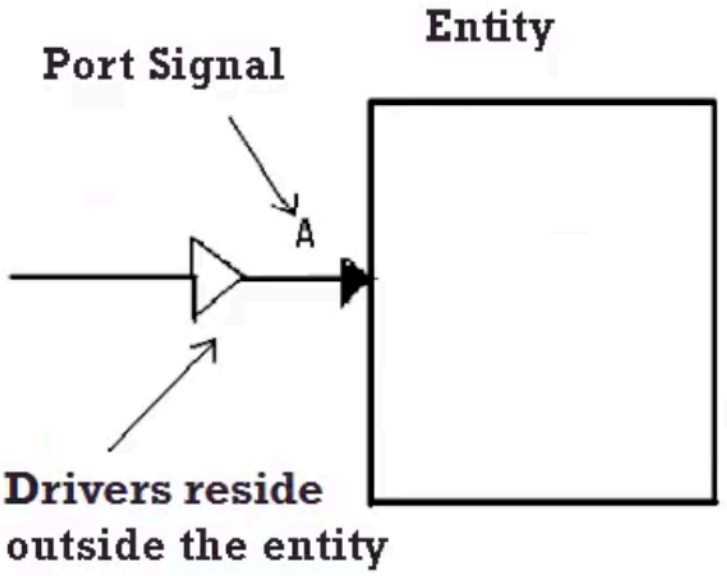
- Signal in the port has a ‘mode’ indicates the driver direction.
- Mode also indicates if the port can be read within the entity or not.
- Five modes in VHDL:
  - IN
  - OUT
  - INOUT
  - BUFFER
  - LINKAGE

Mode **IN**

- Value can be read but not assigned.

- Example:

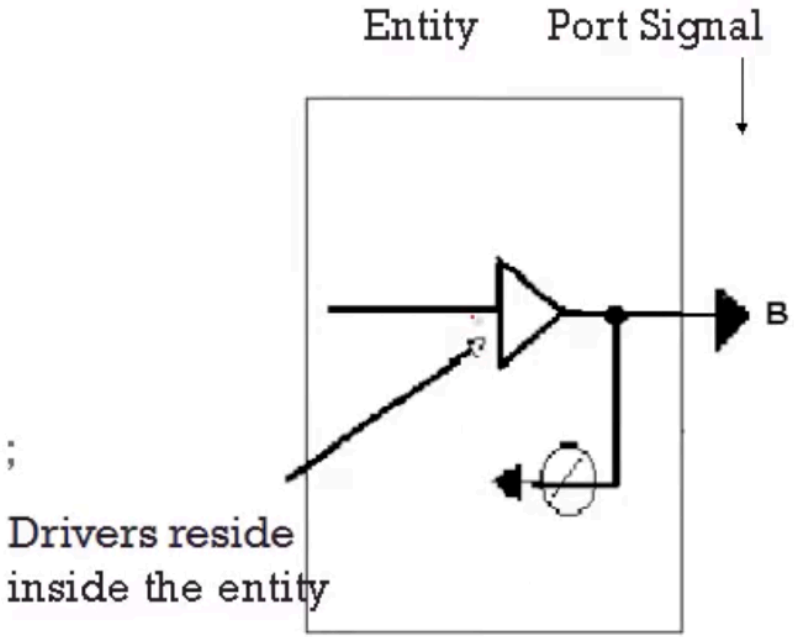
```
entity  driver is
port ( A : in  std_logic;
       B : out std_logic;
       Data : inout std_logic;
       Count : buffer std_logic ) ;
end driver ;
```



Mode **OUT**

- Value can be assigned but not read.Example:

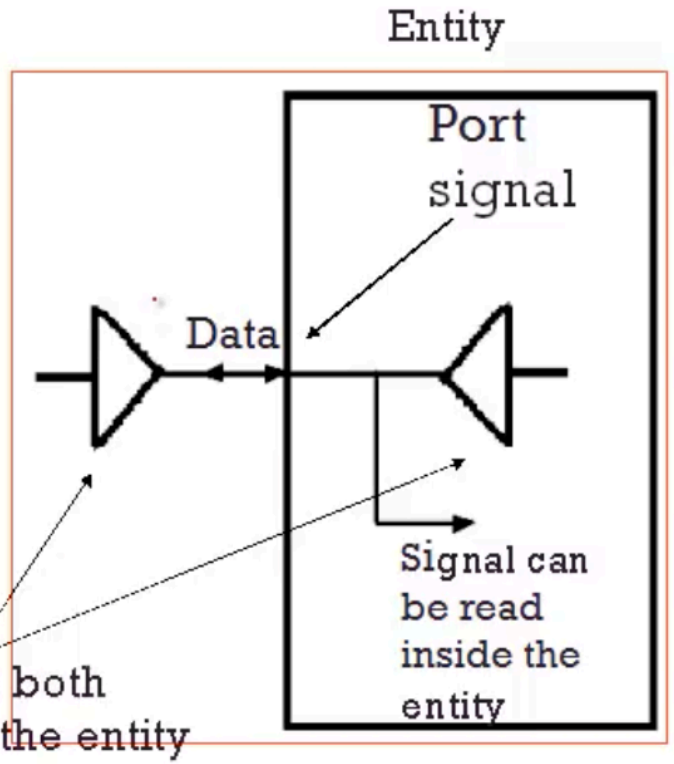
```
entity  driver is
port ( A : in  std_logic;
       B : out std_logic;
       Data : inout std_logic;
       Count : buffer std_logic ) ;
end driver ;
```



Mode **INOUT**

- Bi-directional
- Value can be read and assigned
- Example:

```
entity driver is
port ( A : in  std_logic;
       B : out std_logic;
       Data : inout std_logic;) ;
end driver ;
```



VHDL Data Types: Predefined

- Data types:
  - Predefined types:

Type	Description
bit	'0' or '1'
boolean	FALSE or TRUE
integer	an integer in the range $-(2^{31} - 1)$ to $+(2^{31} - 1)$
real	floating-point number in the range −1.0E38 to +1.0E38
character	any legal VHDL character including upper and lowercase letters, digits, and special characters
time	An integer with units fs, ps, ns, us, ms, sec, min, or hr

  - Can also have user-defined types

VHDL Data type: BIT

- Bit – Supports the values '0' & '1'.

```
entity driver is
port ( A : in  BIT;
       B : out BIT;
       Data : inout BIT);
end driver ;
```

VHDL Data type: Boolean

- **Boolean** – Supports literals FALSE & TRUE is defined as

  Ex : variable error_flag : boolean := true.

VHDL Data type: STD_LOGIC

- **Std_logic_type** – Data type defined in the std_logic_1164 package of IEEE library. It is defined as

type std_logic is ('U','X','0','1','Z','W','L','H','-');

'U' : Uninitialized
'X' : Unknown
'0' : Logic 0
'1' : Logic 1
'Z' : High impedance
'W' : Unknown
'L' : Low logic 0
'H' : Low logic 1
'-' : Don't care

U, X W, - represent behavior of model itself rather than the behavior of hardware being synthesized.

VHDL Operators

- VHDL Operators:
  1. Binary logical operators: **and or nand nor xor xnor**
  2. Relational operators: = /= < <= > >=
  3. Shift operators: **sll srl sla sra rol ror**
  4. Adding operators: + - & (concatenation)
  5. Unary sign operators: + -
  6. Multiplying operators: * / **mod rem**
  7. Miscellaneous operators: **not abs** **
- Class 7 has highest precedence, then 6, 5, 4, …
- Operators in the same class are applied left to right
- Parentheses change the order of precedence

VHDL Libraries

- Extend the functionality of VHDL.
- Need a library statement and a use statement.

```
library IEEE;
use IEEE.numeric_bit.all;
```

- The **numeric_bit** package uses bit_vectors to represent unsigned and signed binary numbers.