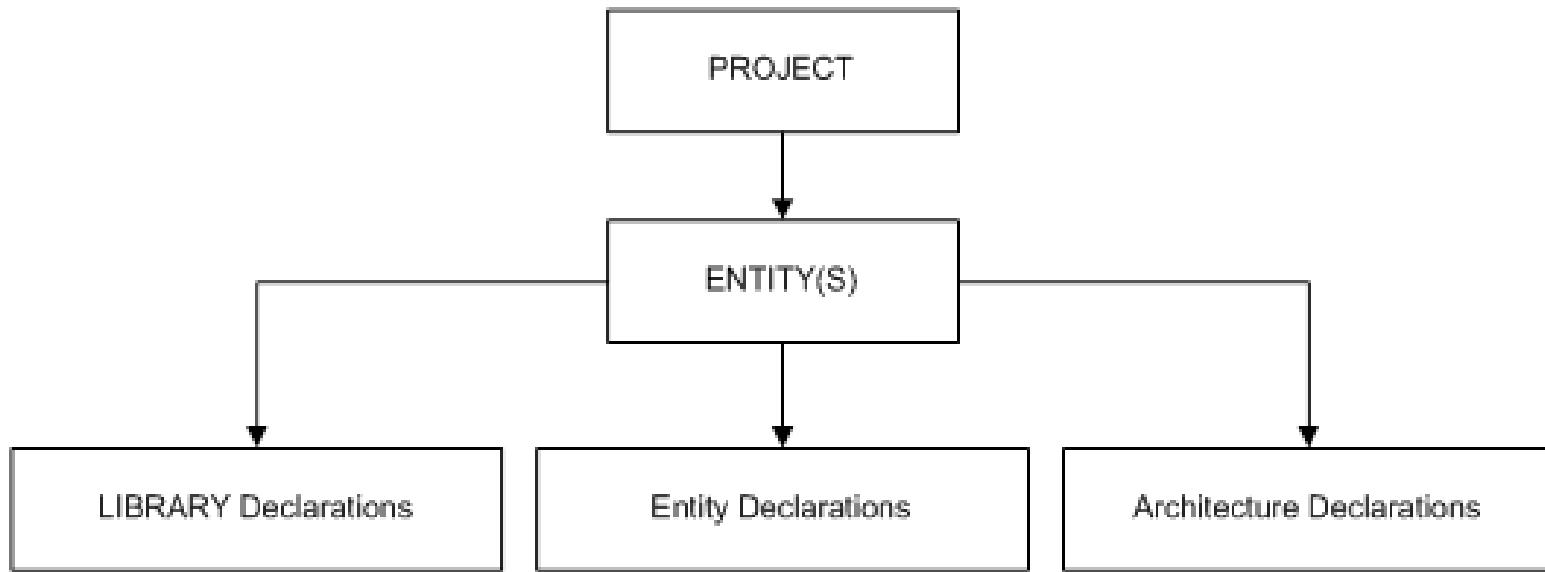


# VHDL 1 Slides

# Project Organization



```
1  -- A library clause declares a name as a library. It
2  |--- does not create the library; it simply forward declares it.
3  |   library <library_name>;
4  |
5  |   -- We will use IEEE library
6  |   library IEEE;
7  |
8  |   -- Import all the declarations in a package
9  |   use <library_name>.<package_name>.all;
10 |
11 |   -- Commonly imported packages:
12 |
13 |   -- BIT, STD_LOGIC and STD_LOGIC_VECTOR types, and relevant functions
14 |   use ieee.std_logic_1164.all;
15 |
16 |   -- SIGNED and UNSIGNED types, and relevant functions
17 |   use ieee.numeric_std.all;
18 |
19 |ENTITY <entity_name> IS
20 |  |PORT
21 |  |(
22 |  |  -- Input ports
23 |  |  <name> : IN <type>;
24 |  |  <name> : IN <type> := <default_value>;
25 |
26 |  |  -- Inout ports
27 |  |  <name> : inout <type>;
28 |
29 |  |  -- Output ports
30 |  |  <name> : OUT <type>;
31 |  |  <name> : OUT <type> := <default_value>;
32 |  );
33 |END <entity_name>;
34 |
35 |ARCHITECTURE <arch_name> OF <entity_name> IS
36 |
37 |  -- Declarations (optional)
38 |
39 |BEGIN
40 |  -- This section includes statements (code) that define the behavior of this entity.
41 |
42 |  -- Some commonly used statements:
43 |  -- Concurrent Signal Assignment (optional)
44 |  -- Process Statement (optional)
45 |  -- Conditional Signal Assignment (optional)
46 |  -- Selected Signal Assignment (optional)
47 |  -- Component Instantiation Statement (optional)
48 |  -- Generate Statement (optional)
49 |
50 |END <arch_name>;
```

# Library Declaration and Comments

```
LIBRARY ieee; -- Make ieee library available  
-- Make std_logic_1164 package available  
USE ieee.std_logic_1164.all;
```

- *ieee* – The library
- *std\_logic\_1164* – The Package
- *all* – What parts of the Package
- ; - The end of a statement
- -- The start of a comment

# The Entity Section

```
ENTITY entity_name IS
  PORT (
    input_1, input_2, ... , input_n: IN type;
    output_1, output_2, ... , output_m: OUT type;
  );
END entity_name ;
```

Table 3.2  
Synthesizable data types.

- Naming
- Data Types
- Vectors

Data types	Synthesizable values
BIT, BIT_VECTOR	'0', '1'
STD_LOGIC, STD_LOGIC_VECTOR	'X', '0', '1', 'Z' (resolved)
STD_ULOGIC, STD_ULOGIC_VECTOR	'X', '0', '1', 'Z' (unresolved)
BOOLEAN	True, False
NATURAL	From 0 to +2,147,483,647
INTEGER	From -2,147,483,647 to +2,147,483,647
SIGNED	From -2,147,483,647 to +2,147,483,647
UNSIGNED	From 0 to +2,147,483,647
User-defined integer type	Subset of INTEGER
User-defined enumerated type	Collection enumerated by user
SUBTYPE	Subset of any type (pre- or user-defined)
ARRAY	Single-type collection of any type above
RECORD	Multiple-type collection of any types above

# The Architecture Section

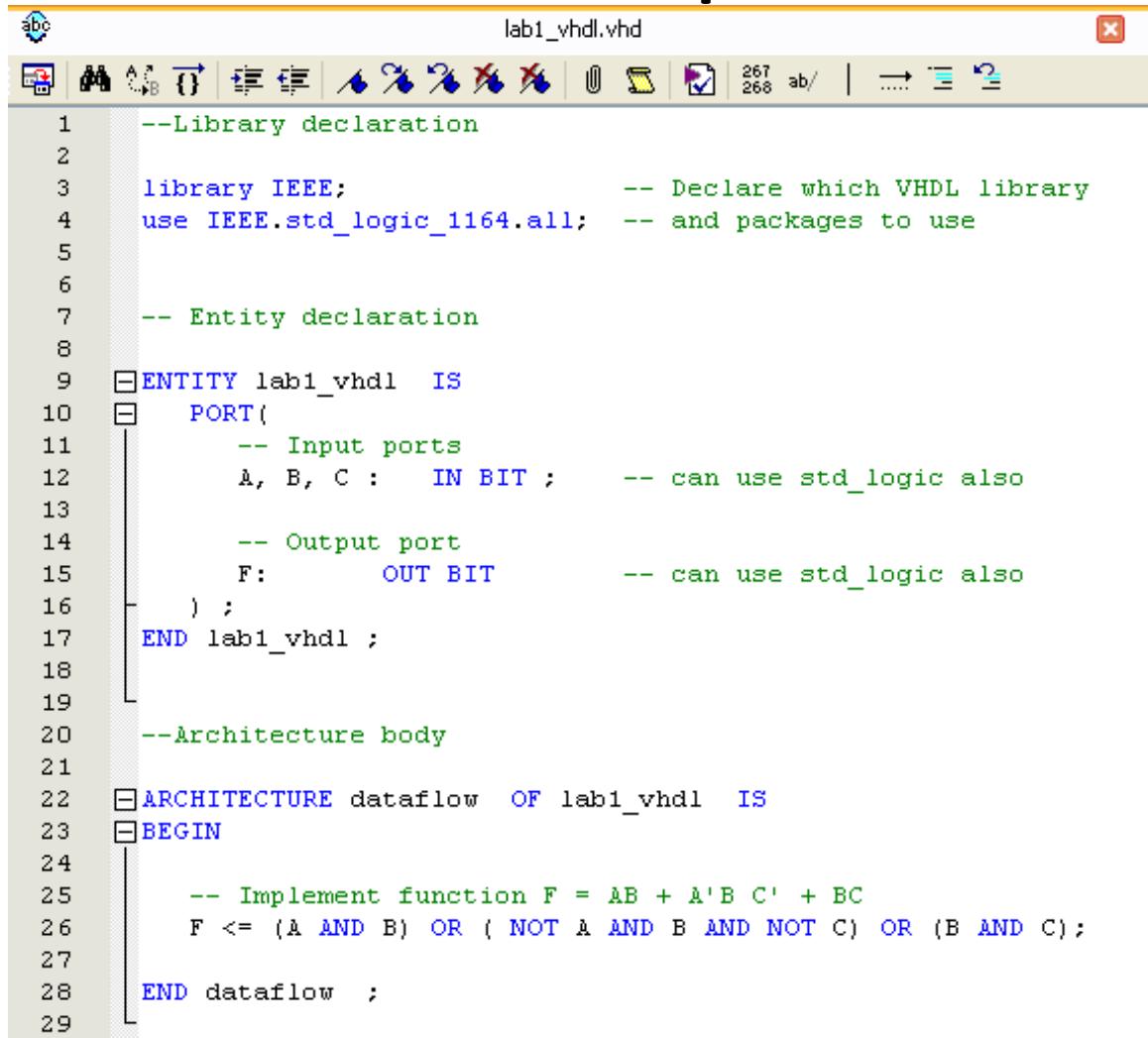
```
ARCHITECTURE arch_name OF entity_name IS
  SIGNAL internal_signal: type;
BEGIN
  --the statements that define the behavior
END arch_name ;
```

- Concurrent Execution
- Assignments
- Right Side

Table 4.1  
Operators.

Operator type	Operators	Data types
Assignment	$\leq=$ , $:=$ , $=>$	Any
Logical	NOT, AND, NAND, OR, NOR, XOR, XNOR	BIT, BIT_VECTOR, STD_LOGIC, STD_LOGIC_VECTOR, STD_ULOGIC, STD_ULOGIC_VECTOR
Arithmetic	$+$ , $-$ , $*$ , $/$ , $**$ (mod, rem, abs)*	INTEGER, SIGNED, UNSIGNED
Comparison	$=$ , $/=$ , $<$ , $>$ , $\leq=$ , $\geq=$	All above
Shift	sll, srl, sla, sra, rol, ror	BIT_VECTOR
Concatenation	$\&$ , $(\dots)$	Same as for logical operators, plus SIGNED and UNSIGNED

# Example



The screenshot shows a VHDL editor window titled "lab1\_vhdl.vhd". The code is a simple VHDL entity and architecture pair. The entity "lab1\_vhdl" has three input ports (A, B, C) and one output port (F). The architecture "dataflow" implements the function  $F = AB + A'B'C' + BC$ .

```
1 --Library declaration
2
3 library IEEE;           -- Declare which VHDL library
4 use IEEE.std_logic_1164.all; -- and packages to use
5
6
7 -- Entity declaration
8
9 ENTITY lab1_vhdl  IS
10    PORT(
11        -- Input ports
12        A, B, C :  IN BIT ;      -- can use std_logic also
13
14        -- Output port
15        F:        OUT BIT       -- can use std_logic also
16    );
17 END lab1_vhdl ;
18
19
20 --Architecture body
21
22 ARCHITECTURE dataflow  OF lab1_vhdl  IS
23 BEGIN
24
25    -- Implement function F = AB + A'B'C' + BC
26    F <= (A AND B) OR ( NOT A AND B AND NOT C) OR (B AND C);
27
28 END dataflow ;
```