Packages

- A package provides a convenient mechanism to store and share declarations that are common across many design units. A package is represented by

1. a package declaration

2. a package body.

Libraries or Design Libraries

- A compiled VHDL description is stored in a design library.
- A design library is an area of storage in the file system of the host environment.
- The format of this storage is not defined by the language. Typically, a design library is implemented on a host system as a file directory and the compiled descriptions are stored as files in this directory

Design Libraries cont..

- An arbitrary number of design libraries may be specified. Each design library has a logical name with which it is referenced inside a VHDL description.
- There is one design library with the logical name, STD, predefined in the language; this library contains the compiled descriptions for the two predefined packages, STANDARD and TEXTIO.
- Exactly one design library must be designated as the working library with the logical name, WORK.

Design Libraries cont..

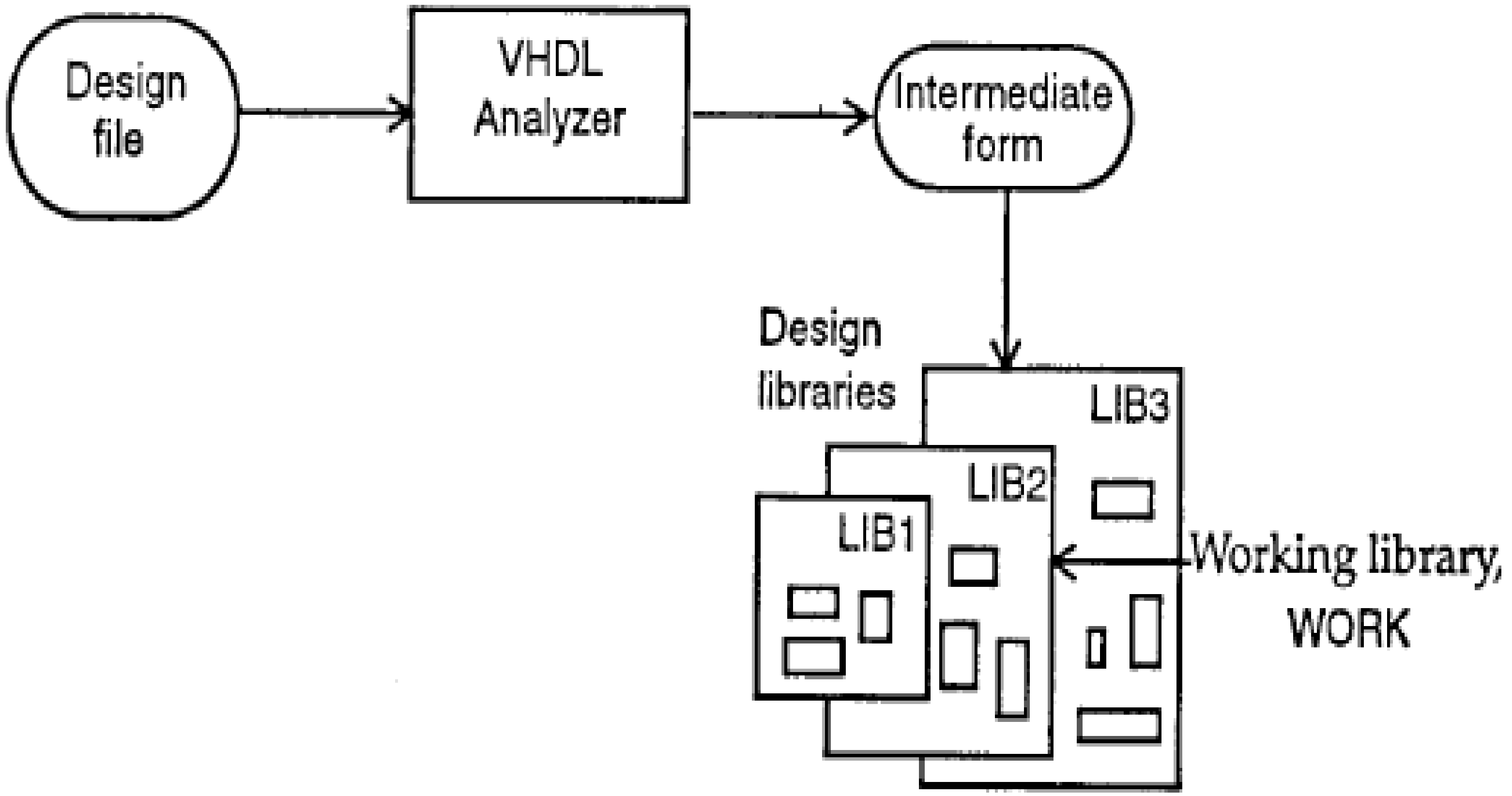


Figure 9.1 Atypical compilation process.

Design File

- The design file is an ASCII file containing the VHDL source. It can contain one or more design units, where a design unit is one of the following:

• entity declaration,
• architecture body,
• configuration declaration,
• package declaration,
• package body.

Design File

- Design units are further classified as

1. *Primary units: These units allow items to be exported out of the design unit. They are*

a. *entity declaration: The items declared in an entity declaration are implicitly visible within the associated architecture bodies.*

b. *package declaration: Items declared within a package declaration can be exported to other design units using context clauses.*

Design File cont..

2. *Secondary units: These units do not allow items declared within them to be exported out of the design unit, that is, these items cannot be referenced in other design units. These are*

a. *architecture body: A signal declared in an architecture body, for example, cannot be referenced in other design units.*

b. *package body.*

Generics

- It is often useful to pass certain types of information into a design description from its environment.
- Examples of such information are rise and fall delays, and size of interface ports. This is accomplished by using generics. Generics of an entity are declared along with its ports in the entity declaration.

An example of a generic N-input and gate-

```
entity AND_GATE is
generic (N: NATURAL);
port (A: in BIT_VECTOR(1 to N); Z: out BIT);
end AND_GATE;
architecture GENERIC_EX of AND_GATE is
begin
process (A)
    variable AND_OUT: BIT;
begin
    AND_OUT := '1';
    for K in 1 to N loop
        AND_OUT := AND_OUT and A(K);
    end loop;
    Z <= AND_OUT;
end process;
end GENERIC_EX;
```

Generics cont..

In this example, the size of the input port has been modeled as a generic.

A generic declares a constant object.

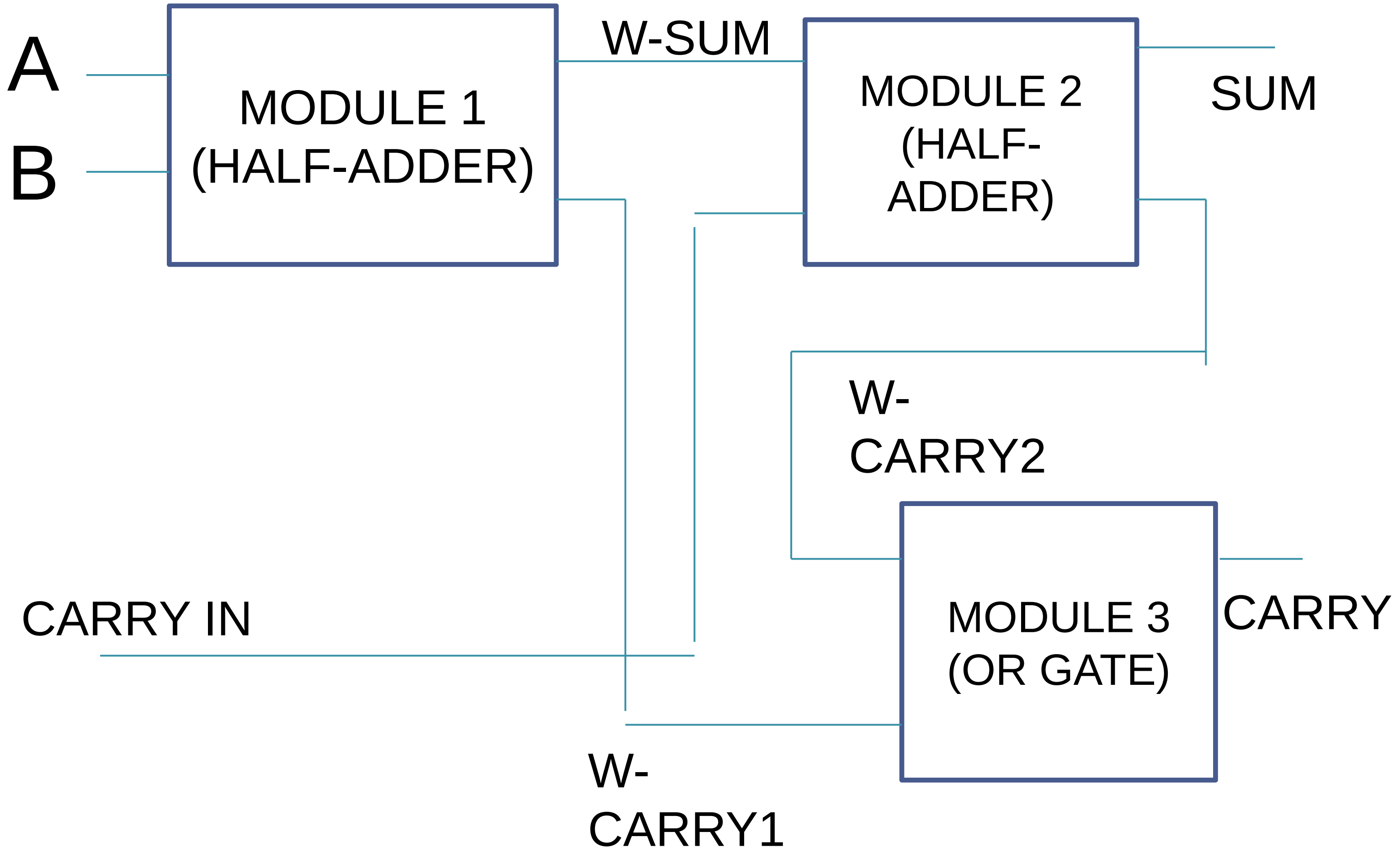
The value of this constant can be specified as a locally static expression in one of the following:

1. **entity declaration**
2. **component declaration**
3. **component instantiation**
4. **configuration specification**
5. **configuration declaration**

Generics cont..

- value for a generic must be explicitly specified.
- The value for a generic may be specified in the entity declaration for an entity as shown in this example. This is the default value for the generic.
- **entity NAND_GATE is**
  - **generic (M: INTEGER := 2);**
  - **port (A: in BIT_VECTOR(M downto 1); Z: out BIT);**
- **end NAND_GATE;**

Structural Layout



- Allow for hierarchical model layout which means that a module can be assembled out of several sub modules.
- The connection between these sub modules are defined within the architecture of a top module.
- A Full adder can be built with the help of two half adders (module 1, module 2) and OR gate (module 3)