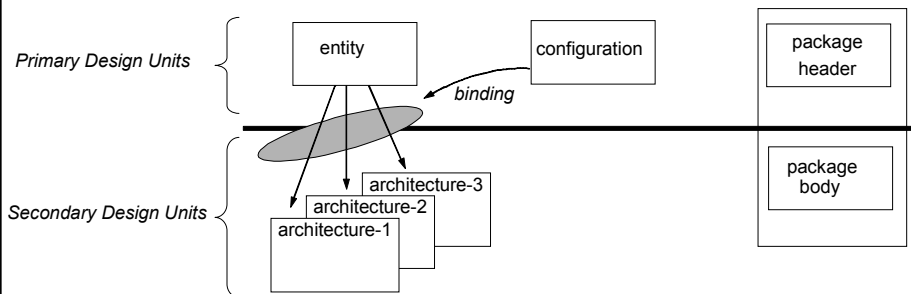


Programming Mechanics

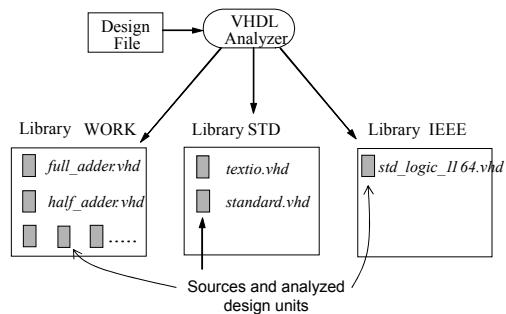
Design Units

- Basic unit of VHDL programming is a design unit which is one of the following



- Multiple instances of a name can exist
 - Local names are the default
 - Provide the full path name for a name

- Visibility of names
 - Visibility of a name follows the hierarchy
 - Declared in a package → all design units that use the package
 - Entity → in architectures used for this entity
 - Architecture → processes used in the architecture
 - Process → in the process
 - Subprogram → subprogram



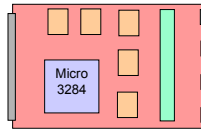
- Design unit names are used to construct intermediate file names
- The libraries WORK and STD

- Use packages to separate synthesizable from simulatable
- Change secondary design units without recompilation of the design hierarchy
- Use several configurations
 - Record/compare different architectures

- Analysis (Compilation) and Analysis Order
 - Primary vs. secondary design units
 - Organization of design units and files

```
architecture structural of full_adder is
component half_adder is
port (a, b : in std_logic;
sum, carry : out std_logic);
end component half_adder;
component or_2 is
port (a, b : in std_logic;
c : out std_logic);
end component or_2;
signal s1, s2, s3 : std_logic;
begin
H1: half_adder port map (a => In1, b => In2, sum => s1, carry=> s3);
H2: half_adder port map (a => s1, b => c_in, sum => sum, carry => s2);
O1: or_2 port map (a => s2, b => s3, c => c_out);
end architecture structural;
```

Compilation Dependencies



entity board is
port (...)

dependency

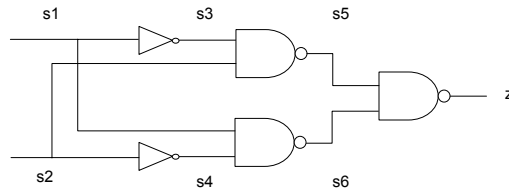
entity micro3284 is
port (..)

- Compilation dependencies follow hardware dependencies
 - Changes in the interface
 - Architecture changes can be insulated

- Note that recompilation is interpreted as a change
 - Locating architectures and entities in the same file
 - Creates dependencies that may not in fact exist

Basic Steps: Simulation

- Elaboration
 - Of the hierarchy → produces a netlist of processes

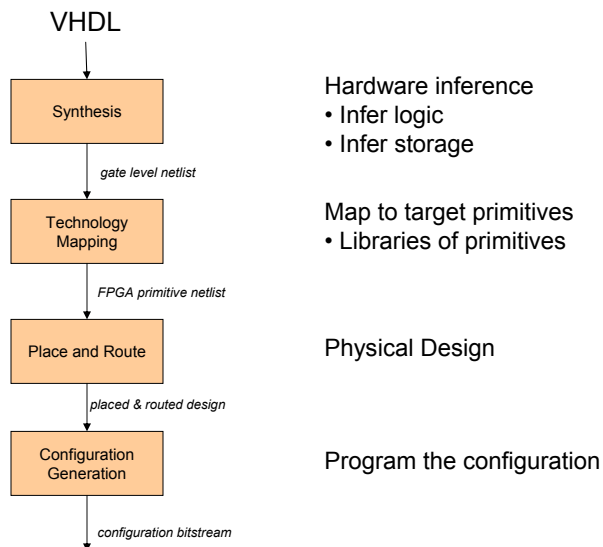


- Of declarations
 - Generics and type checking
- Storage allocation
- Initialization

Basic Steps: Simulation

- Initialization
 - All processes are executed until suspended by wait statements or sensitivity lists
 - All nets initialized to default or user specified values
 - Initialize simulation time
- Simulation
 - Discrete event simulation
 - Two step model of time
 - Set net values
 - Execute all affected processes and schedule new values for nets
 - Simulator step time

Basic Steps: Synthesis



Technology Mapping

