

★ Subprograms (Chapter 8, Pg. 183)

* Subprogram: A sequential algorithm that performs a certain computation.

- i) FUNCTIONS: • used for computing a single value
• executes in zero ~~simulated~~ time
- ii) PROCEDURES: • can return zero or more values
• may contain WAIT statements

* Syntax

subprogram specification is

subprogram item declarations

begin

subprogram statements

end [function/procedure] [subprogram name];

→ name is interface of subprog

→ temp variable only
available during this function

→ sequential statements

* Example:

function Largest (totalno: integer; set: pattern)

return real is

-- pattern is defined elsewhere 2 b a 1D array type

variable return_value: real := 0.0;

begin

for k in set'range

loop

if set(k) > return_value then

return_value := set(k);

end if;

end loop;

→ only constants/signals
allowed in

functions
• only 1D mode sig.
allowed

return return_value;
end Largest;

★ Type of FUNCTIONS

- Pure: Time invariant o/p
- Impure: Time varying o/p
Ex: NOW

→ Default

★ How to call a Function?

* Eg:

sum := sum + Largest(^{use?} Max-Coin, collection)

★ Write a function

- TO-BIT

- Parity-Checker

- Byte-Reversal

★ Procedures (Pg. 189)

* can return zero value or more values

* parameter list can contain const., signals, variables

* modes can be in, out or inout

* Example:

type OP_CODE is (ADD, SUB, MUL, DIV, LT, LE, EQ);

procedure ARITH_UNIT (A, B: in integer; OP: in OP_CODE;
Z: out integer; ZComp: out boolean) is

begin

case OP is

when ADD => Z := A + B;

SUB
MUL
DIV

when LT => ZComp := A < B;

LE => ZComp := A <= B;

EQ => ZComp := A = B;

end case;

end ARITH_UNIT;

* Packages (Chapter 9, Pg. 203)

* Students' list who went to IIIT Sports Meet?

* Package Declaration

It contains set of declarations that may possibly be shared by many design units.

* Syntax:

package package_name is

package-item-declaration --> These may be:

-- type declaration

-- subtype

-- constant

-- signal

-- variable

-- component

end package package_name;

* Eg:

package SYNTH_PACK is

constant low2high_time := 20ns;

type ALUop is (ADD, SUB, MUL, DIV, EQL);

component NAND2

port (A, B: in bit; C: out bit);

end component;

end package SYNTH_PACK;

* Items declared in a package declaration can be accessed using "library" & "use" clause.

* Package declaration may also include function / procedure declaration.

★ Package Body

- * It contains the behaviours of the subprograms & the values of the deferred constants declared in a package declaration. §

* Syntax:

package body package_name is
package-body-declarations --> These are:
-- subprogram bodies
-- complete constant declarations
-- type & subtypes
end package body package_name;

* Eg:

```
package body Prog_Pack is
  use work_tables.all;
  constant Prop_Delay: Time := 15ns;
  procedure LOAD (signal array_name: inout MVL_VECTOR;
                 start_bit, stop_bit, int_value: in integer) is
  begin
    -- procedure behaviour here
  end LOAD;
end package body Prog_Pack;
```

* Note:

1. An item declared inside package body has its scope restricted to be within the package body. & this item is not visible in other design units.
2. Items declared in package "declaration" can be accessed by other design units.
3. ∴ Package "Body" is used to store private declarations while a package "declaration" is used to store public declarations.

Q] Why?

- * Analogy → entity & architecture.