

# *Configuration Declaration*

- It is used to select one of the possibly many architecture bodies.
- Used to bind components.
- Used to represent structure in that architecture body.

# Example of Configuration Declaration

- Consider the following configuration declaration for the HALF\_ADDER entity.

```
library CMOS_LIB, MY_LIB;  
configuration HA_BINDING of  
  HALF_ADDER is  
for HA-STRUCTURE  
  for X1:XOR2
```

# Example of Configuration Declaration cont..

```
use entity CMOS_LIB.XOR_GATE(DATAFLOW);  
end for;  
for A1:AND2  
    use configuration MY_LIB.AND_CONFIG;  
    end for;  
end for;  
end HA_BINDING;
```

# *Package Declaration*

- A package declaration is used to store a set of common declarations like components, types, procedures, and functions.
- These declarations can then be imported into other design units using a use clause.

# *Example of Package*

## *Declaration*

```
package EXAMPLE_PACK is  
  type SUMMER is (MAY, JUN, JUL, AUG,  
    SEP);
```

```
  component D_FLIP_FLOP  
    port (D, CK: in BIT; Q, QBAR: out BIT);  
  end component;
```

```
  constant PIN2PIN_DELAY: TIME := 125 ns;
```

```
  function INT2BIT_VEC (INT_VALUE: INTEGER)  
    return BIT_VECTOR;  
end EXAMPLE_PACK;
```

## *Package Declaration cont..*

- Assume that this package has been compiled into a design library called DESIGN\_LIB. Consider the following clauses associated with an entity declaration.

**library DESIGN\_LIB;**

**use DESIGN\_LIB.EXAMPLE\_PACK.all;**

**entity RX is . . .**

## *Package Declaration cont..*

- It is also possible to selectively import declarations from a package declaration into other design units. For example:

```
library DESIGN_LIB;
```

```
use DES[GN_LIB.EXAMPLE_PACK.D_FLIP_FLOP;
```

```
use DESIGN_LIB.EXAMPLE_PACK.PIN2PIN_DELAY;
```

```
architecture RX_STRUCTURE of RX is . . .
```

# *Package Body*

- A package body is primarily used to store the definitions of functions and procedures that were declared in the corresponding package declaration.
- Used to do constant declarations for any deferred constants that appear in the package declaration.

## *Package Body Cont..*

- A package body is always associated with a package declaration.
- A package declaration can have at most one package body associated with it.
- Contrast this with an architecture body and an entity declaration where multiple architecture bodies may be associated with a single entity declaration.

# *Example of Package Body*

- Here is the package body for the package EXAMPLE\_PACK declared.

```
package body EXAMPLE_PACK is  
function INT2BIT_VEC (INT_VALUE:  
INTEGER)  
    return BIT_VECTOR is  
begin  
    --Behavior of function described here.  
end INT2BIT_VEC;  
end EXAMPLE_PACK;
```