© Zainalabedin Navabi, 1994

PART 5

Design_Organization

- Subprograms
  - Describing Logic
  - Aid in testing
  - Utilities
- Packaging
  - Components
  - Utilities
- Parametrization
  - Generic design
  - Generic specification
    - Using defaults
    - Using fixed values
    - Passing values
- Top level configuration
  - Comparator example
  - Register example
- Design libraries
- A complete example

© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994

Design_Organization.subprograms.describing_logic

a_gt_b = a. gt + b'. gt+ a.b'
a_eq_b = a.b.eq + a'.b'.eq
a_lt_b = b.lt + a'.lt + b.a'

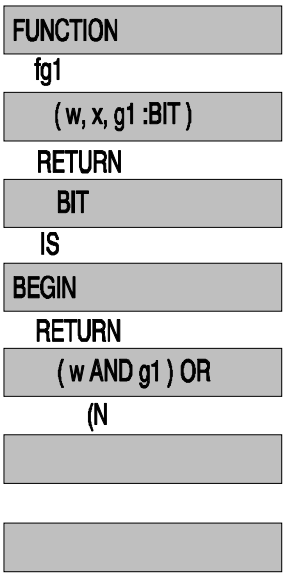
```
ARCHITECTURE functional OF bit_comparator IS
  FUNCTION fgl (w, x, gl : BIT) RETURN BIT IS
  BEGIN
     RETURN  (w AND gl) OR (NOT x  AND  gl) OR (w  AND  NOT x);
  END fgl;
  FUNCTION feq (w, x, eq : BIT) RETURN BIT IS
  BEGIN
     RETURN  (w AND x AND eq) OR (NOT w  AND  NOT x  AND  eq);
  END feq;
BEGIN
  a_gt_b <= fgl (a, b, gt) AFTER 12 NS;
  a_eq_b <= feq (a, b, eq) AFTER 12 NS;
  a_lt_b <= fgl (b, a, lt) AFTER 12 NS;
END functional;
```

- Will begin with simple functions
- Functions are used to simplify descriptions
- Function definition can only be used by this architecture

© Zainalabedin Navabi, 1994

© Zainalabedin Navabi, 1994

Design_Organization.syntax



© Zainalabedin Navabi, 1994