GONZAGA UNIVERSITY

VHDL: Generics

GONZAGA UNIVERSITY

GENERIC

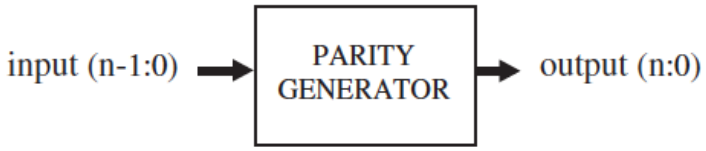
- GENERIC is a way of specifying a generic parameter
- The purpose is to confer the code more flexibility and reusability
- A GENERIC statement, when employed, must be declared in the ENTITY

```
GENERIC (parameter_name : parameter_type := parameter_value);
```

GONZAGA UNIVERSITY

Example:

```
1  ------ File parity_gen.vhd (component): -----
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.all;
4  ------------------------------------
5  ENTITY parity_gen IS
6     GENERIC (n : INTEGER := 7);  -- default is 7
7     PORT ( input: IN  std_logic_vector (n DOWNTO 0);
8            output: OUT std_logic_vector (n+1 DOWNTO 0));
9  END parity_gen;
10 ------------------------------------
11 ARCHITECTURE parity OF parity_gen IS
12 BEGIN
13    PROCESS (input)
14       VARIABLE temp1:  std_logic;
15       VARIABLE temp2:  std_logic_vector (output'RANGE);
16    BEGIN
17       temp1 := '0';
18       FOR i IN input'RANGE LOOP
19          temp1 := temp1 XOR input(i);
20          temp2(i) := input(i);
21       END LOOP;
22       temp2(output'HIGH) := temp1;
23       output <= temp2;
24    END PROCESS;
25 END parity;
26 ------------------------------------------------
```



GONZAGA UNIVERSITY

Example:

```
1  ------ File my_code.vhd (actual project): ------------
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.all;
4  ------------------------------------
5  ENTITY my_code IS
6     GENERIC (n : POSITIVE := 2); -- 2 will overwrite 7
7     PORT ( inp: IN  std_logic_vector (n DOWNTO 0);
8            outp: OUT std_logic_vector (n+1 DOWNTO 0));
9  END my_code;
10 ------------------------------------
11 ARCHITECTURE my_arch OF my_code IS
12 ------------------------
13    COMPONENT parity_gen IS
14       GENERIC (n : POSITIVE);
15       PORT (input: IN  std_logic_vector (n DOWNTO 0);
16             output: OUT  std_logic_vector (n+1 DOWNTO 0));
17    END COMPONENT;
18 ------------------------
19 BEGIN
20    C1: parity_gen GENERIC MAP(n=>3) PORT MAP(inp,outp);
21 END my_arch;
22 ------------------------------------------------------
```

```
-- 3 will
-- overwrite
-- 2
```