
SEL0384/SEL0606 – Laboratório de Sistemas Digitais

Aula 6 – Genéricos e Esquemas de Geração

Prof. Dr. Maximilian Luppe

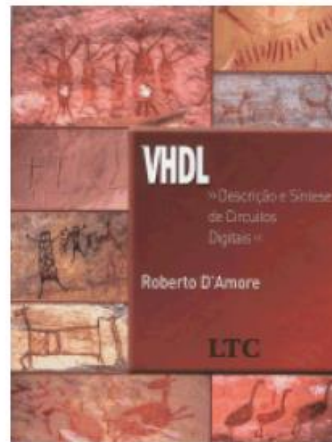
Livro adotado:

VHDL - Descrição e Síntese de Circuitos Digitais

Roberto d'Amore

ISBN 85-216-1452-7

Editora LTC www.ltceditora.com.br



Para informações adicionais consulte: www.ele.ita.br/~damore/vhdl

Componentes e esquemas de iteração

Tópicos

- **Definição de genéricos**
- **Comando GENERATE**
 - esquema de geração FOR
- **Comando LOOP**
 - esquema de interação FOR

Definição de genéricos

- **Genéricos:** fornecem um meio de levar informações externas estáticas para entidades de projeto e blocos
- **Informações:**
 - características de desempenho
 - parâmetros que definem a lei de formação na interligação de componentes.
- **Exemplos:**
 - declaração com definição do valor
 - declaração sem definição do valor

```
ENTITY nome_entidade IS
  GENERIC (generico_1 :tipo_do_generico_1 := valor_inicial; -- declaracao com valor
           generico_2 :tipo_do_generico_2);               -- apenas declaracao
  PORT    (.....);
END nome_entidade;
```

Definição de genéricos - exemplo

- **Registrador**: número de bits definido pelo genérico **n**
- **Após a declaração**, o genérico pode ser empregado
 - **vide linhas 5 e 6**

```
1 ENTITY flipn_3 IS
2   GENERIC (n      : INTEGER := 3); -- declaracao e definicao do valor do generico
3   PORT      (ck    : IN  BIT;      -- relógio
4              rst   : IN  BIT;      -- rst=1 leva q=000 assíncrono
5              d     : IN  BIT_VECTOR(n-1 DOWNTO 0); -- definido pelo generico
6              q     : OUT BIT_VECTOR(n-1 DOWNTO 0)); -- definido pelo generico
7 END flipn_3;
8
9 ARCHITECTURE teste OF flipn_3 IS
10 BEGIN
11   PROCESS (ck, rst)
12   BEGIN
13     IF      (rst = '1')          THEN q <= (OTHERS => '0'); -- q=00...0
14     ELIF (ck'EVENT AND ck = '1') THEN q <= d;              -- armazena dado
15     END IF;
16   END PROCESS;
17 END teste;
```

Comando **GENERATE**

- Cópia de comandos concorrentes:

- esquema de geração **FOR**
- esquema de geração **IF**

- **Aplicação: geração automática**

- circuitos regulares
- circuitos que seguem uma lei de formação
- exemplo:
 - unidades lógicas aritméticas
 - somadores
 - multiplicadores

Comando **GENERATE** - esquema de geração **FOR**

- **FOR** identificador **IN** limites_da_geracao **GENERATE**

- Identificador: não necessita ser declarado

- Limites da geração:

- faixa discreta de valores: (0 TO 7) (23 DOWNT0 0)

- atributos que retornem faixa de valores: a'RANGE b'REVERSE_RANGE

```
abc: FOR identificador IN valor_inicial TO valor_final GENERATE
    -- comando concorrente
END GENERATE abc;
```

- Exemplos:

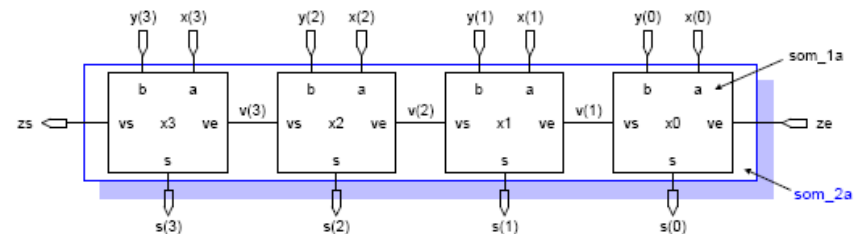
```
def: FOR i IN sinal'RANGE GENERATE
    -- comando concorrente
END GENERATE def;

ghi: FOR x IN valor_final DOWNT0 valor_inicial GENERATE
    -- comando concorrente
END GENERATE;                                -- ultimo rotulo opcional

xyz: FOR k IN sinal'REVERSE_RANGE GENERATE
    -- comando concorrente
END GENERATE;                                -- ultimo rotulo opcional
```

Componentes

- Descrição do somador empregando o componente: somador de 1 bit



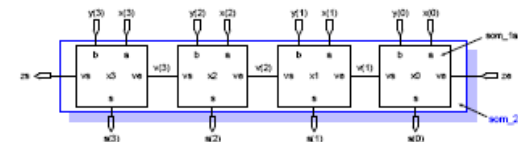
```

1 ENTITY som_2a IS
2   PORT ( x, y : IN BIT_VECTOR (3 DOWNTO 0); -- entradas do somador
3         ze : IN BIT; -- entrada vem um
4         s : OUT BIT_VECTOR (3 DOWNTO 0); -- soma
5         zs : OUT BIT); -- vai um
6 END som_2a;
7
8 ARCHITECTURE estrutural OF som_2a IS
9
10  COMPONENT som_1a
11    PORT (a, b, ve : IN BIT; s, vs : OUT BIT);
12  END COMPONENT;
13
14  SIGNAL v : BIT_VECTOR (3 DOWNTO 1); -- vai um interno
15
16 BEGIN
17   x0: som_1a PORT MAP( x(0), y(0), ze, s(0), v(1));
18   x1: som_1a PORT MAP( x(1), y(1), v(1), s(1), v(2));
19   x2: som_1a PORT MAP(b =>y(2), a =>x(2), s =>s(2), ve =>v(2), vs =>v(3));
20   x3: som_1a PORT MAP( x(3), y(3), v(3), s(3), zs);
21 END estrutural;

```


Comando **GENERATE** - esquema de geração **FOR**

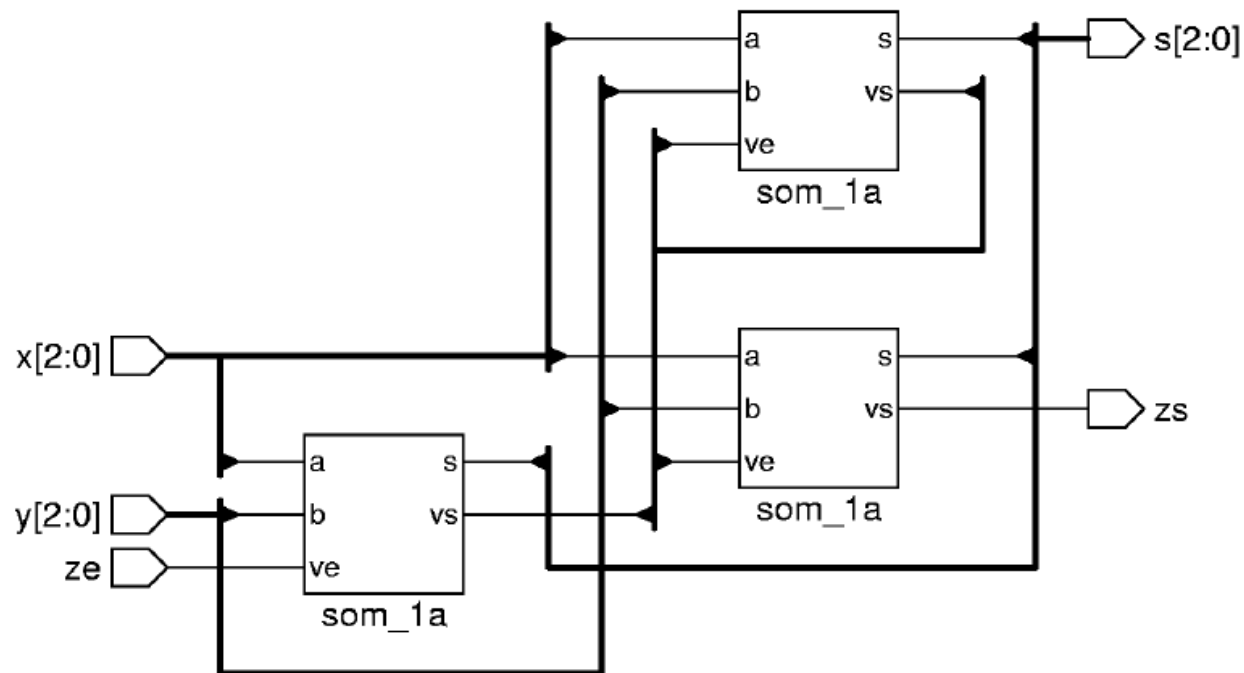
• Exemplo de um somador empregando componente



```
1 ENTITY som_6aa IS
2   GENERIC (n      : INTEGER := 3 );           -- numero de bits
3   PORT    (x, y   : IN  BIT_VECTOR (n-1 DOWNTO 0); -- entradas do somador
4           ze      : IN  BIT;                  -- vem um
5           s       : OUT BIT_VECTOR (n-1 DOWNTO 0); -- saida
6           zs      : OUT BIT);                 -- vai um
7 END som_6aa;
8
9 ARCHITECTURE estrutural OF som_6aa IS
10  COMPONENT som_1a
11    PORT    (a, b, ve : IN  BIT;  s, vs : OUT BIT);
12  END COMPONENT;
13  SIGNAL v : BIT_VECTOR (n DOWNTO 0); -- vai um interno
14 BEGIN
15  v(0) <= ze;
16  zs  <= v(n);
17  abc: FOR i IN 0 TO n-1 GENERATE
18    centro: som_1a PORT MAP (x(i),  y(i),  v(i),  s(i),  v(i+1));
19  END GENERATE abc;
20 END estrutural;
```

Comando **GENERATE** - esquema de geração **FOR**

- **Exemplo de um somador empregando componente**
- circuito sintetizado:



Comando **GENERATE** -- esquema de geração **FOR**

- Exemplo de um somador empregando comandos concorrentes

```
1 ENTITY som_6c IS
2   GENERIC(n      : INTEGER := 3 );           -- numero de bits
3   PORT    (x, y   : IN  BIT_VECTOR (n-1 DOWNT0 0); -- entradas do somador
4           ze      : IN  BIT;                -- vem um
5           s       : OUT BIT_VECTOR (n-1 DOWNT0 0); -- saida
6           zs      : OUT BIT);                -- vai um
7 END som_6c;
8
9 ARCHITECTURE teste OF som_6c IS
10  SIGNAL v : BIT_VECTOR (n DOWNT0 0); -- vai um interno
11 BEGIN
12  v(0) <= ze;
13  zs <= v(n);
14  abc: FOR i IN 0 TO n-1 GENERATE
15    s(i) <= x(i) XOR y(i) XOR v(i);
16    v(i+1) <= (x(i) AND y(i)) OR (x(i) AND v(i)) OR (y(i) AND v(i));
17  END GENERATE abc;
18 END teste;
```

Comando LOOP

- **Repetição da execução de comandos seqüenciais:**
 - esquemas de iteração: FOR e WHILE
- **Aplicação:** geração automática
 - circuitos regulares
 - rotinas de conversão (não necessariamente aplicadas em síntese)

Comando **LOOP** modo iterativo **FOR**

- **FOR** **identificador** **IN** **limites_da_iteracao** **LOOP**

- **Identificador**: não necessita ser declarado
- **Limites da iteração**:
 - faixa discreta de valores: (0 TO 7) (23 DOWNT0 0)
 - atributos que retornem faixa de valores: a'RANGE b'REVERSE_RANGE

- **Exemplos:**

```
abc: FOR i IN valor_inicial TO valor_final LOOP      -- faixa crescente
    -- comando sequencial
END LOOP abc;

def: FOR i IN valor_final DOWNT0 valor_inicial LOOP -- faixa decrescente
    -- comando sequencial
END LOOP def;

ghi: FOR i IN x'RANGE LOOP      -- faixa: elementos na faixa definida, exemplo
    -- comando sequencial      --      (final DOWNT0 inicial), (inicial TO final)
END LOOP ghi;
```

Comando **LOOP** - modo iterativo **FOR**

• Exemplo de um somador - comandos sequenciais

```
1 ENTITY som_8a IS
2   GENERIC(n      : INTEGER := 3 );           -- numero de bits
3   PORT    (x, y   : IN  BIT_VECTOR (n-1 DOWNT0 0); -- entradas do somador
4           ze      : IN  BIT;                -- vem um
5           s       : OUT BIT_VECTOR (n-1 DOWNT0 0); -- saida
6           zs      : OUT BIT);                -- vai um
7 END som_8a;
8
9 ARCHITECTURE teste OF som_8a IS
10 BEGIN
11   abc: PROCESS (x, y, ze)
12     VARIABLE v : BIT_VECTOR (n DOWNT0 0); -- vai um interno
13   BEGIN
14     v(0) := ze;
15     abc: FOR i IN 0 TO n-1 LOOP
16       s(i) <= x(i) XOR y(i) XOR v(i);
17       v(i+1) := (x(i) AND y(i)) OR (x(i) AND v(i)) OR (y(i) AND v(i));
18     END LOOP abc;
19     zs <= v(n);
20   END PROCESS;
21 END teste;
```