



Circuitos Digitais II - 6882

Paulo Roberto de Oliveira

Universidade Estadual de Maringá
Departamento de Informática

Bacharelado em Ciência da Computação

Aula de Hoje

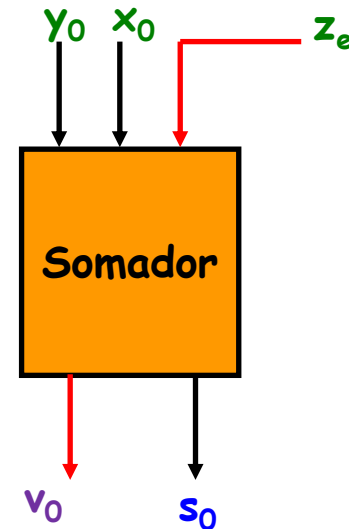
Projeto e Simulação dos seguintes circuitos:

- Somador de 2 bits
 - ✓ Comando FOR LOOP
 - ✓ Comando WHILE LOOP

HDL - Linguagem de Descrição de Hardware

Somador Completo de 1 Bit

Entradas			Saídas	
x_0	y_0	z_e	s_0	v_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



$$s_0 = x_0 \oplus y_0 \oplus z_e$$

$$v_0 = x_0 \cdot y_0 + y_0 \cdot z_e + x_0 \cdot z_e$$

ou

$$v_0 = (x_0 \oplus y_0) \cdot z_e + x_0 \cdot y_0$$

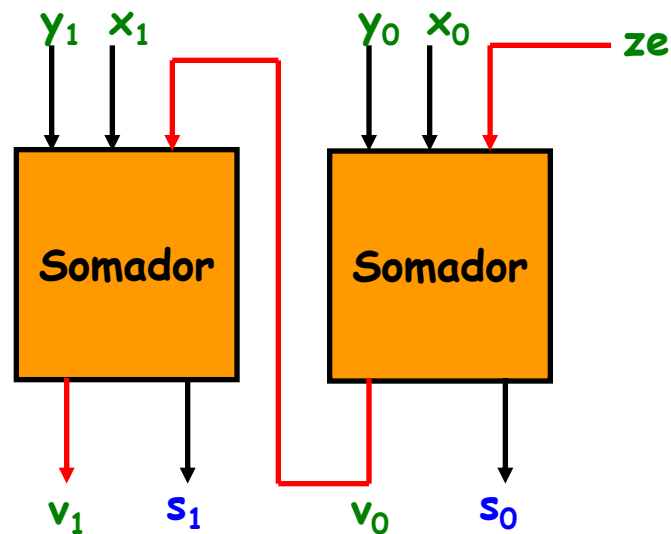
HDL - Linguagem de Descrição de Hardware

Comando de Repetição: FOR LOOP

- Exercício 01:
- Implemente um código para o circuito somador de 2 bits usando FOR LOOP.
- Use os seguintes nomes para as entradas e saídas:
 - x,y: para os dados;
 - ze para a entrada do vem-1;
 - v vai-1 interno;
 - zs para a saída do vai-1 final;
 - s para a saída da soma.
 - Veja figura a seguir

HDL - Linguagem de Descrição de Hardware

Somador Completo de 2 Bits



HDL - Linguagem de Descrição de Hardware

FOR LOOP

- Permite a repetição de instruções uma quantidade de vezes preestabelecida
- Restrição de uso dentro de procedimentos, funções e processos
- Um contador vai sendo incrementado ou decrementado a cada iteração até atingir um valor limite.
- Contador não pode ser alterado com operações de atribuição

Sintaxe

```
FOR contador IN valor_inicial TO|DOWNTO valor_final LOOP  
    comandos  
    ...  
END LOOP;
```

HDL - Linguagem de Descrição de Hardware

Solução

• Exercício 01:

ENTITY som_2bits IS

GENERIC (n	: INTEGER := 2);	-- numero de bits
PORT (x, y	: IN BIT_VECTOR (n-1 DOWNT0 0);	-- entradas do somador
ze	: IN BIT;	-- vem um
s	: OUT BIT_VECTOR (n-1 DOWNT0 0);	-- saida
zs	: OUT BIT);	-- vai um

END som_2bits;

ARCHITECTURE logica OF som_2bits IS

BEGIN

PROCESS (x, y, ze)

VARIABLE v : BIT_VECTOR (n DOWNT0 0); -- vai um interno

BEGIN

v(0) := ze;

FOR i IN 0 TO n-1 LOOP

s(i) <= x(i) XOR y(i) XOR v(i);

v(i+1) := (x(i) AND y(i)) OR (x(i) AND v(i)) OR (y(i) AND v(i));

END LOOP;

zs <= v(n);

END PROCESS;

END logica;

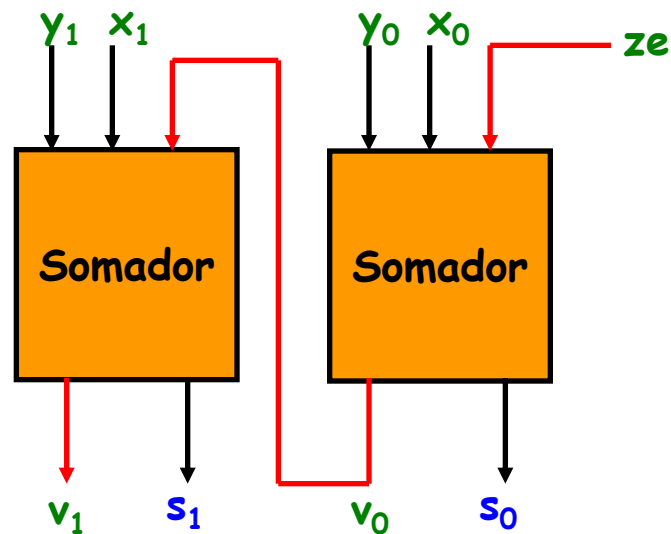
HDL - Linguagem de Descrição de Hardware

Comando de Repetição: WHILE LOOP

- Exercício 02:
- Implemente um código para o circuito somador de 2 bits usando WHILE LOOP.
- Use os seguintes nomes para as entradas e saídas:
 - x,y: para os dados;
 - ze para a entrada do vem-1;
 - v vai-1 interno;
 - zs para a saída do vai-1 final;
 - s para a saída da soma.

HDL - Linguagem de Descrição de Hardware

Somador Completo de 2 Bits



HDL - Linguagem de Descrição de Hardware

WHILE LOOP

- Permite a repetição de instruções se uma condição for verdadeira
- A iteração termina se a condição for falsa
- Restrição de uso dentro de procedimentos, funções e processos

Sintaxe

```
WHILE condição LOOP  
    comandos  
    ...  
END LOOP;
```

HDL - Linguagem de Descrição de Hardware

Solução

• Exercício 02:

ENTITY som_2bits IS

GENERIC (n	: INTEGER := 2);	-- numero de bits
PORT (x, y	: IN BIT_VECTOR (n-1 DOWNT0 0);	-- entradas do somador
ze	: IN BIT;	-- vem um
s	: OUT BIT_VECTOR (n-1 DOWNT0 0);	-- saida
zs	: OUT BIT);	-- vai um

END som_2bits;

ARCHITECTURE teste OF som_2bits IS

BEGIN

PROCESS (x, y, ze)

VARIABLE i : INTEGER ;

VARIABLE v : BIT_VECTOR (n DOWNT0 0); -- vai um interno

BEGIN

i := 0; -- deve ser atualizado a cada iteracao

v(0) := ze;

WHILE i <= n-1 LOOP -- executado enquanto verdadeiro

s(i) <= x(i) XOR y(i) XOR v(i);

v(i+1) := (x(i) AND y(i)) OR (x(i) AND v(i)) OR (y(i) AND v(i));

i := i+1;

END LOOP;

zs <= v(n);

END PROCESS;

END teste;