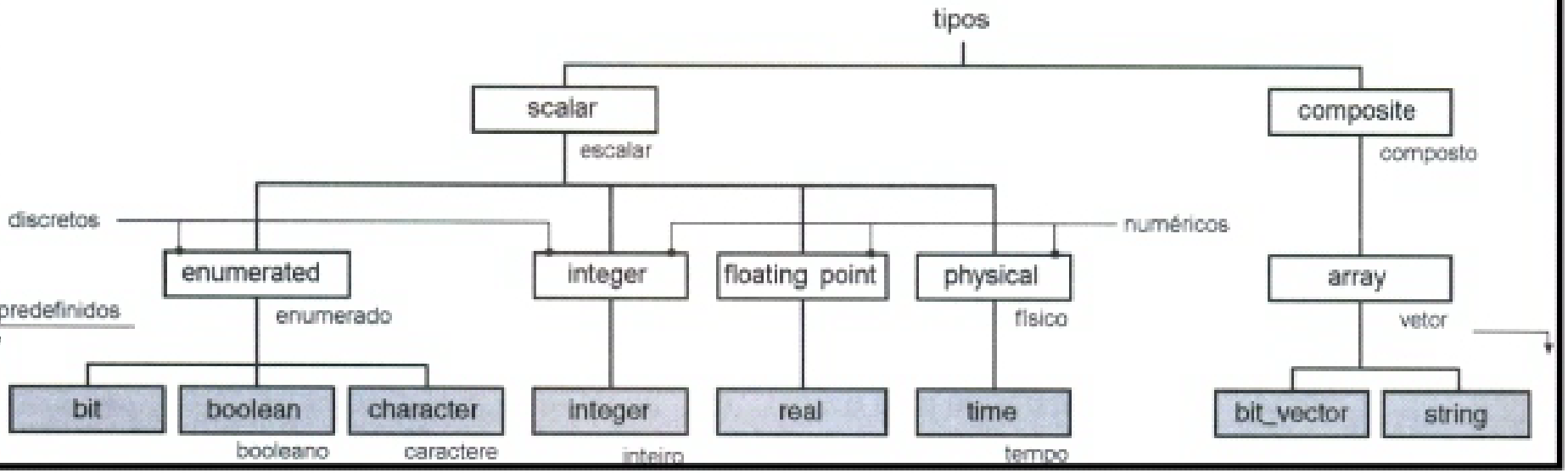
Dispositivos Lógicos Programáveis

Temas de hoje:

- Tipos;
- Exemplo de circuito de Seleção de Dados (DEMUX);
- Estrutura WHEN ELSE;
- Estrutura WITH SELECT;
- Estrutura BLOCK;
- Estrutura PROCESS;

Tipos

- Cada objeto (CONSTANT, VARIABLE, SIGNAL) deve ser declarado com um tipo;
- O tipo define o conjunto de valores que o objeto pode assumir, bem como o conjunto de operações que pode executar.



Tipos Escalares

TIPO	VALOR	EXEMPLOS
BIT	Um, zero	1,0
BOOLEAN	Verdadeiro, falso	TRUE, FALSE
CHARACTER	Caracteres ASCII	A,b,c,d,?,(
INTEGER	$-2^{31}-1 \leq x \leq 2^{31}-1$	123, 16#7B#, 2#11_11_011#
NATURAL	$0 \leq x \leq 2^{31}-1$	123, 16#7B#, 2#11_11_011#
POSITIVE	$1 \leq x \leq 2^{31}-1$	123, 16#7B#, 2#11_11_011#
REAL	$-3.65\text{x}10^{-47} \leq x \leq 3.65\text{x}10^{47}$	1.23, 1.23E+2
TIME	ps, ns, us, ms, sec, min, hr	1 us, 100 ps, 13ns

Tipos Compostos ou ARRAY

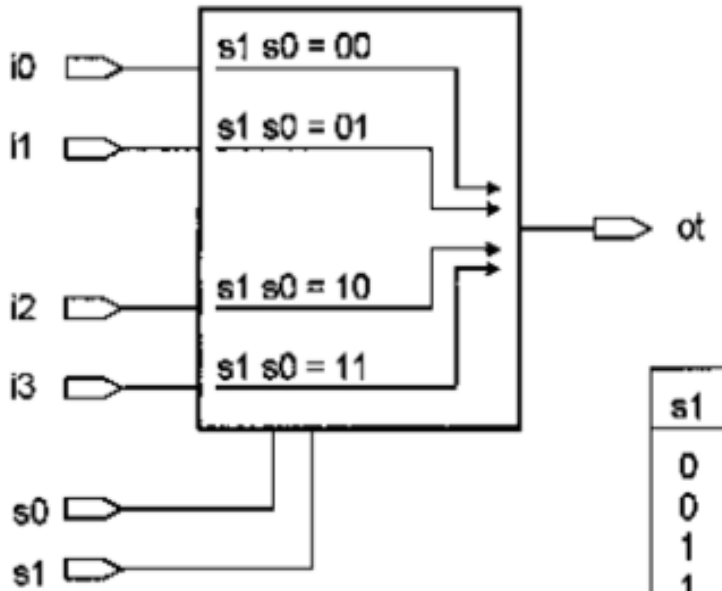
TIPO	VALOR	EXEMPLOS
BIT_VECTOR	Um, zero	"100101"
STRING	caracteres	"texto", "vhdl"

- São vetores unidimensionais
- Os elementos podem ser referenciados individualmente
- Ex.:
  - SIGNAL a,b: BIT_VECTOR (0 T0 4)
  - b(4) <= a(0)
  - a(0 TO 3) <= b(3 DOWNTO 0)

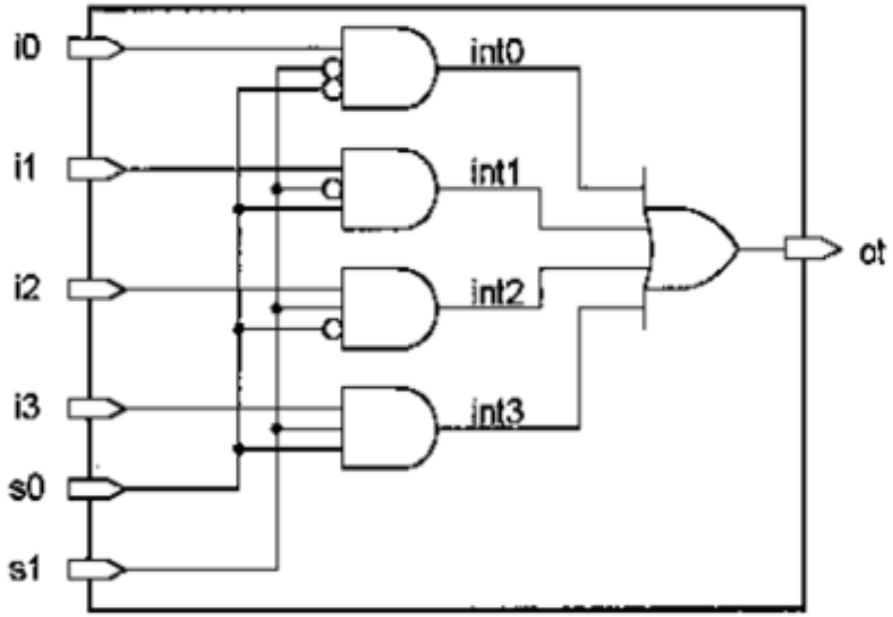
Circuito de Seleção de dados

A seleção dos dados que serão apresentados na saída é feita através das chaves s0 e s1.

Figura 1 – Circuito de Seleção de dados através das chaves s0 e s1.



s1	s0	ot
0	0	i0
0	1	i1
1	0	i2
1	1	i3



– Descrição utilizando sinais internos

```
ENTITY mux00 IS
  PORT (i0, i1, i2, i3   :  IN   BIT;  --entradas
        s0, s1           :  IN   BIT;  --seleção
        ot               :  OUT  BIT); --saída
END mux00;
ARCHITECTURE nivel_logico OF mux00 IS
  SIGNAL  int0, int1, int2, int3: BIT; --sinais
  BEGIN
      ot   <= int0 OR int1 OR int2 OR int3;
      int0 <= i0 AND NOT s1 AND NOT s0;
      int1 <= i1 AND NOT s1 AND     s0;
      int2 <= i2 AND     s1 AND NOT s0;
      int3 <= i3 AND     s1 AND     s0;
END nivel_logico;
```

– Estrutura “WHEN ELSE”

Permite a transferência do conteúdo de um sinal quando determinada condição ocorrer.

Esta lista de opções contêm PRIORIDADE, sendo maior na primeira linha e menor na última.

```
sinal_destino <= expressao_a  WHEN condicao_1 ELSE      -- condicao_1 = verdadeira
                 expressao_b  WHEN condicao_2 ELSE      -- condicao_2 = verdadeira
                 expressao_c;                           -- nenhuma condicao verdadeira
```

A primeira condição que retornar um valor verdadeiro irá definir o valor que é transferido para o sinal de destino.

```
s0 <= i0 OR i1   WHEN na= 8 OR  nb= 2 ELSE
      i1         WHEN na= 3 AND nb= 5 ELSE
      i0 AND i1  WHEN na= 7           ELSE
      i0;
```

– Estrutura “WHEN ELSE”

Exemplo:

```
architecture COND of BRANCH is
begin
    Z <= A when X > 5 else
         B when X < 5 else
         C;
end COND;
```

– Descrição utilizando a estrutura WHEN ELSE

```
 1  ENTITY mux_1 IS
 2    PORT (i0, i1, i2, i3      :   IN  BIT;
 3          s0, s1              :   IN  BIT;
 4          ot                  :   OUT BIT);
 5  END mux_1;
 6
 7  ARCHITECTURE teste OF mux_1 IS
 8   BEGIN
 9    ot <= i0 WHEN s1= '0' AND s0='0' ELSE
10          i1 WHEN s1= '0' AND s0='1' ELSE
11          i2 WHEN s1= '1' AND s0='0' ELSE
12          i3;
13  END teste;
```

– Estrutura “WITH SELECT”

Transfere um valor a um sinal de destino segundo uma relação de opções. A lista de opções não contêm prioridade.

Descrição do circuito de seleção de dados utilizando a estrutura WITH SELECT:

```
WITH expressao_escolha SELECT                    -- expressao_escolha =
 sinal_destino <= expressao_a WHEN condicao_1,                    -- condicao_1
                  expressao_b WHEN condicao_2,                    -- condicao_2
                  expressao_c WHEN condicao_3 | condicao_4,       -- condicao_3 ou condicao_4
                  expressao_d WHEN condicao_5 TO condicao_7,      -- condicao_5 ate condicao_7
                  expressao_e WHEN OTHERS;                        -- condicoes restantes
```

- Estrutura “BLOCK”

Divide o código em regiões para facilitar o entendimento da descrição, deixando-a mais organizada.

Pode-se declarar sinais dentro da estruturra BLOCK, sendo que estes servirão como “variáveis locais”.

A divisão do código não interfere no circuito sintetizado.

– Estrutura “BLOCK”

```
1  ENTITY mux_5 IS
2    PORT (i0, i1, i2, i3     :   IN  BIT;                          -- entradas
3          sel                :   IN  INTEGER RANGE 3 DOWNTO 0;     -- selecao
4          ot                 :   OUT BIT);                         -- saida
5  END mux_5;
6
7  ARCHITECTURE com_block OF mux_5 IS
8    SIGNAL global      : BIT_VECTOR(0 TO 1);
9  BEGIN
10   abc: BLOCK
11   BEGIN
12     ot <= i0 WHEN global = "00" ELSE
13           i1 WHEN global = "01" ELSE
14           i2 WHEN global = "10" ELSE
15           i3;
16   END BLOCK abc;
17
18   def: BLOCK
19     SIGNAL interno_def   : BIT_VECTOR(0 TO 1);   -- sinal visivel no bloco def
20   BEGIN
21     WITH sel SELECT
22       interno_def <= "00" WHEN 0,
23                      "01" WHEN 1,
24                      "10" WHEN 2,
25                      "11" WHEN OTHERS;
26     global <= interno_def;
27   END BLOCK def;
28 END com_block;
```

ortis 17/03/2015, 16:44:17

"SINAL GLOBAL",semelhante a "VARIÁVEL GLOBAL" em linguagem C

ortis 17/03/2015, 16:44:51

"SINAL LOCAL",semelhante a "VARIÁVEL LOCAL" em linguagem C

– Estrutura “PROCESS”

Um processo permite definir uma área contendo comandos sequenciais. Ele é composto de TRÊS regiões:

1 – Lista de sensibilidade;
2 – Local de declarações;
3 – Local dos comandos sequenciais.

```
abc: PROCESS (lista de sensibilidade)
  -- parte_de_declaracao_do_processo
    -- declaracao_de_constante;
    -- declaracao_de_variavel;
BEGIN
  -- parte_de_comandos_sequenciais
    -- comando_sequencial;
    -- comando_sequencial;
            ..
    -- comando_sequencial;
END PROCESS abc;
```

Lista de exercícios (aula 1, dia 11/03/15)

1 – Qual a estrutura do comando WHEN ELSE?

2 – Qual a estrutura do comando WITH SELECT?

3 – Qual a diferença funcional entre eles?

4 – Qual a estrutura do comando BLOCK e para que ele é utilizado?

5 – É possível ter um sinal global *sinal_1* e um sinal local sinal_2 dentro da estrutura BLOCK?

6 – Para que é utilizado o comando PROCESS?

7 – Qual a estrutura do comando PROCESS?

8 –Utilize a estrutura **WHEN ELSE** para atribuir à SAÍDA os valores ‘0’ ou ‘1’, de acordo com os sinais de entrada. Verifique o funcionamento do circuito através da tabela verdade.

Considera 000 como maior prioridade e 111 como menor prioridade.

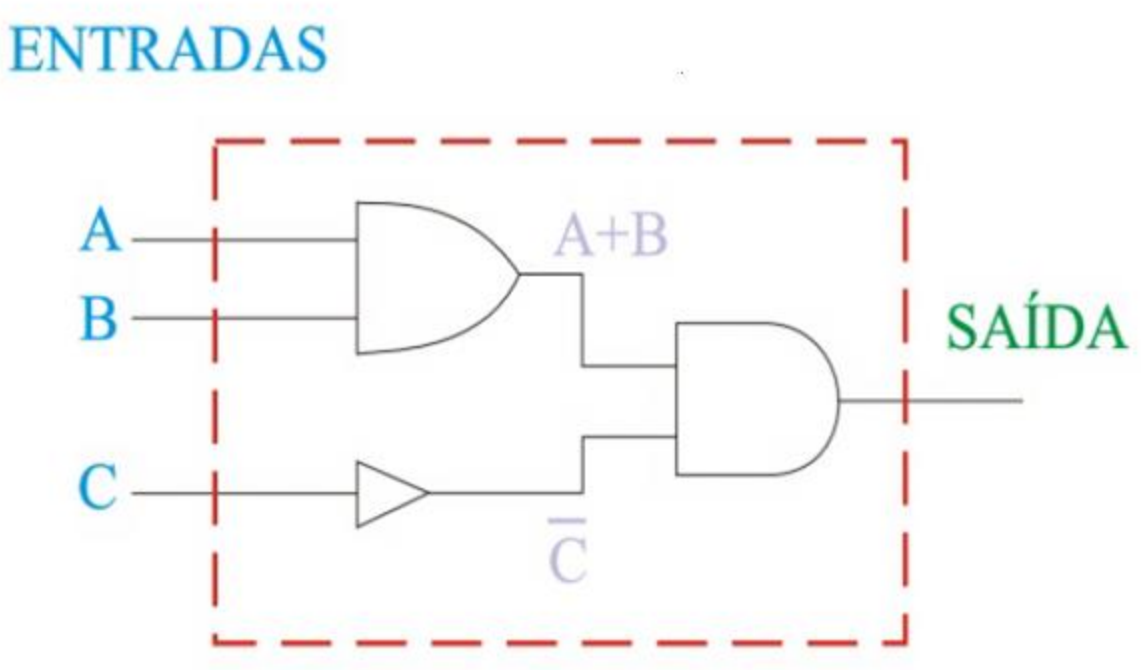


TABELA VERDADE

A	B	C	SAÍDA
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

9 – Declare o sinal chave_seletora (*tipo bit_vector*) na arquitetura e utilize a estrutura WITH SELECT para atribuir à SAÍDA os valores ‘0’ ou ‘1’, de acordo com os sinais de entrada (A, B e C). Verifique a Tabela verdade.

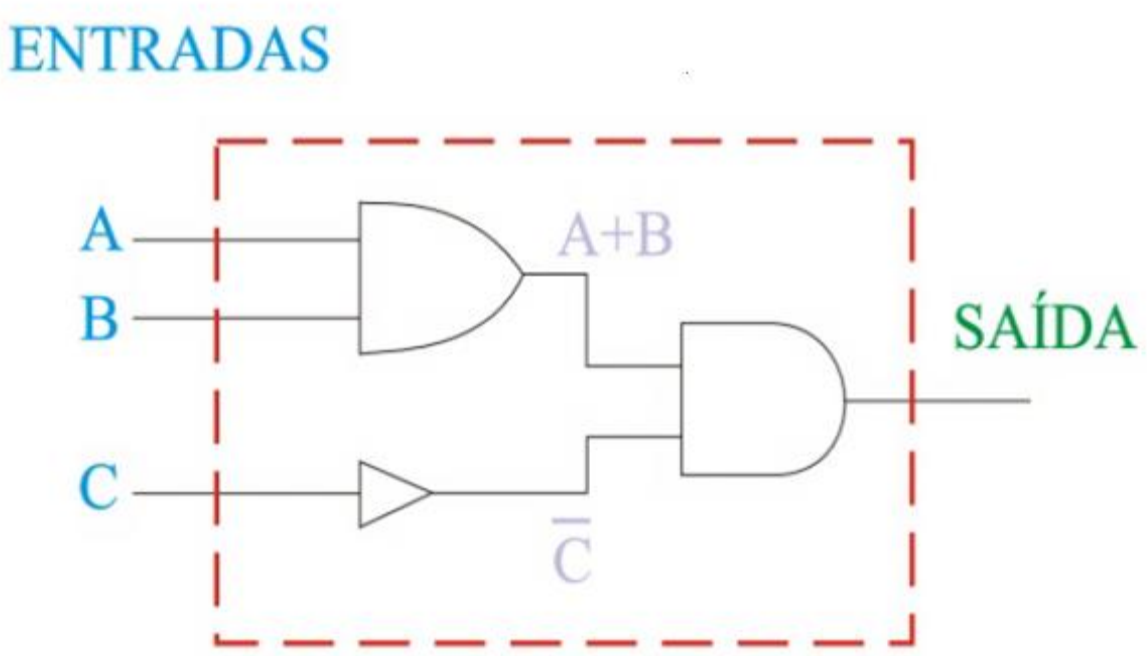


TABELA VERDADE

A	B	C	SAÍDA
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0