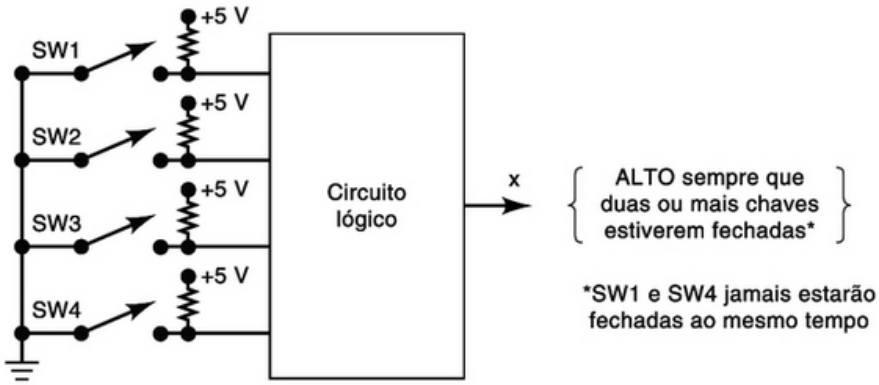
Sistemas Digitais Avançados

Prof. Átila Girão

Comandos concorrentes

Atribuição condicional de valores

1) Leia sobre as construções When – Else e With - Select retirados do livro VHDL – Descrição e síntese de circuitos digitais. Autor: D'Amore, Roberto.
2) Descreva em VHDL dois programas para sintetizar o circuito do problema 3, o primeiro utilizando a construção when-else e o segundo utilizando a construção with – select. Em seguida, faça o mesmo para o exercício 4, simule e teste na placa de desenvolvimento.
3) A figura abaixo mostra quatro chaves que fazem parte do circuito de controle de uma máquina copiadora. As chaves estão posicionadas em diversos pontos ao longo da trajetória do papel dentro da máquina. Cada chave está no estado normalmente aberta (nível lógico ALTO) e, quando o papel passa sobre a chave, ela é fechada (nível lógico BAIXO). É possível que mais de uma chave seja fechada ao mesmo tempo, porém é impossível que SW1 e SW4 sejam fechadas simultaneamente. Projete um circuito lógico que gere a saída em nível ALTO sempre que duas ou mais chaves estiverem fechadas ao mesmo tempo. Use mapa K e aproveite as vantagens das condições de irrelevância. (Retirado do livro: Sistemas Digitais: princípios e aplicações. Autor: Tocci, Ronald.)



4) Considere quatro sensores para um carro: um sensor de ignição, um sensor para o cinto de segurança, um sensor para os faróis e um sensor para as portas. Os estados dos sensores são definidos da seguinte forma:

Sensor	Nível Alto	Nível Baixo
Ignição	Motor ligado	Motor desligado
Cinto de segurança	Travado	Destravado
Faróis	Ligados	Desligados
Portas	Fechadas	Abertas

Deseja-se projetar um circuito combinacional que acione um alarme caso o motor esteja ligado e o cinto destravado ou as portas abertas, ou caso o motor esteja desligado e os faróis acesos. Inclua um display de 7 segmentos que irá mostrar a letra A caso o alarme esteja ativo, a letra C caso o alarme venha da condição do cinto de segurança, a letra F caso o alarme venha da condição dos faróis e a letra P caso o alarme venha da condição das portas.

3.3 CONSTRUÇÃO " WHEN ELSE "

A construção " WHEN ELSE " permite a transferência condicional de um sinal, e segue o formato ilustrado no Quadro 3.3.1. Nessa construção, uma lista de opções é apresentada estabelecendo qual valor de uma expressão deve ser transferido a um sinal de destino. A primeira condição que retorna o valor verdadeiro define o valor que é transferido para o sinal de destino.

```
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
```

Quadro 3.3.1 Construção " WHEN ELSE " ou " atribuição condicional de sinal ".

O Quadro 3.3.2 contém algumas linhas de código que ilustram a construção " WHEN ELSE ". O sinal de destino " s0 " pode receber diretamente o valor dos sinais " i0 ", " i1 " ou o resultado de uma operação lógica entre esses sinais. As opções de escolha são definidas por expressões que retornam um valor booleano. Como a construção " WHEN ELSE " define uma prioridade na ordem das opções, o circuito equivalente corresponde a uma cadeia de seletores (ver Figura 3.3.1). Cada seletor é comandado por blocos que detectam as opções; caso a opção seja verdadeira, o seletor transfere o valor de uma expressão, caso contrário, transfere o valor de um outro seletor.

```
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;
```

Quadro 3.3.2 Parte de um código empregando a construção " WHEN ELSE ".

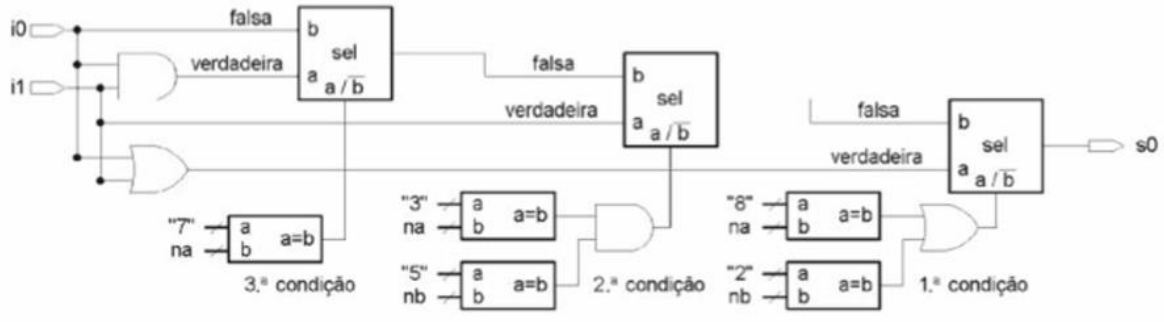


Figura 3.3.1 Circuito equivalente ao código do Quadro 3.3.2.

No Quadro 3.3.3 é apresentado um código completo empregando a construção " WHEN ELSE " para descrever o circuito de seleção proposto na Figura 3.2.1. Neste caso, a descrição possui um estilo mais próximo do comportamento do circuito.

```
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;
```

Quadro 3.3.3 Circuito de seleção empregando a construção " WHEN ELSE ".

3.4 CONSTRUÇÃO " WITH SELECT "

A construção " WITH SELECT " transfere um valor a um sinal de destino segundo uma relação de opções. Todas as condições de seleção devem ser consideradas, e elas devem ser mutuamente exclusivas. A lista de opções nessa construção não contém uma prioridade, como se observa na construção " WHEN ELSE ". Essas construções seguem o formato apresentado no Quadro 3.4.1.

```
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
```

Quadro 3.4.1 Construção " WITH SELECT " ou " atribuição selecionada de sinal ".

As opções podem ser agrupadas através do delimitador " | ", equivalendo, neste caso, a uma condição " ou " entre elas. De modo semelhante, as palavras reservadas " TO " e " DOWNTO " podem ser empregadas para delimitar uma faixa de condições de um tipo escalar. A palavra reservada " OTHERS " é válida, como última alternativa, para englobar as condições restantes.

O valor de retorno da expressão de escolha deve ser um tipo discreto ou um vetor unidimensional. Uma expressão de escolha tendo como retorno um valor do tipo não discreto como " REAL ", por exemplo, não é válida, pois o número de opções de escolha é infinito. Uma boa prática de programação é manter a expressão de escolha simples. Expressões complexas devem ser avaliadas isoladamente, e o resultado transferido para o campo da expressão de escolha.

No Quadro 3.4.2 é apresentada a parte de um código contendo dois exemplos com construção " WITH SELECT ". No primeiro, a expressão de escolha é o sinal " s0 " do tipo " CHARACTER ", e as condições de escolha ilustradas são do tipo: única, uma ou mais condições, faixa de condições e condições restantes. No segundo exemplo, a expressão de escolha retorna um valor do tipo " BIT ", e as duas condições de escolha possíveis são apresentadas. Como a construção não identifica nenhuma prioridade nas opções, o circuito equivalente corresponde a um circuito de seleção comandado por um decodificador que detecta as condições contidas no comando (ver Figura 3.4.1).

```
WITH s0 SELECT    -- s0 tipo CHARACTER
  x0 <= i0 AND i1 WHEN 'a',                 -- condicao unica
        i0 OR  i1 WHEN 'b'| 'c',            -- uma ou outra condicao
        i0 XOR i1 WHEN 'd' TO 'g',          -- faixa crescente de condicoes
        i0        WHEN 'x' DOWNTO 'k',      -- faixa decrescente de condicoes
        i1        WHEN OTHERS;              -- condicoes restantes

WITH b1 AND b0 SELECT  -- b1 e b0 tipo BIT
  x1 <= i0 WHEN '0',
        i1 WHEN '1';
```

Quadro 3.4.2 Parte de um código contendo construções " WITH SELECT ".

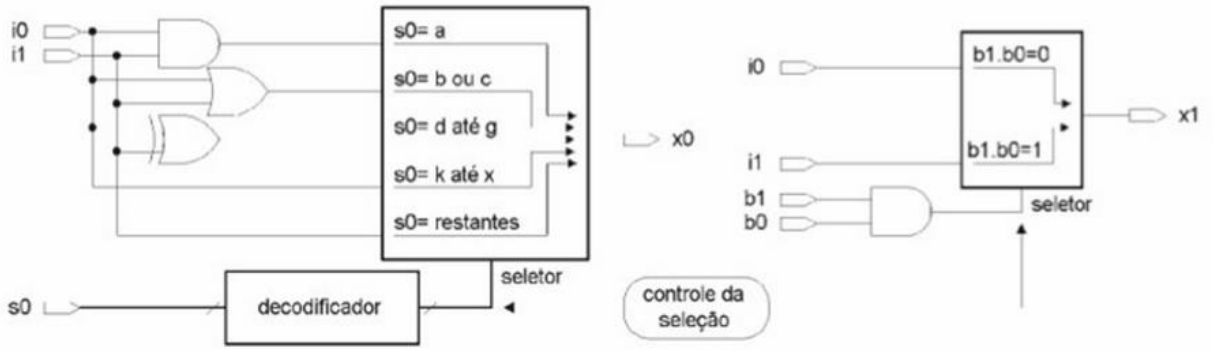


Figura 3.4.1 Circuito equivalente aos códigos do Quadro 3.4.2.

No Quadro 3.4.3 é apresentada uma nova descrição para o circuito da Figura 3.2.1. Na declaração da entidade, as portas de entrada responsáveis pela seleção foram concatenadas no sinal interno denominado " sel ". Isso se deve ao fato de que uma operação de concatenação na expressão de escolha não é válida. Observações pertinentes à expressão de escolha são apresentadas no Capítulo 4.

```
1  ENTITY mux_9 IS
2    PORT (i0, i1, i2, i3  : IN  BIT;
3          s0, s1          : IN  BIT;
4          ot              : OUT BIT);
5  END mux_9;
6
7  ARCHITECTURE teste OF mux_9 IS
8    SIGNAL sel : BIT_VECTOR (1 DOWNTO 0);
9  BEGIN
10   sel <= s1 & s0;
11   WITH sel SELECT
12     ot <= i0 WHEN "00",
13           i1 WHEN "01",
14           i2 WHEN "10",
15           i3 WHEN "11";
16 END teste;
```

Quadro 3.4.3 Circuito de seleção empregando a construção " WITH SELECT ".