

**Ex. 1** Projete um circuito deslocador com 8 bits na entrada ( $e_i, i \in \{0..7\}$ ) e que permite deslocamentos de uma posição para a esquerda e para a direita – atenção com os bits de saída  $s_0$  e  $s_7$ . Sua resposta deve conter o circuito que efetua os deslocamentos do bit  $s_i$ , bem como a composição de 8 destes circuitos para implementar o deslocador de 8 bits.

**Ex. 2** Estenda a resposta do Ex. 1 para permitir rotações de um bit nos dois sentidos.

**Ex. 3** Estenda a resposta do Ex. 1 de forma a manter o sinal de números representados em complemento de dois nos deslocamentos.

**Ex. 4** Repita o Ex. 1, de forma a que cada saída  $s_i$  pode transmitir  $e_i$ , ou  $e_{i-1}$ , ou  $e_{i+1}$ .

**Ex. 5** Projete um circuito combinacional que produz em sua saída uma versão deslocada da entrada, conforme especificado pela Equação 1.

$$\begin{aligned} E : \mathbb{B}_{16}, \quad S : \mathbb{B}_{32}, \quad d : \mathbb{B}_4 \\ \text{circ} : (\beta_{16} \times \beta_4) \mapsto \beta_{32} \\ \text{circ}(E, S, d) \equiv S = E \times 2^d \end{aligned} \quad (1)$$

**Ex. 6** Suponha que uma biblioteca VHDL contenha a seguinte primitiva, que é um *circuito combinacional*: `mult-por-1(A,B,S,R,N) ≡ R <= ( S ? A+B : 0&A )` sendo  $A$  e  $B$  inteiros representados em  $N$  bits,  $R$  é um inteiro representado em  $N + 1$  bits,  $S$  é um bit, e  $X\&Y$  representa a concatenação de  $X$  com  $Y$ ,  $N$  é o parâmetro que indica a largura do componente `mult-por-1`, e  $(x ? y : z)$  é a expressão de seleção da linguagem C.

- Usando várias instâncias de `mult-por-1`, mostre como implementar um multiplicador combinacional de  $6 \times 4$  bits (seis×quatro);
- Supondo que o pior caso do tempo de propagação do componente `mult-por-1` seja proporcional a  $N$ , qual o pior caso do tempo de propagação do multiplicador? Justifique;
- Usando registradores de largura apropriada, mostre como transformar o circuito do item (a) para permitir a multiplicação de mais de um número simultaneamente, i.e. mostre como segmentar o multiplicador;
- Dê uma especificação para a temporização do seu projeto no item (c) e a justifique.

**Ex. 7** Acrescente à Unidade de Lógica e Aritmética vista em aula as operações de deslocamento para E/D e rotação para E/D.

**Ex. 8** Em algumas aplicações de processamento digital de sinais a operação MADD (*multiply-add*) ocorre com alguma frequência. Um trecho de código em C que utiliza esta operação é mostrado ao lado. Mostre como implementar uma unidade funcional que efetue a operação MADD, quando todos os  $b[i]$  contém uma potência de 2.

```
short a[N], b[N], c[N], m[N];
int i;

for (i=0; i<N; i++) {
    // m <= MADD(a,b,c)
    m[i] = a[i]*b[i]+c[i];
}
```