

Laboratório 6

1. Rotação binária é uma operação que consiste em, dado uma palavra de n bits, fazer um reposicionamento dos bits na palavra de maneira circular, mantendo a ordem. Isto é, quando o reposicionamento é diferente de um múltiplo de n , o bit mais significativo da palavra rotacionada é aquele de posição anterior, no vetor original, ao bit menos significativo. Ex.: $x_3 x_2 x_1 x_0$, rotacionado em 2 é $x_1 x_0 x_3 x_2$.

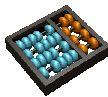
Uma forma de implementar essa operação em hardware é através do circuito chamado *barrel shifter*. Dados uma palavra binária $W = w_3 w_2 w_1 w_0$, um valor de rotacionamento $S = s_1 s_0$, a saída $Y = y_3 y_2 y_1 y_0$ será conforme a tabela verdade mostrada abaixo:

| s_1 | s_0 | y_3 | y_2 | y_1 | y_0 |
|-------|-------|-------|-------|-------|-------|
| 0 | 0 | w_3 | w_2 | w_1 | w_0 |
| 0 | 1 | w_0 | w_3 | w_2 | w_1 |
| 1 | 0 | w_1 | w_0 | w_3 | w_2 |
| 1 | 1 | w_2 | w_1 | w_0 | w_3 |

- a) **Implemente em VHDL** um *barrel shifter* de 4 bits a partir da tabela verdade acima. **Simule** seu funcionamento. Dica: utilize multiplexadores 4:1.
 - b) Utilizando o `demo_setup`, **execute** sua implementação na DE1, utilizando as *switches* como entradas e os *leds* como saídas.
- 2.
- a) **Implemente em VHDL** um somador *carry look-ahead* (CLA) de 4 bits. Verifique sua implementação **simulando**. Compare, usando o analisador de tempo, o caminho crítico e seu tempo dessa implementação com a do laboratório anterior (somador *ripple-carry* de 4 bits). Qual tem o menor tempo? Utilizando o `demo_setup`, **execute** sua implementação na DE1, utilizando as *switches* como entradas e os *leds* como saídas.
 - b) **Implemente em VHDL** um somador de 8 bits com CLA parcial, utilizando dois somadores CLA de 4 bits, interconectados em cascata (carry-out o CLA menos significativo conectado ao carry-in do CLA mais significativo). Compare, usando o analisador de tempo, o caminho crítico e seu tempo dessa implementação com a do laboratório anterior (somador *ripple-carry* de 8 bits). Qual tem o menor tempo?
 - c) **Implemente em VHDL** um somador *carry look-ahead* (CLA puro) 8 bits, sem nenhuma ligação em cascata ou do tipo ripple carry. Para isto, você terá que estender as equações de CLA de 4 bits (dadas em aula) para 8 bits. Compare, usando o analisador de tempo, o caminho crítico e seu tempo dessa implementação com as implementações anteriores de 8 (*ripple carry*, CLA parcial). Qual tem o menor tempo?

3. [Opcional]

- a) Baixe o arquivo `CLA_test_8bits.zip` em material complementar no site da disciplina. Crie um projeto com `CLA_test_8bits.bdf` como top-level na pasta e adicione seu somador



CLA de 8 *bits* do item b). **Execute** o analisador de *timing*. Grave o valor de **frequência máxima** obtida. **Repita** os passos deste item com o somador do item c). Houve diferença nos valores encontrados? Quais são essas frequências máximas? Qual das implementações tem maior frequência?

Observação.: Essa análise permite definir, por exemplo, qual a frequência máxima de um processador. Nas próximas aulas essas questões serão aprofundadas.