

### Lista de Exercícios No. 02

- 1) Um ULA que manipula funções de lógica e aritmética é representada pela Figura 1 e tem seu funcionamento determinado pela Tabela 1.

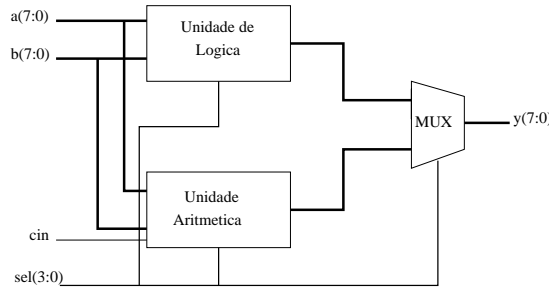


Figura 1: Unidade de Lógica e Aritmética.

sel	Operação	Função	Unidade
0000	$y \leq a$	Transfere o valor de a	Aritmética
0001	$y \leq a+1$	Incrementa a	Aritmética
0010	$y \leq a-1$	Decrementa a	Aritmética
0011	$y \leq b$	Transfere o valor de b	Aritmética
0100	$y \leq b+1$	Incrementa b	Aritmética
0101	$y \leq b-1$	Decrementa b	Aritmética
0110	$y \leq a+b$	Soma a e b	Aritmética
0111	$y \leq a+b+cin$	Soma a e b com carry	Aritmética
1000	$y \leq \text{NOT } a$	Complementa a	Lógica
1001	$y \leq \text{NOT } b$	Complementa b	Lógica
1010	$y \leq a \text{ AND } b$	AND de a com b	Lógica
1011	$y \leq a \text{ OR } b$	OR de a com b	Lógica
1100	$y \leq a \text{ NAND } b$	NAND de a com b	Lógica
1101	$y \leq a \text{ NOR } b$	NOR de a com b	Lógica
1110	$y \leq a \text{ XOR } b$	XOR de a com b	Lógica
1111	$y \leq a \text{ XNOR } b$	XNOR de a com b	Lógica

Tabela 1: Operações da ULA.

Projetar e implementar a ULA apresentada em VHDL. C e simule a solução verificando se funciona conforme esperado.

- 2) Projetar e implementar em VHDL o circuito digital que funcione como um *vector shifter*. Nesse tipo de circuito, a saída deve ser o valor deslocado para a esquerda do valor de entrada. O tamanho da saída é 2x o tamanho da entrada. A saída deve ser escolhida por um valor de entrada (sel). A Tabela 2 apresenta o funcionamento de um *vector shifter* com uma entrada de tamanho 4 bits e com valor “1111”. A saída será uma das linhas (row) escolhida de acordo com uma entrada.

row(0):	0	0	0	0	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>
row(1):	0	0	0	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0
row(2):	0	0	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0	0
row(3):	0	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0	0	0
row(4):	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0	0	0	0

Tabela 2: Operação de um *vector shifter*.

deslocada de uma posição à esquerda. A saída é determinada por um valor de entrada (sel) e corresponderá a uma das linhas.

- 3) Toda entrada de um circuito digital, quando fornecida por chaves ou botões, é acompanhada por algumas oscilações nos momentos de transição. A onda desejada reflete a entrada durante os momentos de estabilidade do sinal e, para garantir essa estabilidade, existe um pequeno atraso que só é utilizado para garantir a estabilidade. Gerar uma onda estabilizada com base numa onda com oscilações é chamado *debounce*. Para implementá-lo, é necessário um clock com frequência alta o suficiente para que a entrada seja amostrada várias vezes antes de garantirmos uma saída estável. Isso significa que estamos interessados num “zoom” do sinal de entrada, especificamente nos momentos de oscilações. A Figura 2 apresenta o funcionamento de um circuito de *debounce*. O circuito de *debounce* só considera como estável um sinal quando o sinal é lido pela terceira vez com o mesmo valor (observe o momento exato da transição).

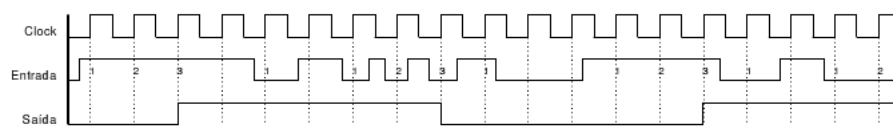


Figura 2: Circuito de *debounce*.

Implemente em VHDL um circuito que receba como entrada um sinal *w* e forneça como saída o sinal *z*, que é a entrada *w* com *debounce* de 3 ciclos de clock (conforme o exemplo da Figura 2. Sua entidade principal deve ser chamada *debounce*. Gere uma forma de onda de entrada e simule o circuito. Procure simular diversas situações para garantir que seu circuito esteja funcionando adequadamente.