



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO – Departamento de Engenharia Elétrica
Sistemas Digitais – ELE 08752 - Primeira Prova Parcial 2019/2- 18/10/2019

Aluno1: _____
Aluno2: _____

1ª QUESTÃO (3,5 pontos): Multiplicador de números em complemento de dois.

O algoritmo de Booth pode ser usado para multiplicar dois números em complemento de dois, **x_{in}** e **y_{in}** de n bits, gerando um produto em complemento de dois **p_{out}** de 2n bits, e precisa de n iterações para gerar o resultado.

Após a inicialização, o multiplicador é carregado na metade direita do registrador de produto, enquanto as outras posições são preenchidas com '0's, incluindo um bit extra inicialmente '0' que deve ser colocado à direita do produto.

O algoritmo é então o seguinte:

- se os dois bits mais à direita do produto são "10", o multiplicando deve ser subtraído da metade esquerda do registrador de produto e o resultado deve ser deslocado aritmeticamente à direita;
- se os bits são "01", então o multiplicando deve ser somado à metade da esquerda do produto e o resultado aritmeticamente deslocado à direita;
- finalmente, se os bits são "00" ou "11" então apenas o deslocamento aritmético à direita deve ocorrer.

Aplicando-se este procedimento n vezes o resultado é o produto dos dois números com sinal (Veja exemplo na tabela abaixo).

Descreva o circuito em VHDL que realiza o produto de dois números em complemento de dois de acordo com o algoritmo acima, considerando n = 8. Inclua um sinal que notifique que o multiplicador está ocioso (**p_{idle}**) e pronto para uma nova operação, um sinal **p_{start}** que inicie a operação e um sinal (**p_{done}**) que avise que a multiplicação solicitada foi executada e que o resultado em **p_{out}** está pronto.

Dica1- O complemento de dois de um número pode ser obtido invertendo-se todos os *bits* do número e somando-se 1 ao resultado. Por exemplo, em oito *bits*: (+6) = 00000110 e (-6) = complemento de dois de 00000110 = 11111001+1=11111010

Note que a representação em complemento de dois é diferente da representação em sinal magnitude onde (-6) seria (10000110), entretanto o *bit* mais significativo também é o *bit* de sinal.

Dica2- Em complemento de dois, fazer A-B = A+(-B) onde -B é o complemento de dois de B. Além disto, o resultado da operação em complemento de dois em n bits tem n bits, mesmo que a soma gere um número de n+1 bits.

Dica2- Em VHDL pode-se usar *alias* para acessar parte de um vetor. Por exemplo, se **x** é um número em sinal-magnitude de 8 *bits*:

```
signal x: std_logic_vector(7 downto 0);  
alias magn_x: std_logic_vector(6 downto 0) is x(6 downto 0);  
alias sign_c: std_logic is x(7);
```

As alterações devem ser feitas no signal x, mas partes deste vetor podem ser lidas usando magn_x e sign_x.

Dica3- Para que o circuito faça cada passo em um ciclo de *clock*, deve existir um circuito combinacional que realize a escolha entre as operações e que realize a operação escolhida.

Dica4- O deslocamento aritmético de 1 *bit* para a direita consiste em deslocar o número repetindo o *bit* de sinal. Desta maneira realiza-se a divisão por dois de números com sinal. Por exemplo,

(+6) = 00000110 => 00000011 = (+3) e

(-6) = 11111010 => 11111101 = (-3) (complemento de dois de 11111010 = 00000010 + 1 = 00000011 = +3)

Exemplo: O multiplicando e o multiplicador são "11110010" (-14) e "00001110" (+14), respectivamente.

A resposta é 11111111 00111100 = -196 (complemento de dois de 11111111 00111100 = 00000000 11000011 + 1 = 00000000 11000100 = 196)

Iterações	Procedimento	Multiplicando	Produto		
			Esquerda	Direita	Extra
0	Inicialização (Multiplicador carregado em Produto_Direita)	11110010	00000000	00001110	0 0
1	Bits 00 – Não fazer operação Deslocamento aritmético		00000000	00000111	1 0
2	Bits 10 – Produto_Esquerda - Multiplicando Deslocamento aritmético		00000000+ 00001110 00001110	00000111	0 1 1

3	Bits 11 – Não fazer operação Deslocamento aritmético		00000011 10000001 1
4	Bits 11 – Não fazer operação Deslocamento aritmético		00000001 11000000 0 1
5	Bits 01 – Produto_esquerda + Multiplicando Deslocamento aritmético		00000001+ 11110010 11110011 11000000 1 11111001 11100000 0 0
6	Bits 00 – Não fazer operação Deslocamento aritmético		11111100 11110000 0 0
7	Bits 00 – Não fazer operação Deslocamento aritmético		11111110 01111000 0 0
8	Bits 00 – Não fazer operação Deslocamento aritmético		11111111 00111100 0

2ª QUESTÃO (3,5 pontos): Interface Teclado PS2

O circuito na Figura 1 implementa um receptor no protocolo PS2, gerando na saída **dout** que são os 8 bits de dados recebidos pelo protocolo PS2 cujo diagrama de tempo é descrito na Figura 2. A Figura 3 apresenta a carta ASMD do circuito onde se pode ver que o valor de **dout** é válido somente quando **rx_done_tick** está ativo (em '1'). **Descreva em VHDL** para um circuito de interface de um teclado PS2 com o programa multiplicador da 1ª Questão.

O circuito de interface deve instanciar o circuito **ps2_rx** e tratar os dados recebidos juntando as informações do teclado (veja a Figura 4) para gerar os sinais de saída: **k_done_tick** que é um sinal de *status* que é '1' quando tem dado novo de movimentação de uma tecla, **k_press** que é '1' se a tecla foi pressionada e '0' se ela foi solta, **k_normal** que é '1' se a tecla é normal e '0' se ela é estendida e **key(7 downto 0)** que é o *makecode* da tecla em 8 bits (1C_H para o A, 75_H para a seta para cima↑).

Funcionamento do teclado: Cada vez que uma tecla é pressionada o teclado envia para a FPGA o *makecode* da tecla (1C_H no caso do A e E0 75_H no caso da seta para cima↑). Quando uma tecla é solta, o teclado envia o *breakcode* para a FPGA (F0 1C_H no caso do A e E0 F0 75_H no caso da seta para cima↑). O pacote do protocolo PS2 é como o da Figura 2, portanto apenas 1 *byte* é enviado por vez.

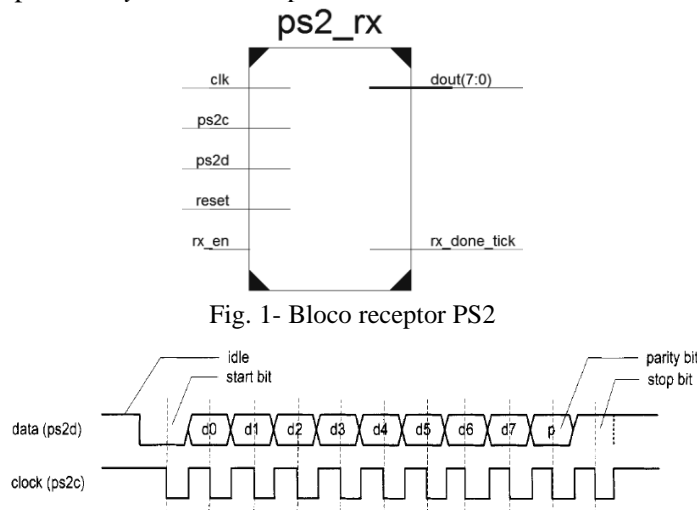


Fig. 1- Bloco receptor PS2

Fig. 2- Diagrama de tempo na comunicação do Teclado para a FPGA

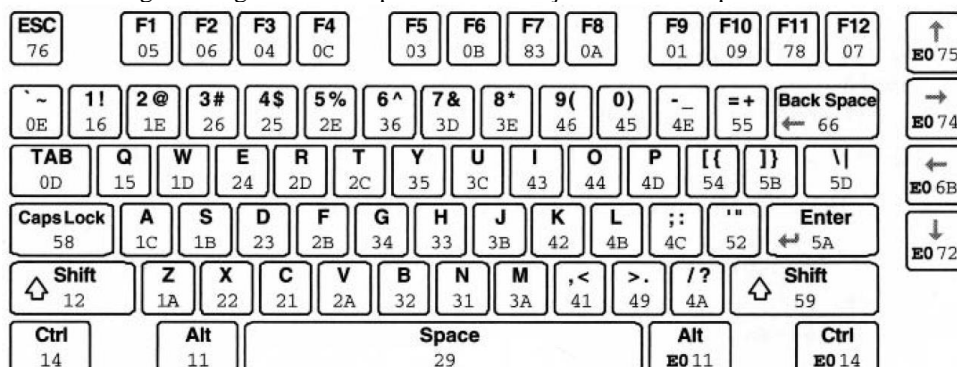


Fig. 4- Teclado PS2 - Makecode das teclas

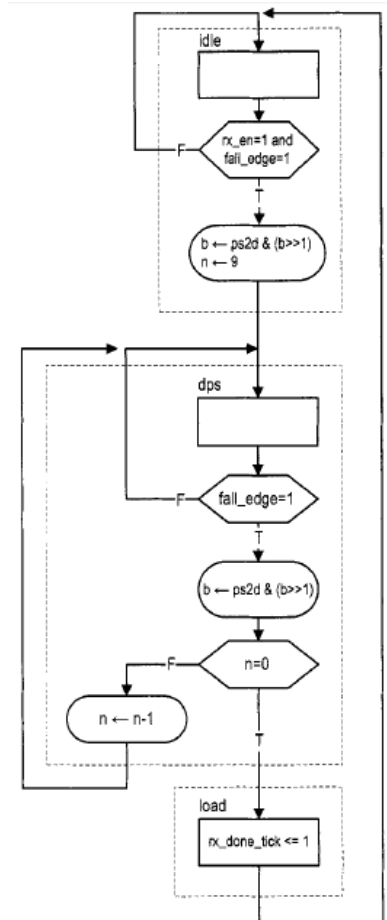


Fig. 3- Carta ASMD do receptor

3ª QUESTÃO (3 pontos): Programa geral de multiplicação

Projete em VHDL o circuito que instancia o multiplicador Booth da 1ª Questão, o circuito de interface com o Teclado PS2 da 2ª Questão, fazendo as corretas conexões entre eles. Para entrar com dados para o multiplicador o usuário deve pressionar a letra x (ou y) e em seguida os 0's e 1's do bit mais para o menos significativo; os valores digitados devem ser armazenados no vetor **x_reg** (**y_reg**) fazendo deslocamento à esquerda para formar um número em complemento de 2 com 8 bits. Ao pressionar o sinal de igual (=) o circuito deve comandar o multiplicador para efetuar a operação com o **x_reg** e o **y_reg** armazenados. Cada vez que recebe um igual (=) o circuito verifica se o multiplicador está ocioso (**p_idle** = '1') e solicita a ele a realização da operação fazendo **p_start** = '1' com os valores atuais de **x_reg** e **y_reg**. Quando o multiplicador finaliza a operação e faz **p_done** igual a '1', o valor obtido deve ser armazenado em **z_reg** em 16 bits.

Usando-se o circuito de multiplexação entre os *displays* com entrada em hexadecimal (descrito no Capítulo 4 no arquivo `list_ch04_15_disp_hex.vhd`) e uma chave é possível ver os valores de **x_reg** e **y_reg** (com `sw(0)` = '0') e o valor de **z_reg** (`sw(0)` = '1').

Note que os operandos **x_reg** e **y_reg** têm 8 *bits*, portanto cada um pode ser mostrado em dois dígitos do *display* nessa ordem. O *bit* de sinal deve ser invertido e colocado como ponto decimal à esquerda do dígito mais significativo de cada operando. O número **z_reg** por outro lado deve ter 16 *bits*, ocupando os quatro dígitos do *display* e o sinal de **z_reg** deve ser apresentado no ponto decimal mais à esquerda. Além disto, o valor de **x_reg**, **y_reg** e **z_reg** estarão em complemento de dois e, portanto, para mostrá-los é preciso calcular o seu complemento caso eles sejam negativos.

Os arquivos `.vhd` gerados no projeto ou usados do livro devem ser compactados e enviados por e-mail (para eliete.caldeira@ufes.br) ao fim do período de avaliação. O nome do arquivo compactado deve ter o nome dos componentes da dupla.

Os alunos podem consultar o livro e os códigos utilizados no laboratório.

Não pode haver conversa com pessoa de fora da dupla, nem troca de e-mails ou mensagens.

Haverá verificação dos códigos e, sendo detectada origem fora dos materiais permitidos, a nota será alterada.

Bom trabalho! *Profª. Eliete Caldeira*