

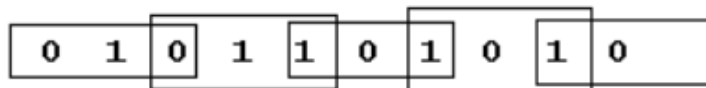
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO – Departamento de Engenharia Elétrica  
Sistemas Digitais – ELE 08572 - Primeira Prova Parcial 2022/1- 27/06/2022

Aluno1: \_\_\_\_\_ Aluno2: \_\_\_\_\_

**1ª QUESTÃO** (4,0 pontos): Multiplicador de números positivos pelo método de Booth Radix4.

O algoritmo de Booth pode ser usado para acelerar a multiplicação de dois números positivos, **a\_in** e **b\_in** de  $n$  bits, gerando um produto **p\_out** de  $2n$  bits, e precisa de  $n/2$  iterações para gerar o resultado.

O algoritmo agrupa os bits do multiplicador em tríades (*triplets*) de forma que a primeira fica mais à direita e é composta pelos dois bits menos significativos do multiplicador e um '0' extra. As demais são montadas de forma que tenham apenas um bit em comum com as tríades de cada lado. Por exemplo, para o número  $362_{10}=0101101010_2$



As tríades para  $k$  de 0 a  $(n/2)-1$  vão conter os bits  $2k+1$  até  $2k-1$  exceto para a primeira que tem o bit '0' extra.

As operações para cada tríade são mostradas na tabela, sendo que o multiplicando deve estar alinhado com o bit  $2k$ , ou seja, o bit central da tríade em cada caso.

Triplet	Operação	Ação
000	Fazer nada	Fazer nada
001	Somar o multiplicando	+ multiplicando* $2^{2k}$
010	Somar o multiplicando	+multiplicando* $2^{2k}$
011	Somar o dobro do multiplicando	+multiplicando* $2^{2k+1}$
100	Subtrair o dobro do multiplicando	-multiplicando* $2^{2k+1}$
101	Subtrair o multiplicando	-multiplicando* $2^{2k}$
110	Subtrair o multiplicando	-multiplicando* $2^{2k}$
111	Fazer nada	Fazer nada

Para o número de 10 bits do exemplo:

$k$	triplet	Bits	Operação	Detalhamento
0	100	10extra	-multiplicando* $2^{2k+1}$	-multiplicando* $2^{2(0)+1}$ = -multiplicando* $2^1$
1	101	321	-multiplicando* $2^{2k}$	-multiplicando* $2^{2(1)}$ = -multiplicando* $2^2$
2	101	543	-multiplicando* $2^{2k}$	-multiplicando* $2^{2(2)}$ = -multiplicando* $2^4$
3	011	765	+multiplicando* $2^{2k+1}$	+multiplicando* $2^{2(3)+1}$ = +multiplicando* $2^7$
4	010	987	+multiplicando* $2^{2k}$	+multiplicando* $2^{2(4)}$ = +multiplicando* $2^8$

As operações da tabela devem ser feitas no número de bits do resultado que é  $2n$ . Sendo assim, o multiplicando deve ser armazenado em um vetor deste tamanho.

A cada iteração  $k$  é incrementado até  $n/2-1$ , para tratar todos os bits do multiplicador. Este algoritmo é mais rápido porque trata dois bits do multiplicador por vez.

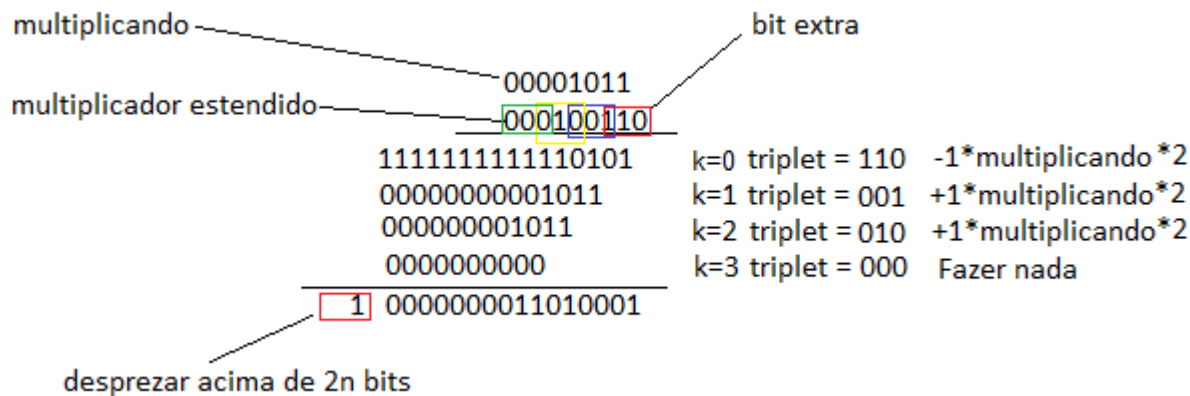
**Descreva o circuito em VHDL considerando  $n = 8$**

Dica1- Desenhe a carta AMSD para o circuito antes de descrever.

Dica2- 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.

Dica3- As atribuições feitas a sinais ou **ports** dentro do **process** não são realizadas imediatamente, apenas no **end process**. Sendo assim se o código precisa realizar uma operação sobre um sinal em seguida usar o resultado para outra operação, então não pode fazer as duas operações no mesmo **process**.

A figura abaixo mostra o exemplo de uma multiplicação de dois números de 8 bits.



## 2ª QUESTÃO (3,0 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** o circuito **keyboard** de interface do teclado PS2. Este circuito 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<sub>H</sub> para o A, 75<sub>H</sub> 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<sub>H</sub> no caso do A e E0 75<sub>H</sub> no caso da seta para cima↑). Quando uma tecla é solta, o teclado envia o *breakcode* para a FPGA (F0 1C<sub>H</sub> no caso do A e E0 F0 75<sub>H</sub> 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.

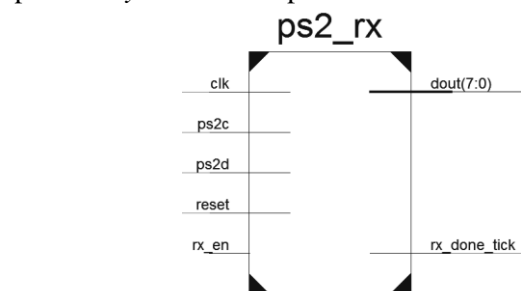


Fig. 1- Bloco receptor PS2

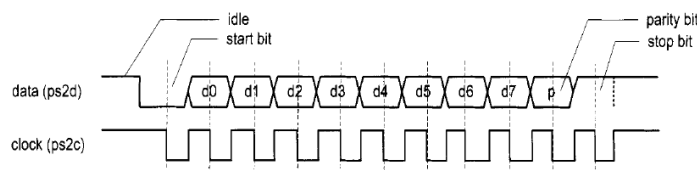


Fig. 2- Diagrama de tempo na comunicação do Teclado para a FPGA

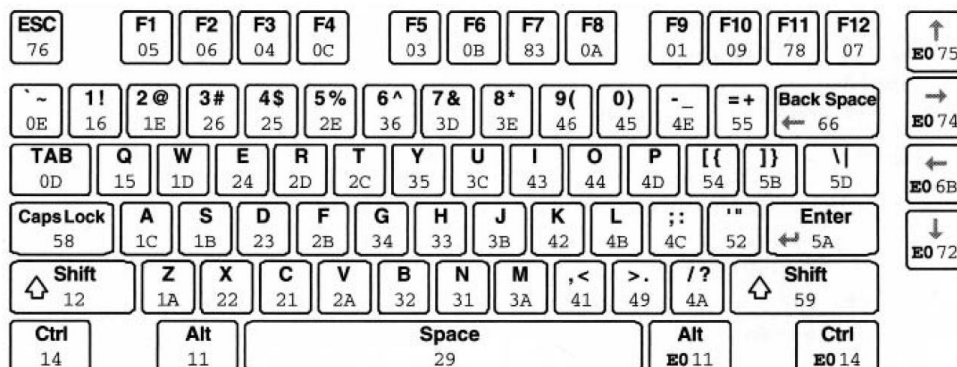


Fig. 4- Teclado PS2 - Makecode das teclas

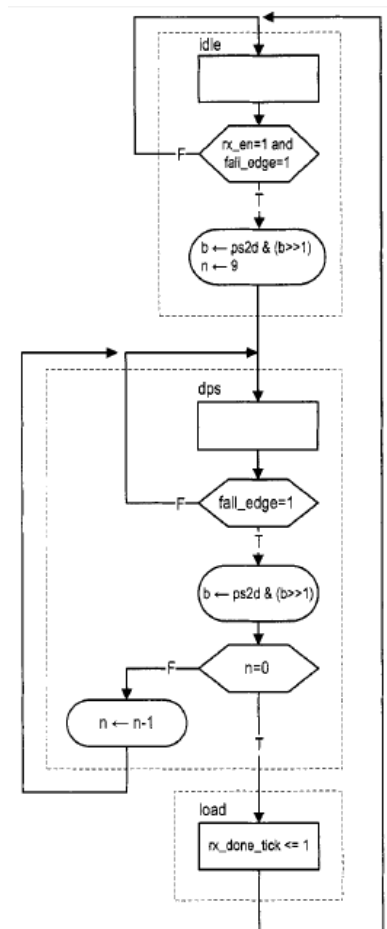


Fig. 3- Carta ASMD do receptor

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

Projete em VHDL o circuito **mult\_8b\_top** que instancia o multiplicador Booth\_Radix4 da 1ª Questão, o circuito de interface com o Teclado PS2 da 2ª Questão, fazendo as corretas conexões entre eles.

Projete também uma máquina de estados que responda às entradas recebidas do teclado armazenando dados, enviando ao multiplicador, esperando a multiplicação finalizar e mostrando os operandos e o resultado na saída.

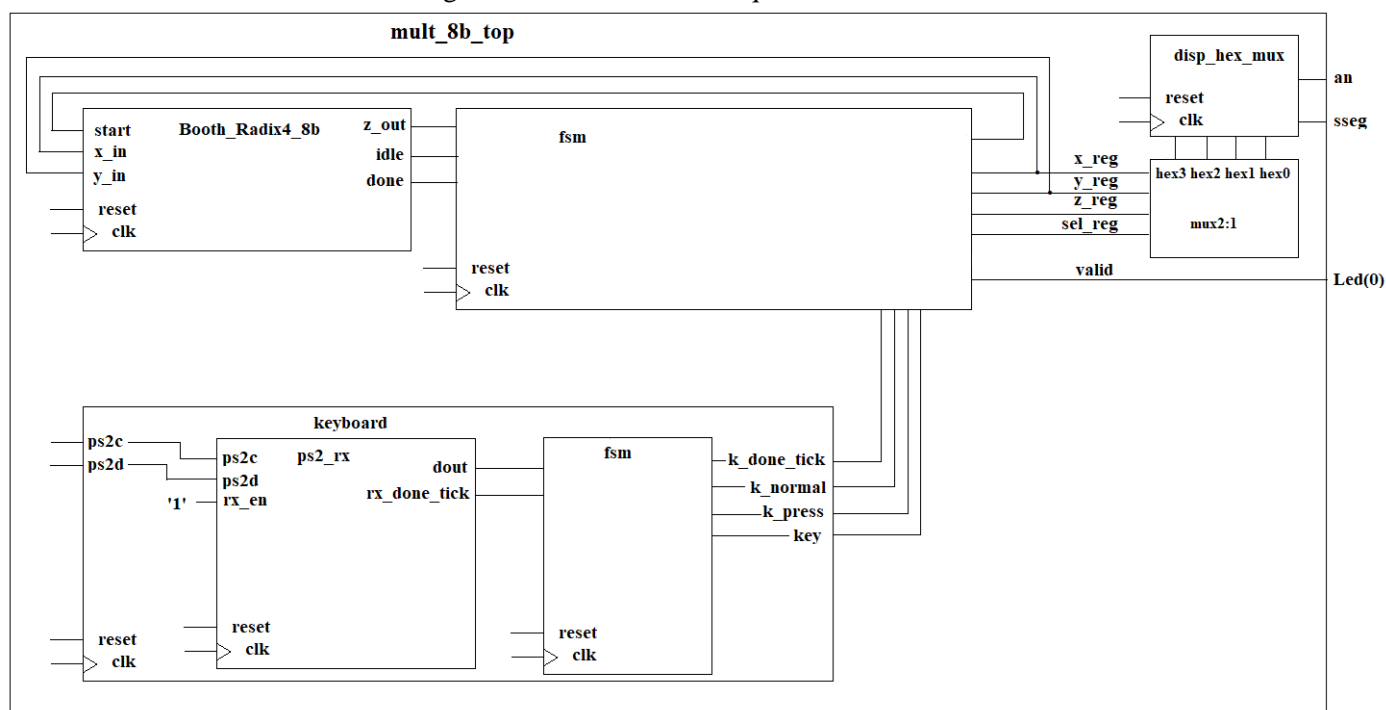
Os caracteres do teclado devem ser tratados quando a tecla é **pressionada**.

Para entrar com dados no teclado para o multiplicador o usuário deve pressionar a letra x (ou y), depois = e em seguida os dois dígitos hexadecimais do número (por exemplo, se  $x = 10_D = 00001010_b = 0A_H$  o usuário deve digitar **x**, **=**, **0** e **A** na sequência; se  $y = 27_D = 00011011_b = 1B_H$  as teclas pressionadas devem ser, na sequência, **y**, **=**, **1** e **B**; os valores digitados devem ser convertidos para binário e armazenados nos respectivos bits de **x\_reg** (**y\_reg**).

Ao pressionar a tecla s (de *start*) o circuito deve comandar o multiplicador para efetuar a operação com o **x\_reg** e o **y\_reg** armazenados. Cada vez que recebe um s, o circuito espera que o multiplicador fique ocioso (**idle** = '1') e solicita a ele a realização da operação fazendo **start** = '1' com os valores atuais de **x\_reg** e **y\_reg**. Quando o multiplicador finaliza a operação e faz **done** igual a '1', o valor obtido deve ser armazenado em **z\_reg** em 16 bits.

O sistema deve ter um sinal **valid** (internamente tem que ter um **valid\_reg**) que expresse se o resultado em **z\_reg** é consistente com **x\_reg** e **y\_reg** presentes no momento. Ou seja, este sinal deve indicar se o valor de **z\_reg** foi obtido do produto dos valores atuais de **x\_reg** e **y\_reg**. Para isto, sempre que um novo valor de x ou y é digitado no teclado este sinal é desativado e permanece assim até que nova multiplicação seja realizada, quando ele é ativado novamente. Para mostrar o resultado, o sinal **valid** deve ser mostrado em um LED da placa. Além disso, os sinais **x\_reg** e **y\_reg** devem ser mostrados no display de 7 segmentos quando **sel\_reg** = '0' e o sinal **z\_reg** deve ser mostrado quando **sel\_reg** = '1' (mux2:1). A mudança no valor de **sel\_reg** é determinada pelo teclado: ao pressionar a tecla i (de *input*) o sistema deve fazer o sinal **sel\_reg** = '0' e ao pressionar a tecla o (de *output*), deve fazer o sinal **sel\_reg** = '1'. Além disso, toda vez que um novo valor de **x\_reg** ou **y\_reg** for recebido, o sistema deve fazer **sel\_reg** = '0' para mostrar essa entrada e toda vez que um novo valor de **z\_reg** for gerado, o sistema deve fazer **sel\_reg** = '1' para mostrar esse valor.

A figura abaixo mostra um esquemático do sistema



A dupla pode e deve consultar e usar os códigos dos livros, **dando preferência aos códigos da segunda edição**. Os arquivos .vhd gerados no projeto ou usados dos livros devem ser colocados em um diretório que permita juntar o projeto e testar na placa. Esse diretório deve ser compactado e enviado via Classroom ao fim do período de avaliação. O nome do arquivo compactado deve ter o nome da dupla e os dois alunos devem mandar o mesmo arquivo no Classroom. Haverá verificação dos códigos e, sendo detectado o uso indevido de códigos entre duplas a nota será alterada.

Em caso de dúvida enviem e-mail para [eliete.caldeira@ufes.br](mailto:eliete.caldeira@ufes.br) e procurarei responder.

Bom trabalho! Profa. Eliete Caldeira