

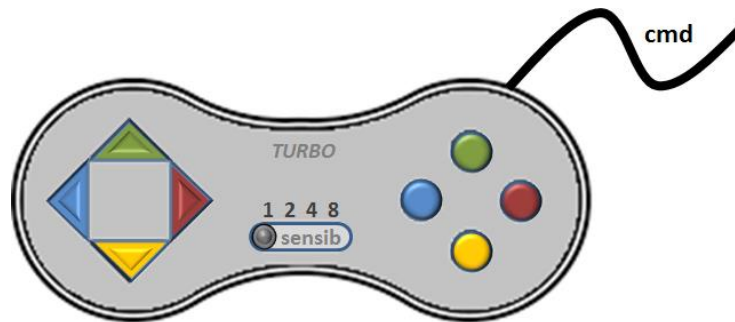
Sistemas Digitais I	Projeto	Data	MS
PCS3115			

Nome:

#USP:

Turma:

Tem-se como objetivo projetar o módulo de “turbo” para os botões de um controle de video-game com 8 botões, o que significa que os comandos relativos a cada botão são enviados ao console repetidamente se aquele botão for mantido pressionado. O projeto deve seguir uma estrutura com Fluxo de Dados e Unidade de Controle, provendo o seguinte comportamento:



- I. Como vários botões podem ser pressionados ao mesmo tempo, é usada uma representação de 8 bits para indicar quais os botões pressionados:  $B_7B_6B_5B_4B_3B_2B_1B_0$ . Por exemplo, caso os botões 7, 1 e 0 sejam pressionados ao mesmo tempo, o comando (**cmd**) a ser enviado para o console é “10000011”; já se nenhum botão é pressionado, a saída enviada ao console é “00000000”
- II. Para não deixar o modo turbo excessivamente sensível, é utilizada uma chave com controle de sensibilidade (**sensib**), que pode assumir os valores 1, 2, 4 ou 8. Isso faz com que, logo após o pressionamento de um conjunto de botões e seu envio ao console, o controle aguarde **sensib** bordas de clock antes de começar a repetir o valor dos botões pressionados; antes desse número de bordas ser atingido, a saída enviada é “00000000” (i.e., “nenhum botão pressionado”)

Os exemplos a seguir ilustram tal comportamento (cada valor de **Button** refere-se aos bits lidos durante a borda de subida do clock)

```

Sensib: 1
Button: 00000011 00000011 00000011 00000011 00000011 00100010 00100010 00100010
cmd:    00000011 00000000 00000011 00000011 00000011 00100010 00000000 00100010
                                     (obs: botões diferentes aqui)

Sensib: 2
Button: 00000011 00000011 00000011 00000011 00000011 00100010 00100010 00100010
cmd:    00000011 00000000 00000000 00000011 00000011 00100010 00000000 00000000

Sensib: 4
Button: 00000011 00000011 00000011 00000011 00000011 00100010 00100010 00100010
cmd:    00000011 00000000 00000000 00000000 00000000 00100010 00000000 00000000
  
```

### **Procedimento de resolução**

Sugere-se que você execute os seguintes passos para projetar o circuito (**não** é necessário responder a essas questões no seu código VHDL submetido ao Juiz, mas responde-las nesta ordem deve te ajudar a desenvolver o seu projeto).

- a) Qual seria um possível circuito para calcular um sinal de “Push”, que indica que um conjunto de botões foi pressionado e, portanto, o módulo turbo deve fazer algo diferente de enviar “00000000”?
- b) Construa o diagrama ASM de alto nível para esse circuito. Sugere-se que seja adotado o modelo Moore para a geração dos comandos.
- c) Liste os elementos necessários para a construção do Fluxo de Dados e então desenhe um diagrama esquemático com a interligação destes elementos. Elabore uma tabela contendo os sinais de controle e os sinais de status/condição e pense em uma descrição sucinta de cada um (essa descrição será útil para documentar o seu código).
- d) Construa o diagrama ASM da Unidade de Controle, permitindo a construção de um circuito digital síncrono. Para facilitar, assuma que o FD é ativado nas bordas de descida do clock: assim, os sinais de controle da UC podem ser habilitados na borda de subida, junto com cada transição de estados, e estarão estabilizados quando forem processados pelo FD naquele estado.
- e) Desenvolva as descrições em VHDL para os seguintes elementos: Unidade de Controle (nomeado turboUC); o Fluxo de Dados (nomeado turboFD); o projeto completo conectando turboUC e turboFD (nomeado turbo); e o testbench.

### **Instruções para Entrega**

Para fazer a submissão, há um link específico no e-Disciplinas. Acesse-o somente quando estiver confortável para enviar sua solução, ou seja, depois de tê-la testado usando o seu próprio testbench.

Em cada atividade, você pode enviar apenas um único arquivo codificado em UTF-8. Portanto, se seguir os passos sugeridos, coloque as entidades “turboUC”, “turboFD” e “turbo” em um único arquivo. O nome do arquivo não importa, e qualquer editor de código moderno suporta UTF-8 (e.g. Atom, Sublime, Notepad++, etc). Porém, a descrição VHDL dentro dele deve ter uma entidade “turbo” dentro dele, seguindo a interface fornecida como anexo a este enunciado: deve ter o mesmo nome, e o mesmo conjunto de sinais de entrada e saída. Caso contrário, o juiz não irá processar seu arquivo.

Quando acessar o link no e-Disciplinas, o navegador abrirá uma janela para envio do arquivo (como aconteceu no simulado). Selecione-o e envie para o juiz. Jamais recarregue a página de submissão, pois seu navegador pode enviar o arquivo novamente, o que vai ser considerado pelo juiz como um novo envio (possivelmente prejudicando sua nota final...). Caso desista do envio, simplesmente feche a janela.

Depois do envio, a página carregará automaticamente o resultado do juiz, quando você poderá fechar a janela. A nota dada pelo juiz é somente para a submissão que acabou de fazer. Sua nota na atividade poderá ser vista no e-Disciplinas e pode diferir da nota dada pelo juiz dependendo da estratégia de atribuição de notas utilizada pelo professor que montou o problema (veja a atividade3 do Simulado). Certifique-se de testar seu código e de não imprimir dados dentro dos códigos submetidos.

**Atenção:** não atualize a página de envio e não envie a partir de conexões instáveis (e.g. móveis) para evitar que seu arquivo chegue corrompido no juiz.