Universidade Federal De Campina Grande Departamento De Engenharia Elétrica
Laboratório De Arquitetura De Sistemas Digitais – LASD
Prof. Rafael B. C. Lima

Universidade Federal de Campina Grande

Aluno: _____
Matrícula: _____ Data: _____

Sprint 3 – Primeiro projeto no Quartus II

Descrição geral do problema: Primeiro projeto no Quartus II. Implemente um decodificador de hexa para 7segmentos, integre-o ao registrador de 4bits previamente desenvolvido e valide-o na placa Altera DE2.

Preparação para a sprint:

1. Revise os conceitos aprendidos na disciplina de Circuitos Lógicos. Sugestão: capítulos 1 ao 5 do livro de referência, "*Sarah Harris and David Harris. Digital Design and Computer Architecture, RISC-V Edition. 1. Morgan Kaufmann. 2021*".
2. Instale o Quartus II (ou utilize os PCs do LTD).
3. Assista o vídeo explicativo dessa Sprint.
4. Assista o tutorial de como simular módulos no ModelSim, utilizando waveforms.
5. Assista o tutorial de como criar testbenches e simular módulos no ModelSim.

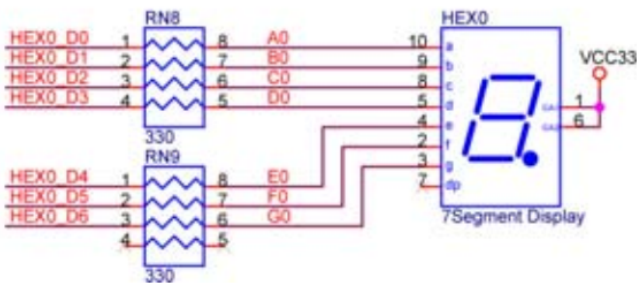
Requisitos mínimos:

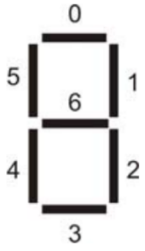
1. Configuração do ambiente de trabalho individual
   a) Crie o diretório "**D ou E:\LASD\20XX.X\SEUNOME_MATRICULA**". Cada aluno deverá usar o mesmo computador em todas as aulas do LASD. Sempre salvar seus arquivos nessa pasta;
   b) Abra o Quartus II 13.0;
   c) Menu "**File -> New Project Wizard**";
   d) Crie o Projeto, com nome "**Mod_Teste**", no seu diretório;
   e) Ignore o pedido de inclusão de arquivos já existentes;
   f) Selecione a família **CycloneII**, FPGA **EP2C35F672C6** e finish;
   g) Copie os seguintes arquivos, do Sigaa, para a pasta local do seu projeto: **Mod_Teste.v**, **LCD_TEST2.v**, **LCD_CONTROLLER.v** e **DE2_PIN_ASSIGNMENT.CSV;**
   h) Use o menu "**Assignments > import assignment**" para incluir o arquivo **DE2_PIN_ASSIGNMENT.CSV**
   i) Vá no menu: "**Assignments > Pin Planner**" e observe o mapeamento gerado, no passo anterior, entre os pinos físicos do FPGA e os TAGs do Mod_Teste;
   j) Adicione, em seu projeto, os arquivos **.v/.sv** do passo g). Na janela "**Project Navigator > pasta Files > Botão direito > Add/Remove files in project**". Selecione os arquivos, Add e OK. *Caminho alternativo: Menu > Project > Add/Remove files in Project"*

   Obs: Nas aulas subsequentes devemos abrir este projeto usando "File > Open Project" e NÃO "File > Open".

2. Dentro do módulo **Mod_Teste**, faça uma atribuição contínua entre o LEDG[0] e o botão KEY[1]. Compile , carregue na placa e teste seu primeiro código!

3. Faça um **assign** entre os leds do display de 7 segmentos **HEX0[0:6]** e as chaves **SW[6:0].** Varie as chaves e observe qual segmento está associado a cada **bit** de **HEX0.**



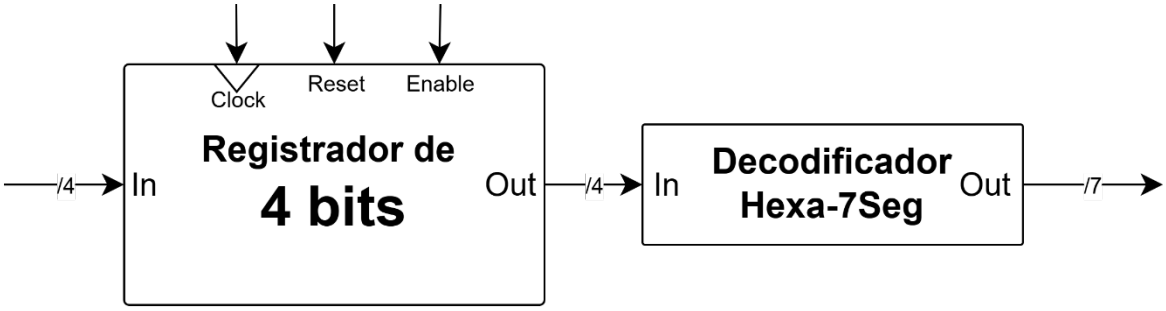


4. Projete um decodificador de hexadecimal para 7 segmentos.
    a) Crie o decodificador em uma estrutura de módulo. Para facilitar o reuso, salve-o em um arquivo **.v/.sv** separado.
    b) O decodificador deve possuir uma entrada de 4bits, para entrar um número de 4'h0 até 4'hF e uma saída de 7bits para conectar um display de 7 segmentos e poder visualizar o dígito.
    c) Para testar seu circuito, instancie um decodificador, no Mod_Teste, com o seguinte mapeamento de entradas e saídas:

    d) Dica: pesquise sobre a estrutura *case*. Implemente a lógica de funcionamento em alto nível.

5. Gere sinais de 4'b0000 até 4'b1111 através das chaves SW[11:8] e valide, manualmente, seu decodificador, observando os dígitos acesos no display HEX3. Chame o professor para Validar seu progresso até esse ponto.

6. Resgate o registrador de 4bits já implementado e validado da Sprint 2 e inclua um decodificador de hexa-7seg para visualizar sua saída.



7. Conecte botões e chaves às entradas, assim como um display de 7 segmento na saída. Gere sinais de teste manualmente e cheque se os resultados estão de acordo com o previsto. Chame o professor para Validar seu progresso até esse ponto.

Após o professor conferir seus testes, compacte todos os arquivos em um **.zip** e submeta-o no **SIGAA**

Desafio (Valendo +0,2 na média geral)

1. Simule a montagem do item 6, no ModelSim, utilizando waveforms para gerar os estímulos das entradas. Dica: assistir o vídeo.
2. Crie um testbench para simular sistematicamente diversos cenários de operação da mesma montagem do item 6. Dica: assistir o vídeo.

Módulo topo – Mod_Teste

```
`default_nettype none //Comando para desabilitar declaração automática de wires
module Mod_Teste (
//Clocks
input                   CLOCK_27, CLOCK_50,
//Chaves e Botoes
input       [3:0]       KEY,
input       [17:0]      SW,
//Displays de 7 seg e LEDs
output      [0:6]       HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, HEX6, HEX7,
output      [8:0]       LEDG,
output      [17:0]      LEDR,
//Serial
output                  UART_TXD,
input                   UART_RXD,
inout       [7:0]       LCD_DATA,
output                  LCD_ON, LCD_BLON, LCD_RW, LCD_EN, LCD_RS,
//GPIO
inout       [35:0]      GPIO_0, GPIO_1
);
assign      GPIO_1              =           36'hzzzzzzzzz;
assign      GPIO_0              =           36'hzzzzzzzzz;
assign      LCD_ON              =           1'b1;
assign      LCD_BLON            =           1'b1;
wire        [7:0]       w_d0x0, w_d0x1, w_d0x2, w_d0x3, w_d0x4, w_d0x5,
                        w_d1x0, w_d1x1, w_d1x2, w_d1x3, w_d1x4, w_d1x5;
LCD_TEST MyLCD      (
            .iCLK       ( CLOCK_50 ),
            .iRST_N     ( KEY[0] ),
            .d0x0(w_d0x0),.d0x1(w_d0x1),.d0x2(w_d0x2),.d0x3(w_d0x3),.d0x4(w_d0x4),.d0x5(w_d0x5),
            .d1x0(w_d1x0),.d1x1(w_d1x1),.d1x2(w_d1x2),.d1x3(w_d1x3),.d1x4(w_d1x4),.d1x5(w_d1x5),
            .LCD_DATA( LCD_DATA ),
            .LCD_RW  ( LCD_RW ),
            .LCD_EN  ( LCD_EN ),
            .LCD_RS  ( LCD_RS )
);
//---------- modifique a partir daqui --------
endmodule
```