Licenciatura em Engenharia Informática

Tecnologia dos Computadores (2018/2019)

– Trabalho #6 –
Introdução à Lógica Programável

Objetivo

O objetivo deste trabalho é introduzir de forma prática a utilização de lógica programável em larga escala (CPLDs e FPGAs). Para isso, será utilizada a linguagem VHDL, especificação de circuitos por desenho, e programação na placa Altera DE2.

Introdução

Este trabalho terá como objectivo a interligação de um contador binário a um descodificador para 7-segmentos. Na prática, irá desenhar um pequeno circuito digital capaz de contar de 0 até 15 (em binário) e que após descodificação apresenta o resultado em hexadecimal. Todo o projecto será executado utilizando lógica programável. É desta forma que atualmente se faz prototipagem rápida de sistemas digitais. O integrado que irá utilizar (Field Programmable Gate Array - FPGA) chama-se **EP2C35** e encontra-se do lado inferior direito da placa Altera DE2 (ver Figura 1).

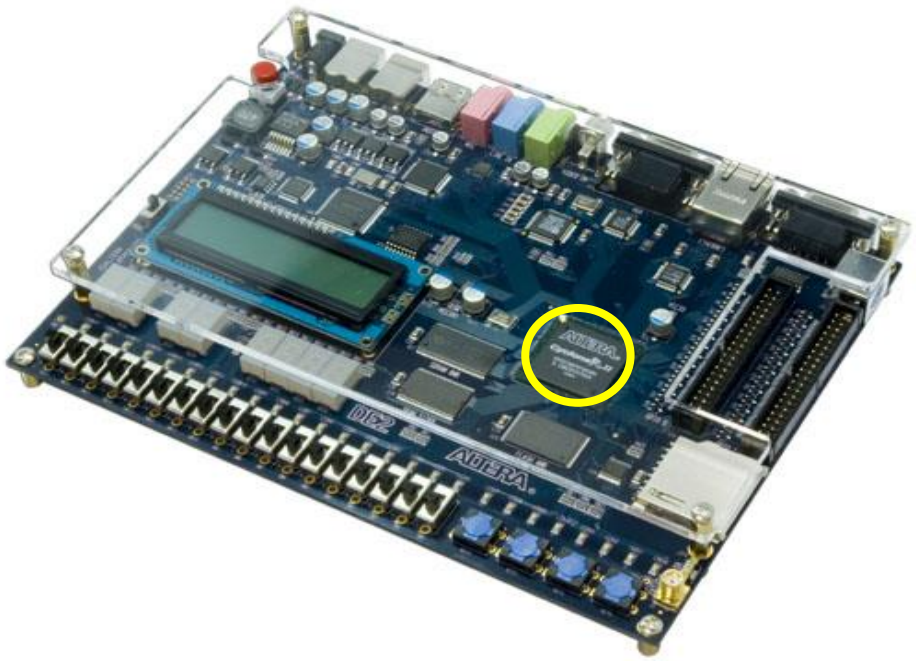

Figura 1 – Placa Altera DE2, com o FPGA Cyclone II EP2C35 assinalado a amarelo

Como irá ver, todo o projecto cabe facilmente na FPGA, que possui o equivalente a cerca de 35 000 elementos lógicos (Logic Elements - LE).

No final deste trabalho deverá conseguir:

- Desenhar um circuito utilizando o programa Quartus II;
- Programar o circuito na placa DE2 e testá-lo.

Circuito base

O circuito inicial a implementar nesta aula é o seguinte:

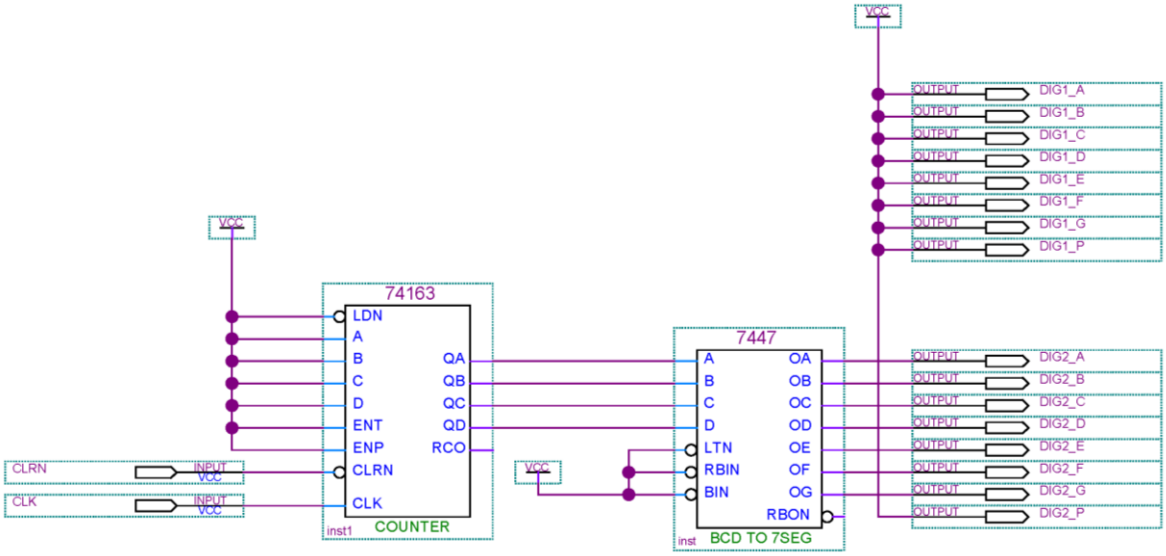

Figura 2 – Circuito contador com descodificador de 7-segmentos

Como pode ver, este circuito possui um contador binário síncrono (74163) ligado a um descodificador para 7 segmentos.

Um contador síncrono é um componente que incrementa o seu próprio valor (soma 1) a cada ciclo de um sinal de relógio. Um sinal de relógio é nada mais que um sinal periódico quadrado, que é frequentemente utilizado nos sistemas digitais para manter sincronizados os diversos elementos de um circuito .

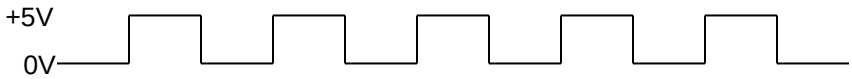

Figura 3 – Sinal de relógio

Habitualmente são usadas as transições (flancos positivos ou negativos) destes sinais para definir os instantes de sincronização.

Um descodificador para 7-segmentos é um componente que permite converter uma palavra binária de 4 bits para um outro padrão de bits de forma a possa ser representada num display de 7 segmentos.

Figura 4 – Display de 7 segmentos

Criação de novo projeto no Quartus

1. Comece por criar uma pasta para o seu projeto. Nesta pasta residirão todos os ficheiros relativos a este trabalho.

Certifique-se que:

- A pasta **"C:\altera\de2"** existe, contendo diversos ficheiros com a terminação VHD, entre outros. Caso isso não aconteça, faça o *download* do ficheiro **de2.zip** do inforestudante, e coloque os ficheiros existentes dentro deste arquivo comprimido nessa pasta. Os ficheiros constantes dessa pasta correspondem a uma biblioteca de funções que irá utilizar neste trabalho e em trabalhos futuros – para este efeito, deverá adicionar esta pasta às definições de bibliotecas (**pergunte ao professor como se faz**).
- Ao correr o programa, a opção "*Run Compilation, simulation or software build at a lower priority*" está ligada. Esta opção encontra-se em "Tools->Options->Processing".

Note que para utilizar este *software* em casa, deve instalar uma versão exactamente igual ao que é usada nas aulas (91sp2), cujo download pode ser feito a partir do site http://www2.deec.uc.pt/~jlobo/altera/ (se quiser, apenas por curiosidade, pode encontrar a última versão deste software em https://www.altera.com/products/design-software/fpga-design/quartus-ii/download.html – não aconselhamos, porém, a sua utilização para as aulas, visto que, por exemplo, já não inclui o simulador que usaremos nas aulas).

2. Execute o programa Quartus II.

Crie um novo ficheiro de projeto de sistemas digitais em interface gráfico (esquemático). Para isso faça "*File->New->Block Diagram/Schematic File*". Guarde o ficheiro utilizando o nome "*contador.bdf*" (*File->Save*) na pasta do seu projeto.

Responda "sim" à pergunta se quer criar um novo projeto baseado neste diagrama. Este ficheiro criado será o "topo" da hierarquia do seu projeto. Representa o circuito global que será implementado no circuito integrado. Carregue sucessivamente em "Next" até lhe ser pedido para selecionar qual a família de dispositivos para a qual deseja programar (*"Which device family do you wish to target?"*).

Selecione a família **Cyclone II**. No ecrã seguinte, ser-lhe-á pedido um dispositivo específico. Selecione o **EP2C35F672C6**.

Desenho do esquemático e compilação

3. Agora irá criar um ficheiro de projeto de sistemas digitais em interface gráfico (esquemático). Para isso faça "*File->New->Block Diagram/Schematic File*". Guarde o ficheiro utilizando o nome "*contador.bdf*" (*File->Save*) na pasta do seu projeto.

Crie o circuito apresentado na figura 2. Os componentes são adicionados fazendo "*Edit->Insert Symbol*", clicando duas vezes no rato, ou selecionando o botão com o símbolo de um AND e carregando na zona de desenho.

Existem inúmeros componentes pré-definidos que pode utilizar, nomeadamente os equivalentes à família TTL 74xx. Os componentes TTL encontram-se na biblioteca "*Others/maxplus2*".

Para ter acesso às funcionalidades correspondentes à placa DE2, é necessário que o projecto tenha conhecimento dos ficheiros que se encontram em "*c:\altera\de2*". Para isso, é necessário aceder previamente ao menu "*Project->Add/Remove Files in Project/User Libraries*" e adicionar a pasta "*c:\altera\de2"* à lista de diretórios do projeto.

Note também que os pinos de entrada da placa correspondem ao componente "*input*". Os pinos de saída correspondem ao componente "*output*". Note que para alterar o nome dos pinos deverá clicar duas vezes sobre o nome do mesmo.

Os componentes VCC e GND são também importantíssimos, como já aprendeu. No entanto, neste tipo de projetos não é necessário pensar na alimentação dos componentes (que, obviamente, são "integrados virtuais", ou seja, emulações).

Uma razão importante para a utilização do componente 7447 como conversor BCD-7 segmentos prende-se com o facto do *display* na placa ser de *ânodo comum*, acendendo cada LED quando se lhe aplica o nível lógico 0 e não o nível 1. Isto é, os *displays* de 7 segmentos são activados a 0, o mesmo acontecendo com as saídas do 7447.

4. Grave o ficheiro e compile-o. Verifique que não existem erros. Para isso faça "*Processing->Start Compilation"*.

Irão surgir diversos *warnings*, que neste caso não são significativos. Simplesmente avisam que existem saídas que estão sempre a 1 (o que é verdade).

Atribuição de pinos e programação da FPGA

5. O passo seguinte será programar o circuito na placa. Para visualizar a evolução do contador, vamos utilizar um botão para emular o sinal de relógio ("clock"). Vamos também utilizar um botão para efectuar o "clear". Para isso, deve adicionar outro *pin* de entrada, com o nome CLRN.

6. Deve agora fazer corresponder pinos físicos às entradas e saídas do seu circuito. Note que cada pino está ligado a um componente diferente na placa (por exemplo, um botão, um interruptor, ou um LED). Para isso deverá ir ao menu "*Assigments->Pins*". Após ter

chamado esse menu, surgirá a lista de entradas e saídas que está a utilizar. Selecione "*Location*" para cada uma das entradas e escreva o código do pino respetivo, consultando a tabela abaixo (ver manual da placa DE2, que pode encontrar no Inforestudante, para mais informação).

Atenção: Se se enganar na atribuição poderá queimar imediatamente o circuito. Verifique duas vezes as atribuições!

Entrada/Saída	Pino	Componente de Teste
CLK	PIN_P23	KEY[2]
CLRN	PIN_W26	KEY[3]
DIG2_A	PIN_AF10	HEX0[0]
DIG2_B	PIN_AB12	HEX0[1]
DIG2_C	PIN_AC12	HEX0[2]
DIG2_D	PIN_AD11	HEX0[3]
DIG2_E	PIN_AE11	HEX0[4]
DIG2_F	PIN_V14	HEX0[5]
DIG2_G	PIN_V13	HEX0[6]

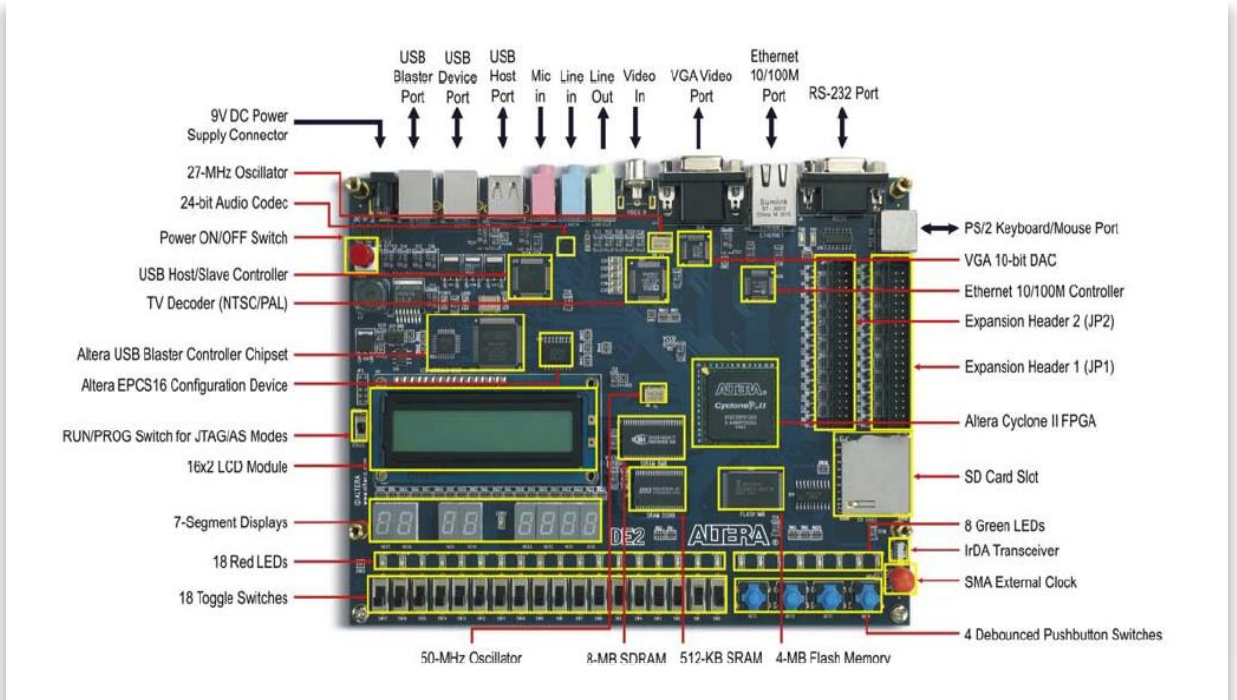

Figura 3 – A placa DE2

7. Compile o projecto. Não deverá obter nenhum erro (embora sejam gerados alguns avisos).

8. Irá agora programar o projeto no dispositivo. Ligue o cabo ao porto USB da sua máquina e ao respetivo conetor da placa DE2. Ligue a placa à alimentação.

9. Vá ao menu "*Tools->Programmer*". Na janela que surge, coloque um visto na *check box* "*Program/Configure*". Isso indica que o ficheiro que acabou de criar deve ser programado. Selecione o *device* **EP2C35F672C6** e o ficheiro "*contador.sof*" que deverá usar para programar a FPGA. Clique em *"Hardware Setup"* e selecione *"Currently Selected Hardware->USB-Blaster"*. Finalmente, faça "***Start***". O ficheiro deverá ser transferido sem problemas.

Teste o *hardware*, verificando se tudo funciona como esperado!

Passos a controlar e perguntas a responder ao longo do trabalho

1. Desenhe o circuito, conforme indicado.

2. Programe e teste o circuito.

3. Como pode ver, números superiores a 9 não são corretamente mostrados. Por que é que isso acontece?

4. Elimine o componente 7447 e substitua-o pelo componente DEC_7SEG. Para usar esse componente terá de aprender a utilizar um BUS (conjunto de fios agrupados) **– chame o professor para lhe explicar como se usa esta opção quando chegar a esta fase**. Consulte também o *help* da aplicação! Teste o circuito mais uma vez.

5. ***Projete e altere*** o circuito para que seja feito um *reset* síncrono após o símbolo 9.

6. Teste o circuito mais uma vez.

7. ***Adicione*** 4 saídas ao circuito, que serão ligadas a quatro LEDs da placa. Essas saídas deverão ser ativadas a 0, correspondendo aos quatro *bits* à saída do contador.

8. Teste o circuito, vendo que o contador efetivamente conta corretamente em binário.

9. Elimine o botão de relógio. Substitua-o por um sinal de 1Hz (use o componente *clk_div* e ligue-o ao sinal de relógio da placa de 50MHz – consulte o manual da placa para descobrir o código de pino respetivo – para dividir a sua frequência), ligando-o à entrada CLK do 74163. Teste o circuito mais uma vez.

10. Por fim, altere o circuito para que agora seja feita uma contagem dos números ímpares até 15, isto é, 1 → 3 → 5 → 7 → 9 → B → D → F→ 1 → 3 → ...!

Nota importante: alguns dos componentes mencionados acima **apenas estarão disponíveis** se a pasta DE2 estiver correctamente instalada (ver ponto 1 da descrição para o circuito base).