ISEL
Departamento de Engenharia
Eletrónica e Telecomunicações
e de Computadores

Licenciatura em Engenharia Informática e de Computadores
e
Licenciatura em Engenharia Informática, Redes e Telecomunicações

Introdução às Tecnologias *TTL* e *FPGA* (*1º Laboratório*)

Lógica e Sistemas Digitais
2024 / 2025 inverno

9 de setembro de 2024

Objetivo

Este trabalho prático tem como principal objetivo a implementação de uma função lógica com as tecnologias *TTL* (*Transistor-Transistor Logic*) e FPGA (*Field-Programmable Gate Array*). Este trabalho não é contabilizado para a classificação prática.

PARTE I – Tecnologia TTL

1 Descrição e Análise da Função Lógica

Considere o logigrama da função lógica F(A, B, C) ilustrado na Figura 1.

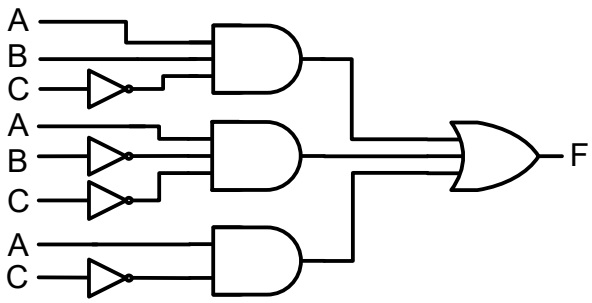


Figura 1 – Logigrama da função F

Considerando a função F, deverá realizar previamente as seguintes alíneas de preparação para o laboratório:

a) Determine a expressão lógica da função F(A, B, C);
b) Elabore a tabela de verdade da função F considerando a variável A como a de maior peso;
c) Simplifique algebricamente a função F e determine a tabela de verdade da função simplificada. Compare com a tabela de verdade da alínea b);

2 Montagem Laboratorial

Na aula laboratorial deverá realizar os seguintes passos:

1. Elabore o esquema elétrico da função determinada na alínea *2c)* utilizando portas lógicas *TTL*;
2. Monte o circuito projetado na *breadboard* tendo em atenção que todos os circuitos integrados têm de ser alimentados (0 e 5V);
3. Elabore uma tabela de verdade experimental em que, para cada combinação das variáveis de entrada, seja determinado o valor da função através da análise da montagem (valores experimentais). Compare com o resultado teórico (valores esperados).

PARTE II – Tecnologia FPGA

1 Fluxo de Desenvolvimento de Circuitos Digitais com VHDL e FPGA

Para este e os demais trabalhos da unidade curricular, dada a especificação do problema, será seguido um fluxo de projeto com os passos seguintes:

1. Descrição do circuito em *VHDL*;
2. Síntese do circuito;
3. Simulação do circuito;
4. Implementação do circuito na tecnologia *FPGA*;
5. Implementação do circuito na placa:
6. Geração do ficheiro de programação;
7. Programação da *FPGA*;
8. Validação do circuito.

Neste documento é apresentado o desenvolvimento de um pequeno projeto para ilustrar os passos deste fluxo.

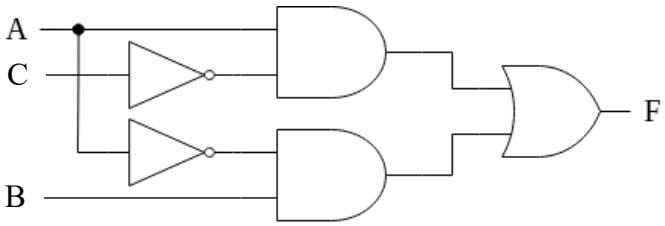
2 Realização do Trabalho

2.1 Criação do Projeto

Comece por criar um projeto do seguinte modo:

1. Escolha **File → New Project Wizard** e prima **Next**;
2. Especifique a diretoria do projeto, o nome do projeto e o nome da entidade de topo (*lab2*), que pode ser o mesmo do nome do projeto;
    a. **Observação**: em alguns sistemas, o *Quartus* coloca, por omissão, sua diretória de instalação como a diretoria sugerida para o projeto. Deve-se evitar selecionar esta diretoria para o projeto, pois pode resultar em problemas de permissão que impedem a síntese do projeto.
3. Em seguida escolha a opção para criar um projeto vazio (*Empty project*).
4. Na janela seguinte, a ferramenta permite adicionar ao projeto ficheiros já existentes. Neste tutorial, não escolha qualquer ficheiro e avance.
5. Na janela seguinte, deverá escolher o dispositivo em que o circuito lógico irá ser implementado e testado. Escolha o *tab* "Board";
6. Escolha a placa de laboratório: MAX10 DE10-Lite (deverá selecionar a família MAX 10). **Desmarque a opção "*Create-top Level Design File*"**;
7. Na janela seguinte, poderá selecionar outras ferramentas externas ao *Quartus*. Não o vamos realizar nesta fase, pelo que deve avançar para a próxima janela;
8. Termine.

Neste tutorial, vamos considerar o seguinte circuito lógico:



A descrição em *VHDL* do circuito é a seguinte:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY lab2 IS
  PORT(A, B, C : IN STD_LOGIC;
  F : OUT STD_LOGIC);
END lab2;

ARCHITECTURE logicFunction OF lab2 IS
BEGIN
F <= (A AND NOT C) OR (NOT A AND B);
END LogicFunction;
```

Escreva o código *VHDL* num editor qualquer ou usando o editor do *Quartus*.

Consideremos a utilização do editor do *Quartus*:

1. Escolha **File → New,** selecione *VHDL* File e avance;
2. Este procedimento abre o editor de texto;
3. Escreva a sua descrição *VHDL* do circuito;
4. Guarde o ficheiro (**File → Save)**. Escolha o tipo de ficheiro *VHDL* e a opção de adicionar o ficheiro ao projeto. O nome do ficheiro é *lab2.vhd*. Note que o nome do ficheiro corresponde ao nome da entidade no código *VHDL*.

2.2 Compilação do Circuito

O circuito descrito em *VHDL* é analisado, sintetizado e é gerada uma implementação do circuito na tecnologia da *FPGA*. Para tal, deverá executar **Processing → Start Compilation**.

No fim da compilação, um relatório é gerado e automaticamente visualizado. O relatório pode também ser acedido em qualquer altura selecionando **Processing → Compilation Report**.

No caso do código *VHDL* ter erros, estes são listados na janela de mensagens e podem ser acedidos diretamente a partir desta janela.
Pode visualizar o diagrama lógico do circuito selecionando **Tools → Netlist Viewers → RTL Viewer**.

2.3 Simulação do Circuito

À medida que os circuitos desenvolvidos se tornam mais complexos, aumenta a probabilidade de cometermos erros – seja na etapa de desenho do circuito ou na sua descrição em *VHDL*. Detetar erros e identificar as causas diretamente na *FPGA* é uma tarefa difícil, já que temos acesso apenas às entradas e saídas do circuito, mas não aos pontos intermédios.
Por este motivo, antes de realizar a programação da FPGA, é fundamental simular o circuito e realizar testes de forma a depurar os erros mais facilmente. Para este propósito, o *Quartus* possui integração com um simulador chamado *Altera-Modelsim*.

Uma vez compilado o projeto sem erros, pode-se utilizá-lo da seguinte forma:

1. Configure o *Quartus* para utilizar o simulador *Altera-Modelsim*:
    a. Selecione **Assignments → Settings : EDA Tool Settings → Simulation**;
    b. Verifique que "Modelsim-Altera" está selecionado e que a linguagem é *VHDL*. Aplique as alterações;
        i. Observe que poderíamos ter feito esta configuração também no passo 7 da criação do projeto.
    c. Em seguida, selecione **Tools → Options → General → EDA Tools**;
    d. Coloque o caminho para o simulador em "Modelsim-Altera": "…\Quartus20.1\modelsim_ase\win32aloem"
2. Selecione **Tools → Simulation Tool → RTL Simulation** para abrir o editor de simulação;
3. Execute o simulador. Existem várias formas. Na janela **library**, expanda a diretoria *work*. Em cima da entidade *lab2*, carregue no lado direito do rato e escolha **simulate**. Ou pode simplesmente fazer duplo *click* em *lab2*;
4. Na janela de objetos, selecione os sinais que quer visualizar na simulação, carregue no lado direito do rato e selecione **add wave** (ou CTRL+W). Outra possibilidade é simplesmente arrastar os objetos para a janela de visualização do diagrama temporal.
    a. Note que *objetos* denotam tanto sinais de entrada e saída, quanto ligações internas do circuito. Logo, é possível observar como qualquer um destes elementos se comporta durante o funcionamento do circuito.
5. Altere os valores das portas de entrada do circuito para o caso que se deseja testar. Como exemplo, simularemos A=1, B=1 e C=0. Para isso, na janela **wave**, carregue no botão direito do rato sobre o sinal da entrada A ("/lab2/A") e selecione **Force…**. Na nova janela exibida, coloque o valor 1 no campo **Value** e carregue em **Ok**. Repita o processo para as restantes entradas.

6. Para executar a simulação, selecione o ícone ao lado do tempo de simulação no menu superior do *ModelSim*. Isto simula o circuito durante o tempo indicado e as ondas relativas à evolução dos vários sinais selecionados são exibidas na janela **wave**. Observe se a saída F está de acordo com o esperado.
7. Repita os passos 5 e 6 para as restantes combinações de valores de A, B e C.

Caso altere algum dos ficheiros do projeto, deve voltar a compilar os circuitos e voltar a executar a simulação **Simulate → Restart**.

2.4 Atribuição de Pinos

Durante a compilação, o *Quartus* realizou uma atribuição aleatória dos sinais de entrada e de saída da entidade de topo (no caso deste laboratório, a única do projeto) aos pinos da *FPGA*. Contudo, a *DE10-Lite* já tem ligações pré-estabelecidas entre os componentes da placa (e.g., LEDs, *switches*) e os pinos da *FPGA*. Assim, se quisermos controlar as entradas A, B e C por *switches* específicos e observar a saída F em um *LED* em particular, é necessário atribuir estes sinais aos pinos dos componentes desejados. É possível consultar a ligação entre os componentes da placa *DE10-Lite* e os pinos da *FPGA* através do manual da placa disponível no *Moodle* da unidade curricular. Também no *Moodle* é possível encontrar um ficheiro de exemplo de atribuição de pinos denominado `de10Lite.qsf`.

A atribuição pode ser realizada pino a pino através da opção **Assignments → Assignment Editor**. Contudo, para muitos pinos, é um processo moroso. Em alternativa, podemos importar a atribuição de pinos a partir de um ficheiro que tem o formato QSF (*Quartus Settings File*).

Para importar um ficheiro `.qsf`, deve selecionar **Assignments → Import Assignments**. O comando abre uma janela para que possa escolher o ficheiro pretendido. Para o exemplo do tutorial, crie e grave um ficheiro **com extensão `.qsf`** com as seguintes atribuições e carregue-o:

```
set_global_assignment -name BOARD "MAX 10 DE10 - Lite"
set_global_assignment -name DEVICE 10M50DAF484C6GES
set_global_assignment -name FAMILY "MAX 10"

set_location_assignment PIN_A8 -to F
set_location_assignment PIN_C10 -to A
set_location_assignment PIN_C11 -to B
set_location_assignment PIN_D12 -to C
```

Observe que **a extensão do ficheiro de atribuição de pinos é essencial**. Ao tentar carregar uma atribuição de pinos a partir de um ficheiro com outra extensão qualquer, a interface do *Quartus* não gera nenhuma mensagem de erro. Contudo, a atribuição não é efetivamente processada.

Para confirmar que os pinos estão atribuídos utilize o **Assignments → Assignment Editor**.

Toda vez que a atribuição de pinos é alterada, é necessário **compilar novamente o projeto**.

2.5 Programação da *FPGA*

A *FPGA* tem de ser programada para implementar o circuito projetado. O ficheiro de configuração é gerado pelo *Quartus*. Para programar a *FPGA*, selecione **Tools -> Programmer**. Na janela de configuração, deverá ter *USB-Blaster* no *Hardware Setup* e *JTAG* no *Mode*. De seguida, pressione ***Start***.

2.6 Validação do circuito

Valide experimentalmente na placa *DE10-Lite* atribuindo todas as combinações das entradas e observando o resultado apresentado na saída do sistema, de forma análoga ao procedimento realizado no simulador, comprovando desta forma o correto funcionamento do circuito implementado.