

# TRABALHO DE LABORATÓRIO II

## CIRCUITOS COMBINATÓRIOS TÍPICOS

VERSÃO 1.1

### 1. INTRODUÇÃO

Pretende-se com este trabalho que os alunos se familiarizem com a utilização de ferramentas de simulação e prototipagem de circuitos digitais. Para o efeito, será utilizada a ferramenta *Vivado ML Edition* da *Xilinx* para implementar um circuito combinatório típico constituído por decodificadores e multiplexers. Este trabalho é considerado para **avaliação de conhecimentos**.

#### **Notas preliminares importantes:**

- Este trabalho deverá ser preparado atempadamente e separadamente por cada aluno. Uns dias antes da aula de laboratório, os alunos de cada grupo devem reunir-se, conferir as suas respostas, **preencher a folha de respostas** disponível na página da disciplina e **implementar o projeto em Vivado (uma única folha e um único projeto por grupo)**.
- **A folha de respostas preparada, não sendo avaliada, será verificada pelo docente no início da aula de laboratório. A apresentação da preparação de casa no início da aula é condição necessária para a realização do laboratório.**
- É fundamental que os alunos cheguem a horas ao laboratório.
- No início de cada sessão, será fornecida informação adicional sobre a sessão de laboratório que permitirá preencher a segunda folha de perguntas/respostas que deverá trazer impressa de casa juntamente com a primeira, de onde resultará o circuito **a ser descrito em VHDL**. É fundamental que ambos os alunos preparem devidamente o trabalho, i.e., que ambos os alunos tenham um bom conhecimento da utilização da ferramenta Vivado, da descrição dos circuitos em VHDL.
- Os últimos 10 minutos são reservados à escrita das conclusões. Ambas as folhas de resposta devem ser fotografadas e submetidas via fénix, num zip, juntamente com o projeto do Vivado desenvolvido na aula.

## 2. PROJETO DO CIRCUITO COMBINATÓRIO

O centro logístico da cadeia de supermercados *Trigo Doce* pretende desenvolver um dispositivo de separação de produtos alimentares baseado da identificação presente nos códigos de barras das respetivas embalagens. Para o efeito, instalou um leitor e classificador automático junto a um ponto de bifurcação do tapete rolante que transporta os mesmos. Este separador é capaz de distinguir 16 produtos diferentes e separá-los em diferentes categorias. A identificação de cada produto é codificada em binário utilizando 4 bits ( $A_3, A_2, A_1, A_0$ ). Desta forma, o produto com o código de barras Nº 13 será identificado em binário com a representação  $A_3, A_2, A_1, A_0 = 1101_2$ .

Por ação de uma palavra de comando constituída por 2 bits ( $P_1, P_0$ ), o operador poderá optar por separar todos os produtos alimentares de uma determinada categoria (desviando-os para um ramal secundário do tapete rolante), mantendo todos os outros no ramal principal do tapete. Os produtos não alimentares deverão permanecer sempre no ramal principal do tapete. O resultado da classificação é sinalizado na saída S de 1-bit.

Assume-se que este centro processa as seguintes categorias de produtos alimentares:

- $C_0$  – Frutas/Legumes ( $P_1, P_0 = 00$ )
- $C_1$  - Bebidas ( $P_1, P_0 = 01$ )
- $C_2$  - Lacticínios ( $P_1, P_0 = 10$ )
- $C_3$  – Cereais e Massas/Pastas ( $P_1, P_0 = 11$ )
- NA – Não Alimentar

A tabela seguinte associa cada produto (já pré-classificado na sua categoria) com o código numérico correspondente.

**Tabela 1. Classificação dos produtos.**

Código	Categoria	Descrição
3	$C_0$	Maçã Golden
5	$C_0$	Abacaxi
11	$C_0$	Bananas
9	$C_1$	Sumo de Laranja
7	$C_1$	Coca Cola
13	$C_2$	Leite Meio Gordo
15	$C_3$	Arroz
1	$C_3$	Massa Napolitana
8	NA	Copos
4	NA	Pratos Descartáveis
6	NA	Vassoura
14	NA	Sacos de Lixo
12	NA	Esfregão
10	NA	Lâmpada 60W
2	NA	Balde
0	NA	Escova

## 3. PROCEDIMENTO PARA IMPLEMENTAÇÃO DO CIRCUITO

- 3.1. Ordene a Tabela 1 (por ordem crescente do código do produto) e represente a tabela de verdade das funções Booleanas  $C_0(A_3, A_2, A_1, A_0)$ ,  $C_1(A_3, A_2, A_1, A_0)$ ,  $C_2(A_3, A_2, A_1, A_0)$  e  $C_3(A_3, A_2, A_1, A_0)$ , que realizam a associação entre cada produto e a categoria correspondente. Exemplo: ao passar um pacote de leite pelo leitor, a função  $C_2 = 1$ , enquanto  $C_0 = C_1 = C_3 = 0$ .

**Responda à pergunta 1 na folha de respostas.**

- 3.2. Projete um circuito combinatório que permita ao operador separar todos os produtos de uma determinada classe ( $C_0$ ,  $C_1$ ,  $C_2$  ou  $C_3$ ), indicada por ação da palavra de comando ( $P_1P_0$ ). O circuito deverá ter como variáveis de entrada os bits  $A_3, A_2, A_1, A_0$  (que identificam o produto) e os dois bits  $P_1P_0$  (que identificam a classe de produto que se pretende selecionar). A saída do circuito é constituída por um único bit ( $S$ ), que deverá corresponder ao valor da função Booleana  $C_0$ ,  $C_1$ ,  $C_2$  ou  $C_3$  que está a ser selecionada pelos bits  $P_1P_0$ .  
**NOTA:** utilize descodificadores 3:8 com enable (saídas negadas), um multiplexer 4:1 com enable e o mínimo de lógica adicional (é permitido usar apenas as portas lógicas NAND3).

**Responda à pergunta 2 na folha de respostas.**

- 3.3. Descarregue o ficheiro `lab2.zip` disponível na página da cadeira. Guarde-o no seu ambiente de trabalho e descompacte-o para uma outra pasta de nome `lab2`. Abra o Vivado, escolha a opção *open project* e selecione o ficheiro `lab2.xpr` na pasta `lab2`. Após concluir esses passos, o painel P2 no Vivado deverá ter um aspeto apresentado na Figura 1.

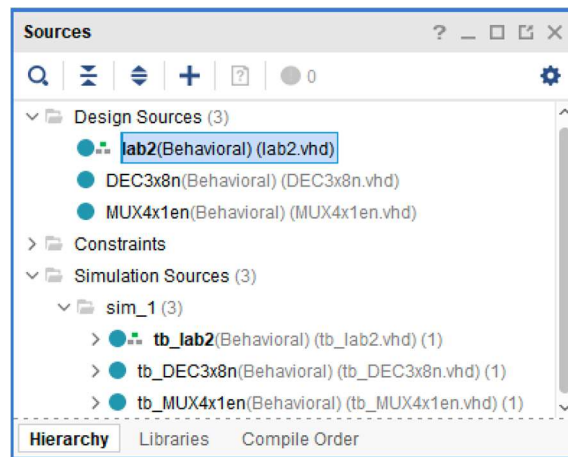


Figura 1. Project Sources no Vivado

- A pasta *Design Sources* contém os ficheiros “.vhd” que representam a descrição (em VHDL) dos circuitos para utilizar neste trabalho, nomeadamente: `DEC3x8n` (descodificador 3:8 com enable, saídas negadas), `MUX4x1en` (multiplexer 4:1 com enable) e `lab2` (o circuito para implementar neste trabalho).
  - A pasta *simulation\_sources* contém os ficheiros “.vhd” utilizados para a simulação (em VHDL) dos circuitos, i.e., *Simulation Sources* ou *Test Benches* (*tb*). Estes ficheiros incluem uma sequência de teste para simular (verificar) o comportamento do circuito.
  - No caso de surgir uma caixa de diálogo perguntando se pretende atualizar o projeto para a sua versão do Vivado responda sim.
- 3.4. Verifique se o componente `lab2` (*Design Sources*) fica configurado como sendo o módulo de topo. Abra a descrição em VHDL deste componente.
- 3.5. Complete a implementação do circuito `lab2` de modo a obter o funcionamento descrito na Secção 2. Quando terminar a implementação, faça a síntese do circuito e corrija **todos** os erros. **Nota importante:** Não pode alterar a definição do componente (“entity” em VHDL)!
- 3.6. Verifique se o *TestBench* `tb_lab2` (*Simulation Sources*) fica configurado como sendo o módulo de topo. Abra a sua descrição em VHDL e analise o conteúdo deste ficheiro.
- 3.7. Complete o *TestBench* de modo a que todas as combinações da entrada  $A$  são testadas (ver linha 77 no VHDL). Percorra os valores de  $A$  pela mesma ordem que apresentado na Tabela 1. Faça a simulação e verifique o correto funcionamento do circuito.

**Verifique a sua resposta à pergunta 2 e responda à pergunta 3 na folha de respostas.**

## 4. SUBMISSÃO DO TRABALHO NO FENIX

A submissão do trabalho no Fenix é condição obrigatória para o trabalho ser avaliado. A submissão consiste num único ficheiro *.zip* com o nome *grupo<número do grupo>.zip*. Este ficheiro deverá conter os seguintes elementos:

1. Folha de respostas preenchida na aula;
2. Ficheiro de arquivo do projeto Vivado (também em formato *.zip*).

O arquivamento do projeto Vivado tem obrigatoriamente de ser feito a partir do próprio Vivado, selecionando a opção **File > Project > Archive Project**, e depois ativando as opções tal como se mostra na Figura 2.

Só serão aceites os projetos cujos ficheiros VHDL tiverem um cabeçalho com o seguinte formato, em que os elementos em *itálico* são substituídos pela respetiva informação:

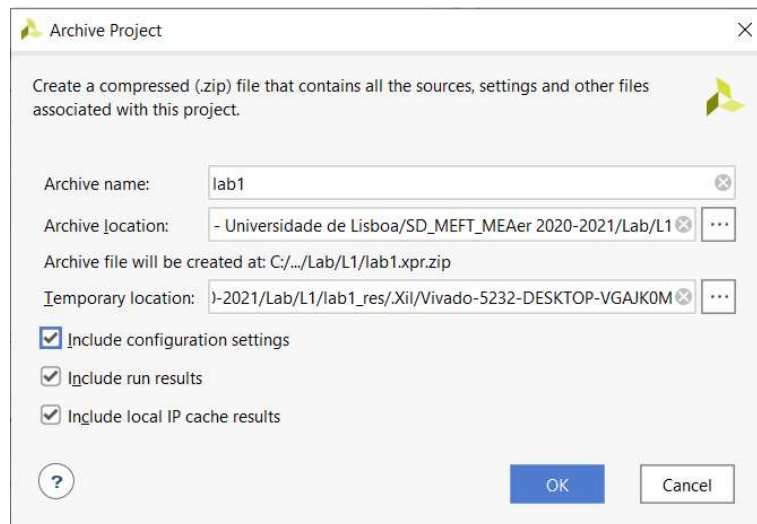
-----

-- Numero do Grupo: *<número do grupo>*

-- *<nome do 1º elemento do grupo>* (*<número do 1º elemento do grupo>*)

-- *<nome do 2º elemento do grupo>* (*<número do 2º elemento do grupo>*)

-----



**Figura 2: Arquivamento de projecto no Vivado.**