

# TRABALHO DE LABORATÓRIO III

## CIRCUITOS SEQUENCIAIS

VERSÃO 1.0

### 1. INTRODUÇÃO

Pretende-se com este trabalho que os alunos se familiarizem com os elementos básicos de memória (flip-flops) e com a utilização de ferramentas de simulação e prototipagem de circuitos digitais. Para o efeito, será utilizada a ferramenta *Vivado WebPack* da *Xilinx* (versão 2018.2) para simular e implementar um circuito sequencial simples. Este trabalho é considerado para **avaliação de conhecimentos**.

Durante a realização deste trabalho, poderá consultar os seguintes documentos (disponíveis na página da cadeira):

- [Guia de Instalação do Vivado](#)
- [Guia do Utilizador do Vivado](#)
- [Guia da Placa Basys3](#)
- [Introdução à linguagem VHDL](#)

Este enunciado inclui notas adicionais na margem esquerda da página (do tipo **G(X.X)**). Essas notas referem-se ao(s) passo(s) **X.X** do **Guia de Utilização do Vivado Design Suite WebPack**, que deve ser consultado durante o desenvolvimento deste trabalho.

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

- Este enunciado deverá ser preparado atempadamente por cada aluno **separadamente**. 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 cadeira** (uma única folha por grupo), e **implementar o projeto no Vivado**. Caso surjam dúvidas, devem recorrer aos horários de dúvidas.
- As folhas de respostas de casa **não serão avaliadas**. No entanto, poderá ser necessário mostrar ao docente a preparação de casa durante a aula, sendo **indispensável** uma **preparação** cuidada do trabalho. As **simulações** devem ter sido efetuadas cuidadosamente e os seus valores **confirmados** com o **esperado teoricamente**. Caso surjam dúvidas durante a preparação, os alunos devem recorrer aos horários de dúvidas.
- É fundamental que os alunos cheguem a horas ao laboratório.
- No início da sessão de laboratório, será distribuída a cada grupo uma nova folha de perguntas/respostas, com perguntas ligeiramente diferentes das preparadas em casa, de onde resultará o circuito **a ser descrito em VHDL e a ser implementado na placa de desenvolvimento**. 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 e da implementação na placa.
- Os últimos 10 minutos são reservados à escrita das conclusões, sendo a folha de perguntas/respostas entregue no final da aula.

## 2. ANÁLISE DE UM CIRCUITO SEQUENCIAL BÁSICO

O esquema da Figura 1 implementa uma máquina de estados constituída por 3 Flip-Flops (FF's) tipo JK, D e T (todos com entradas de *Set* e *Reset* síncronas) e lógica combinatória adicional. A descrição VHDL deste circuito, assim como dos flip-flops, está no ficheiro `sequencial.vhd`, incluído no ficheiro `lab3.zip`, disponível na página da disciplina.

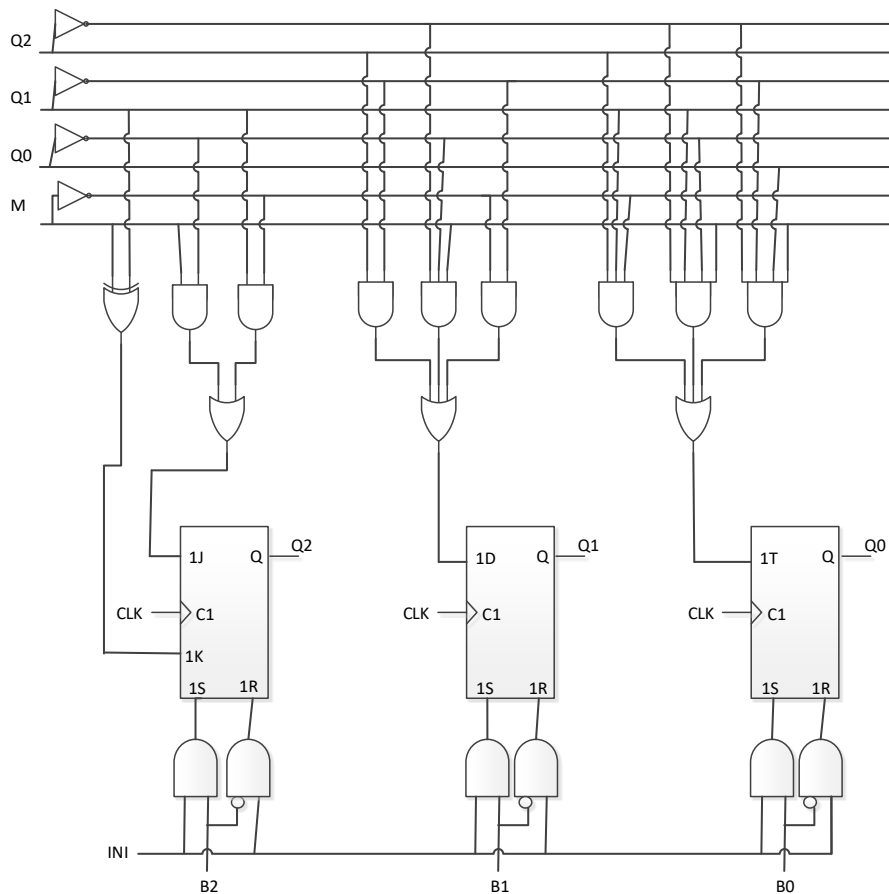


Figura 1 - Circuito sequencial básico.

1. Considere a entrada  $B = \langle b_2, b_1, b_0 \rangle$ , que corresponde ao resto da divisão por 7, da soma do dígito menos significativo do número de aluno com menor valor (quando este número está representado em base 10) com o dia da semana do turno de laboratório e somando 1 ao valor obtido. Por exemplo, para um grupo formado pelos elementos 76146 e 77188, o qual realiza laboratório à terça-feira, a constante  $B$  é  $(6+3 \bmod 7) + 1 = 3$ .

**Responda às perguntas 1, 2 e 3 na folha de respostas.**

2. Confirme o funcionamento do circuito através de simulação no Vivado.  
 Descarregue o ficheiro `lab3.zip` disponível na página da cadeira. Guarde-o no seu ambiente de trabalho e descompacte-o para uma outra pasta de nome `lab3`. A pasta `lab3` tem 2 subdiretorias: `design_sources`, e `simulation_sources`.
  - A pasta `design_sources` contém os ficheiros “.vhd” que representam a descrição (em VHDL) dos vários circuitos utilizados para implementar o circuito `lab3`. Os *Design*

*Sources* fornecidos contêm a definição do circuito no nível do componente (*entity*), bem como a sua implementação detalhada (*architecture*). Apenas os ficheiros *lab3.vhd* e *sequencial2.vhd* devem ser modificados.

- A pasta *simulation\_sources* contém os ficheiros “.vhd” utilizados para a simulação (em VHDL) de **todos** os circuitos, i.e., *Simulation Sources* ou *Test Benches* (*tb*). As *Simulation Sources* fornecidas incluem uma sequência de teste para simular (verificar) o comportamento do circuito.

G(1.1 até 1.4)

2.1. Inicie o ambiente de projeto *Vivado*. Crie um projeto novo com o nome **lab3** e faça a especificação do dispositivo, i.e., *Digilent Basys 3* baseada na FPGA da família Artix-7 com a referência *XC7A35TCPG236-1*.

G(2.1.a, 2.2-2.5)

2.2. Inclua os seguintes ficheiros de descrição do circuito (*Design Sources*): *ff\_d*, *ff\_t*, *ff\_jk* e *sequencial* (navegando até à pasta *design\_sources*). Expandir o circuito *sequencial* clicando no símbolo > (à esquerda) para visualizar todas as instâncias dos flip-flops que o circuito engloba.

G(3.1.a, 2.4,  
3.2-3.8)

2.3. Inclua o ficheiro *tb\_sequencial* para efetuar a simulação do circuito (*Simulation Source*), navegando até à pasta *simulation\_sources*. Certifique-se que o ficheiro é o módulo de topo. Abra-o, inspecione e faça a simulação e verifique se corresponde à sequência calculada na pergunta 3 da folha de respostas

### 3. PROJETO DE UM CIRCUITO SEQUENCIAL BÁSICO

G(2.1.a, 2.2-2.5)

3.1. Inclua o *Design Source* correspondente aos componentes *sequencial2*, *decoder*, *decoder\_neg*. Abra a descrição em VHDL deste componente e analise as suas entradas, saídas, e componentes internos. Nota: a descrição do circuito *sequencial2* fornecida não está completa.

3.2. Com base nos circuitos fornecidos (*sequencial*, *decoder*, *decoder\_neg*) e no mínimo de **componentes combinatórios** adicionais, dimensione o circuito *sequencial2* para funcionar como um contador de 4 bits (saídas S(3:0)) que repita a sequência de 8 números 0->1->6->5->A->6->E->F->0->etc.

Tenha em consideração o seguinte:

- O circuito *sequencial2* tem como entradas apenas os sinais de inicialização INI e CLK.
- Tenha em atenção que a entrada M do circuito *sequencial* está ligada a '0'.
- Tenha em atenção que o valor inicial da contagem (i.e., o valor S = "0000") deverá surgir na saída quando o estado do circuito *sequencial* corresponder ao valor B calculado na alínea anterior.
- Não pode utilizar elementos de memória (flip-flops) adicionais.
- As funções combinatórias S(1) e S(0) devem ser implementadas apenas em função das saídas do decodificador 3:8 DEC e o mínimo de lógica adicional.
- As funções combinatórias S(3) e S(2) devem ser implementadas apenas em função das saídas do decodificador 3:8 com saídas ativas a zero DEC\_NEG e o mínimo de lógica adicional..
- Os decodificadores têm como entrada, Q, a saída do circuito *sequencial*.

**Responda à pergunta 4 na folha de respostas.**

G(3.1.a, 2.4,  
3.2-3.8)

3.3. Inclua o ficheiro `tb_sequencial2` para efetuar a simulação do circuito (*Simulation Source*). Certifique-se que o ficheiro é o módulo de topo. Faça a simulação. Na consola TCL pode visualizar o resultado da verificação automática.

G(2.1.a, 2.2-  
2.5)

3.4. Inclua o *Design Source* correspondente ao componente `lab3` e verifique se este fica configurado como sendo o módulo de topo. Abra a descrição em VHDL deste componente e analise as suas entradas, saídas, e componentes internos. Nota: a descrição fornecida não está completa.

3.5. Com base no circuito fornecido e o mínimo de **componentes combinatórios** adicionais, dimensione agora um novo circuito sequencial cuja saída `Y(3:0)` repete a sequência de 10 números: 5->D->0->E->0->F->1->C->4->E->5 etc. Utilize para tal um elemento de memória adicional do tipo **flip-flop T**, o circuito `sequencial2` e os decodificadores 3:8 DEC com saída ativa a '1', e DEC\_NEG com saída ativa a '0'.

Tenha em consideração o seguinte:

- Os decodificadores do circuito `lab3` têm como entrada `Q(2:0)`, que é a saída do circuito `sequencial` (incluído dentro do circuito `sequencial2`). A saída do **flip-flop T** está ligada ao *enable* dos decodificadores.
- O decodificador DEC está ativo quando a saída do **flip-flop T** é '1', O decodificador DEC\_NEG está ativo quando a saída do **flip-flop T** é '0'.
- Implemente os sinais `fft_t`, `fft_ini` e `seq2_ini`, entradas T e INI do FF e o INI do circuito `sequencial2`, respetivamente, como funções apenas das saídas dos decodificadores e da entrada INI.
- As funções combinatórias `Y(3)`, `Y(2)`, `Y(1)` e `Y(0)` devem ser implementadas apenas em função das saídas dos decodificadores.

**Sugestão:** Comece por determinar as entradas T e INI do **flip-flop T** (respetivamente `fft_t` e `fft_ini`) e a entrada INI do circuito `sequencial2`, de forma a criar uma sequência de 10 valores que deve repetir-se sucessivamente. Por exemplo, pode gerar a sequência [`Q(000)` `Q(001)` ... `Q(111)` `Q(000)` `Q(001)`], ou seja, a sequência Q seguida dos dois primeiros valores de Q. Estando o decodificador DEC ativo nos primeiros 8 valores e o decodificador DEC\_NEG ativo nos 2 últimos. Implemente Y tendo em conta os 10 valores da sequência e o decodificador que está ativo.

### Responda à pergunta 5 na folha de respostas.

3.6. Verifique o seu correto funcionamento utilizando o circuito de teste fornecido (`tb_lab3`).