

TRABALHO DE LABORATÓRIO II

CIRCUITOS COMBINATÓRIOS TÍPICOS

VERSÃO 1.0

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 WebPack* da *Xilinx* (versão 2018.2) para simular um circuito constituído por circuitos combinatórios típicos. Este trabalho é considerado para **avaliação de conhecimentos**.

Durante a realização deste trabalho, poderá é fundamental 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. DESCRIÇÃO DO CIRCUITO

Desenvolva um circuito que, de acordo com a Figura 1, implementa uma unidade combinatória que executa uma de duas funções aritméticas sobre as variáveis de entrada $A = \langle a_4, a_3, a_2, a_1, a_0 \rangle$ e $B = \langle b_4, b_3, b_2, b_1, b_0 \rangle$, de 5 bit, resultando na saída $S = \langle s_7, s_6, s_5, s_4, s_3, s_2, s_1, s_0 \rangle$ com 8 bit. A função a calcular depende de um bit de seleção, I . As funções a implementar encontram-se listadas na Tabela 1.

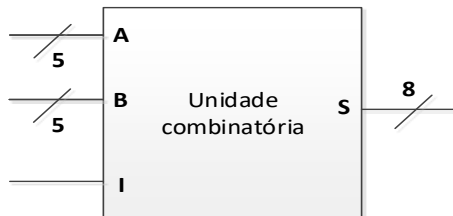


Tabela 1: Funções a implementar.

I	Função
0	$S = (2A+B)/4$
1	$S = (3A - 2B)/8$

Figura 1: Esquema da unidade combinatória.

As operações são realizados **sem nenhuma simplificação algébrica** e assumindo formato de **complemento para 2** de 8 bits. As situações de *overflow* devem ser **ignoradas** (não adicione lógica adicional para tratar esses casos). Tenha em atenção que devido às partes decimais perdidas os resultados obtidos podem diferir do valor arredondado obtido da operação aritmética com casas decimais.

Notas importantes:

- As operações internas são todas executadas com 8 bits (complemento para dois), pelo que os sinais A e B devem começar por sofrer uma extensão de sinal para 8 bits;
- As multiplicações e divisões inteiras por potências de 2 devem ser executadas através de operações de deslocamento (*shift left aritmetico* (SHLA) e *shift right aritmetico* (SHRA), respetivamente), devendo preservar o bit de sinal. Por exemplo, o SHLA de um vector $v(3:0)$ é $[v(2:0); 0]$ e o SHRA $[v(3); v(3:1)]$;
- As somas são feitas por somadores de 8 bits;

Assim, por exemplo, a operação $3xA$ é feita pela soma $A + 2xA$ através de um somador de 8 bits, e $(2xA)$ corresponde ao SHLA de A. Estas operações são feitas depois de se fazer a extensão do sinal de A para 8 bits.

3. PROJETO DA UNIDADE COMBINATÓRIA

Para o Projeto do circuito que concretiza a funcionalidade especificada, utilize sempre o mínimo de componentes, considerando **obrigatoriamente** apenas somadores de 8 bits, e, opcionalmente, multiplexers, descodificadores e o mínimo de lógica adicional (portas XOR, NAND2, NAND3, NOR2, NOR3 ou NOT).


Na folha de respostas de casa:

1. Apresente o logograma do circuito que propõe para implementar a unidade combinatória descrita na secção 2.
2. Descreva o funcionamento da solução proposta.

Responda às perguntas 1 e 2 na folha de respostas.

3. Apresente os cálculos necessários para obter S, com os passos intermédios (extensões de sinal e deslocamentos), para todas as combinações das entradas $I=\{0;1\}$, $A = \{-14\}$ e $B = \{-13;11\}$.

Responda à pergunta 3 na folha de respostas.

4. Confirme que o circuito implementado está a funcionar corretamente através de simulação no Vivado.
 - 4.1. 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**. A pasta **lab2** 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 **lab2**. Os *Design Sources* fornecidos contêm a definição do circuito no nível do componente (*entity*), bem como a sua implementação detalhada (*architecture*). Apenas os ficheiros *lab2.vhd* e *somador8.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.
 - 4.2. Inicie o ambiente de projeto *Vivado*. Crie um projeto novo com o nome **lab2_numgrupo** (e.g., **lab2_21**, para o grupo 21) e faça a especificação do dispositivo, i.e., *Digilent Basys 3* baseada na FPGA da família Artix-7 com a referência *XC7A35TCPG236-1*.
 - 4.3. Inclua os seguintes ficheiros de descrição do circuito (*Design Sources*): *full_adder* e *somador4* (navegando até à pasta **design_sources** - reveja o ponto 4.1). Expanda o circuito *somador4* clicando no símbolo  (à esquerda) para visualizar todas as instâncias do componente *full_adder* que o *somador4* engloba. Defina o *somador4* como módulo de topo e abra-o. Como se pode observar, o somador de 4 bits é implementado utilizando 4 *full adders*, tal como descrito na [aula teórica](#).
 - 4.4. Inclua o *Design Source* correspondente ao componente *somador8* 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 do somador 8 fornecida não está completa.

Responda à pergunta 4 na folha de respostas.
 - 4.5. Complete a descrição do componente *somador8* de forma a obter o correto funcionamento do somador de 8 bit.
 - 4.6. Inclua o ficheiro *tb_somador8* para efetuar a simulação do circuito (*Simulation Source*), navegando até à pasta **simulation_sources** (reveja o ponto 4.1). Certifique-se que o ficheiro é o módulo de topo. Abra-o, inspecione e complete o seu conteúdo e faça a simulação. Não se esqueça de modificar a representação dos sinais (“Radix”) para *Signed Decimal*.
 - 4.7. Inclua o *Design Source* correspondente ao componente *lab2* e verifique se este fica configurado como sendo o módulo de topo. Abra a descrição em VHDL do componente e complete de forma a obter o circuito projetado no ponto 1.

Quando terminar a implementação, faça a síntese do circuito e corrija **todos** os erros.

Nota: O circuito *lab2* fornecido inclui todos os componentes a utilizar (somador 8) completamente definidos e exemplos de instanciação. Tenha em atenção que os sinais que definem as entradas dos somadores estão definidos (*som1_p*, *som1_q*, *som1_ci*, *som2_p*, *som2_q*, *som2_ci*), bem como as saídas (*som1_s*, *som2_s*), as saídas *c_out* não estão ligadas. Existe ainda em comentário um exemplo do código VHDL para a realização de um multiplexer 2 para 1 com 8 bits .
 - 4.8. Inclua o *Simulation Source* *tb_lab2_short*. Verifique se este fica configurado como sendo o módulo de topo e abra a descrição da simulação em VHDL (*test bench*).

G(3.1.a,2.4,3.2,3.3)

Nota: Este *Simulation Source* está completamente preenchido, pelo que não é necessário introduzir nenhuma modificação. Por conseguinte, este ficheiro pode ser utilizado durante a implementação, para verificar o correto funcionamento do circuito `lab2`.

G(3.4-3.8)

4.9. Faça a simulação do circuito `lab2` **implementado** utilizando o *Simulation Source* `tb_lab2_short`.

G(3.1.a,2.4,3.2,3.3)

4.10. Inclua o *Simulation Source* `tb_lab2_all`. Verifique se este fica configurado como sendo o módulo de topo e abra a descrição da simulação em VHDL (*test bench*). Este *Simulation Source* está completamente preenchido e verifica todas as combinações possíveis de I, A e B. Por conseguinte, este ficheiro pode ser utilizado durante a implementação, para verificar o correto funcionamento do circuito `lab2`.

Responda à Pergunta 5 na folha de respostas.

4. TESTE DO CIRCUITO NA PLACA DE PROTOTIPAGEM

Nota importante: Antes de iniciar o teste do circuito é **fundamental consultar (em casa)** o [Guia de Implementação de Circuitos na Placa de Desenvolvimento](#) (Digilent Basys 3), disponível na página da cadeira.

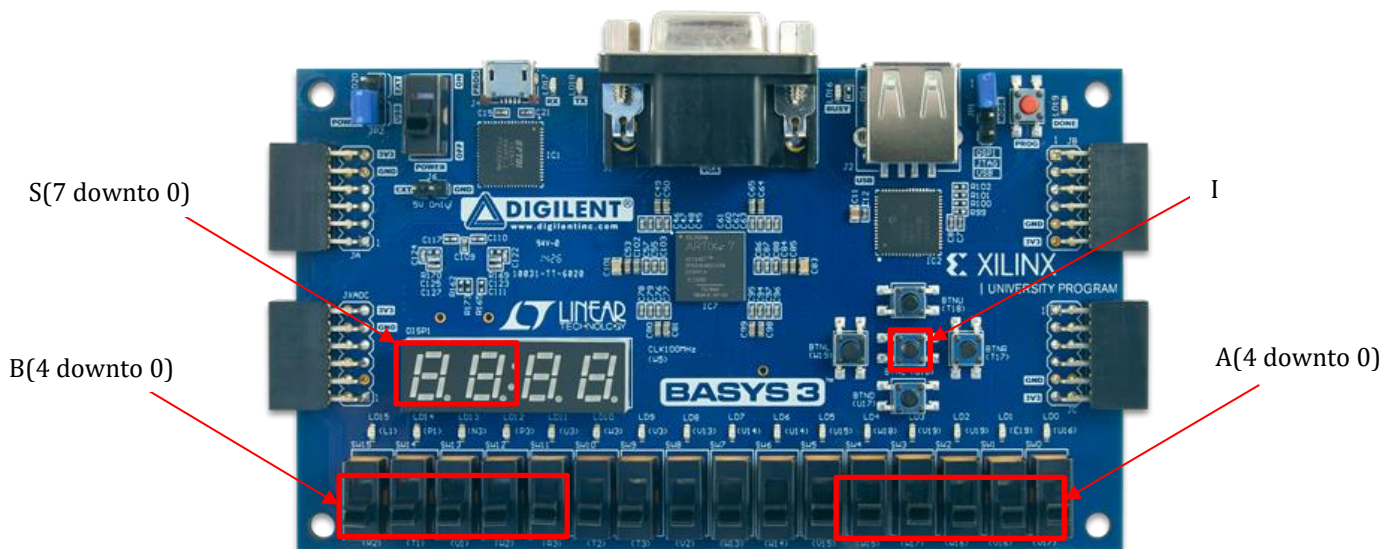


Figura 2. Placa de prototipagem Basys 3.

Para realizar o teste do circuito *lab2* projetado utilizando a placa de prototipagem (*Digilent Basys 3*, equipada com a FPGA *Artix-7* da Xilinx – ver Figura 5), será disponibilizado na aula um conjunto de ficheiros que irá utilizar nesta parte do trabalho:

- `sd.vhd` – descrição do circuito principal (da placa)
- `Basys3_Master.xdc` – configuração dos portos (da placa)
- `clkdiv.vhd` – divisor de frequência (especificação)
- `disp7.vhd` – bloco do controlo do display de 7 segmentos (especificação).

Não modifique os nomes destes ficheiros!

G(2.1.a)

1. Na folha de respostas da aula será pedida a implementação de uma unidade aritmética semelhante á de casa, mas com outras funções.
2. Estará disponível no laboratório um projeto vivado com todos os ficheiros necessários a realização do trabalho na aula. Nomeadamente, os ficheiros `sd.vhd`, `clkdiv.vhd`,

disp7.vhd e Basys3_Master.xdc que definem o interface com a placa de prototipagem Basys 3.

3. Verifique se o ficheiro **sd.vhd** está definido como módulo de topo (faça clique direito no ficheiro e selecione a opção “Set as Top”). Verifique também se a hierarquia do projeto inclui os componentes `clkdiv`, `disp7` e `Basys3_Master.xdc`, conforme indicado na Figura 6. A inclusão destes componentes é obrigatória e deve ser sempre verificada (**em particular, a não inclusão do ficheiro Basys3_Master.xdc, pode DESTRUIR o dispositivo, e caso isso aconteça ser-lhe-ão pedidas responsabilidades**).

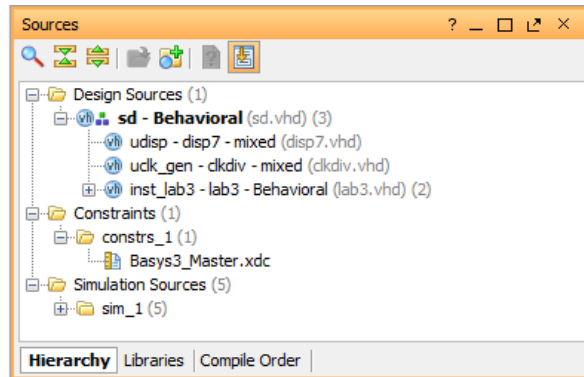


Figura 3. Hierarquia do projeto incluindo `sd`, `clkdiv`, `disp7` e `Basys3_Master.xdc`

4. As seguintes ligações foram estabelecidas de forma a possibilitar a correta interação do utilizador com o circuito;
 - a) Os bits da entrada **A** estão ligados aos interruptores **SW (4 : 0)**, i.e., os primeiros quatro interruptores (*switches*) do lado direito da placa;
 - b) Os bits da entrada **B** estão ligados aos interruptores **SW (15 : 11)**, i.e., os primeiros quatro interruptores (*switches*) do lado esquerdo da placa;
 - c) A entrada **I** está ligada ao botão de pressão **BTN (4)**, i.e., o botão central;
 - d) O resultado calculado pelo circuito, i.e., a saída **S** do circuito **lab2**, é apresentada nos dígitos 1 e 0 do display de 7 segmentos em formato **hexadecimal**, i.e., **disp1** e **disp0**;
5. Na aula de laboratório, implemente o circuito na placa de desenvolvimento. Para tal, siga as instruções disponibilizadas no "[Guia de Implementação de Circuitos na Placa de Desenvolvimento](#)". Note que o interruptor ON/OFF da placa deve estar na posição ON.
Nota: durante a síntese do circuito, a ferramenta poderá indicar um conjunto de avisos (*warnings*) e erros. Os erros deverão ser **todos** corrigidos; os *warnings* podem, em geral, ser ignorados, sendo que alguns são originados pelo facto de ter entradas/saídas no ar.
6. Verifique o correto funcionamento do circuito. Mostre-o ao docente.