

## Desenvolvimento de um circuito para operações sobre vectores booleanos e ternários

Bruno Pereira

**Resumo** – Este artigo descreve um circuito baseado numa FPGA (*Field Programmable Gate Array*) que implementa cinco operações sobre vectores booleanos e ternários. A interação com o circuito é feita através de um *Touch Panel*. A descrição do circuito foi realizada na linguagem VHDL com recurso à ferramenta Xilinx ISE 5.2. O circuito foi testado na FPGA XC2S300E com a ajuda da placa TE-XC2Se da Trenz Electronic. Este artigo tem como objectivo dar a conhecer como foi desenvolvido este circuito e o seu modo de funcionamento. O respectivo projecto para Xilinx ISE 5.2 (que pode ser usado para testar o circuito) está disponível na WebCT [1].

**Abstract** – This paper describes an FPGA-based circuit that implements five operations over Boolean and ternary vectors. The interactions with the circuit has been provided through a Touch Panel. The design flow is based on VHDL and some other tools available for ISE 5.2 software of Xilinx. The circuit was tested in FPGA XC2S300E with the aid of the prototyping board TE-XC2Se supplied by Trenz Electronic. The primary objectives of the paper are to show how this circuit has been developed and how it works. The respective ISE 5.2 project is available on WebCT [1].

### I. INTRODUÇÃO

O trabalho que é apresentado neste artigo foi realizado no âmbito da disciplina de CR (Computação Reconfigurável) leccionada no Departamento de Electrónica e Telecomunicações da Universidade de Aveiro. Para a realização deste trabalho foi necessário o conhecimento adquirido nas aulas de CR assim como os tutoriais que estão colocados no site da disciplina [1].

O trabalho apresentado teve como objectivo a criação de um circuito baseado numa FPGA (*Field Programmable Gate Array*) que permitisse a realização de cinco operações sobre vectores booleanos e ternários através de um *touch panel* (ver fig. 1). As operações permitidas ao utilizador foram: averiguar a existência de uma sequência de uns; indicar o número de uns no primeiro vector e finalmente, permitir verificar se os dois vectores são ortogonais.

No intuito de proporcionar ao utilizador uma situação mais familiar e simplificada com o circuito, no *touch panel* quando ligado o circuito, são apresentadas as opções que o utilizador tem para construir os seus vectores, assim como as operações permitidas (ver fig. 1).



Fig. 1 - Placa TE-XC2Se interligada com o Touch Panel

#### A. Características principais da placa TE-XC2Se[2]:

- 300.000 portas de sistema
- 8 Mbit (1M \* 8 / 512K \* 16) de Memória Flash
- 4 Mbit (256K \* 16) de RAM estática
- LCD com 2\*16 caracteres
- Interface USB (*Universal Serial Bus*)
- Interface RS232
- LEDs, interruptores e botões
- 2 osciladores que geram sinais de relógio de 48 MHz e 25 MHz

#### B. Características principais da FPGA XC2S300E [3]:

- Array de CLBs 32\*48
- RAM distribuída, 98 304 bits
- Blocos de RAM, 64 Kbits

#### C. Touch Panel

O *touch panel* utilizado pertence ao kit EA KIT240-7 e as suas principais características são: possuir 60 áreas de toque (6 linhas \* 10 caracteres) e 240\*128 pixels [4].

#### D. Comunicação entre o Touch Panel e a placa

A comunicação entre os dois dispositivos é controlada pelo interface standard RS232. Esta transferência de dados é realizada assincronamente, logo entre os bits de informação a enviar vão existir bits de controlo. No início da *stream* a transmitir é enviado o *Start Bit*, de seguida são transmitidos os 8 bits de dados (informação para o *touch panel*) e finalmente é enviado o *Stop Bit*. Na *stream* de dados o primeiro bit a ser enviado é o menos significativo.

A taxa de transmissão pode variar entre 1.200 e 115.200 baud.

## II. ESPECIFICAÇÃO

O trabalho consistiu na criação de um circuito, a partir da linguagem VHDL, que implementa operações sobre vectores booleanos e ternários. Os vectores podem ser constituídos no máximo por vinte elementos. É permitido ao utilizador realizar cinco operações: 2, 3, 4, *Cont* e *Ort*. As operações 2, 3 e 4 permitem averiguar se existe no primeiro vector uma sequência de dois, três ou quatro uns respectivamente; a operação *Cont* possibilita contar o número de uns do primeiro vector e finalmente, a operação *Ort* que permite verificar se os dois vectores são ortogonais.

#### A. Descrição por blocos

A especificação do trabalho está dividida em seis blocos (ver fig. 2), tendo cada um destes uma função bem definida. Para uma melhor compreensão do trabalho estes blocos podem ser divididos em três grupos funcionais: *Relógio*, *Cálculos* e *Troca de informação*.

Nesta perspectiva, o primeiro grupo é composto por dois blocos: *clk\_res* e *rs\_divider*, tendo como função sincronizar os dois dispositivos, o *touch panel* e a FPGA. A placa TE-XC2Se possui um relógio de 48 MHz, este sinal vai ter que ser dividido para que deste modo seja possível os dispositivos comunicarem entre si e assim o utilizador interagir.

O segundo grupo, *Cálculos*, tem como função realizar os cálculos das operações disponibilizadas ao utilizador, sendo este constituído por três blocos: *contar*, *orthogo* e *seguidos*. O bloco *contar* permite contar o número de uns do vector que recebe; o bloco *orthogo* permite verificar se os dois vectores são ortogonais; o bloco *seguidos* permite contar o número máximo de uns seguidos do vector que recebe.

O grupo *Troca de informação* é composto pelo bloco *rs\_control* que tem como função ler a informação que recebe do *touch panel* assim como enviar informação para ser visualizada no *touch panel*. É neste bloco que a informação recebida é filtrada e armazenada para futuras operações.

A Fig. 2 apresenta todos os blocos mencionados acima no editor da ISE 5.2.

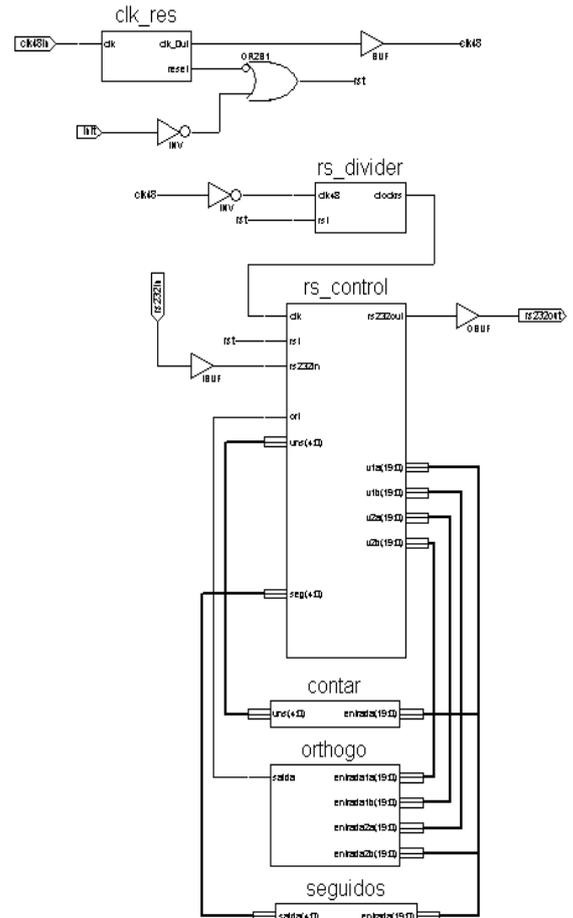


Figura 2 – Blocos do projecto

#### B. Descrição pormenorizada dos blocos

Os blocos *clk\_res* e *rs\_divider* são constituídos por processos que dividem a frequência do relógio. O bloco *rs\_divider* é responsável pelo *baud rate* que neste caso é de 115.200 baud.

O bloco *rs\_control* é composto por dois processos.

- O primeiro processo tem como função enviar informação para o *touch panel*. Assim, inicialmente vai ser enviado para o *touch panel* toda a informação necessária para construir o interface com o utilizador. Este bloco tem ainda como função validar a informação recebida pelo *touch panel* e consoante esta apresentar o resultado da operação solicitada pelo utilizador.

- O outro processo é utilizado para ler a informação recebida do *touch panel*, assim como para guardar os valores dos vectores introduzidos pelo utilizador.

O bloco *contar* é constituído por um processo e por uma função. O processo chama a função tendo esta como valor de entrada o vector que o bloco recebe e de saída o número de uns que esse vector possui. O número de uns é calculado na função através de um ciclo *for*. O processo envia para o bloco *rs\_control* o resultado.

Tal como o bloco anterior, o bloco *orthogo*, também possui um processo e uma função e o modo de

funcionamento é semelhante. Contudo, este bloco possui um maior número de entradas. As entradas deste bloco são os dois vectores e os respectivos inversos.

O bloco *seguidos* é também semelhante aos anteriores. Este tem uma entrada, o vector que se pretende contar os uns e uma saída. A saída é o número máximo de uns seguidos que o vector possui, sendo este número calculado numa função utilizando um ciclo *for* da linguagem VHDL.

### C. Exemplo de código

O código relativo ao circuito implementado está disponível na WebCT [1].

Para uma melhor compreensão de como foram construídos os blocos, que calculam os resultados das operações e da utilização de funções e de ciclos *for*, é apresentado a seguir uma parte de código pertencente ao bloco *seguidos*:

```
architecture Behavioral of seguidos is

function segui (input :
std_logic_vector) return
std_logic_vector is

variable temp : std_logic_vector (4
downto 0) ;
variable t : std_logic_vector ( 4
downto 0);

begin
    temp:=(others=>'0');
    t:=(others=>'0');

    for i in input'range loop
        if (input(i)='1') then
            t:=t+1;
            if t > temp then
                temp:=t;
            end if;
        else
            t := "00000";
        end if;
    end loop;
    return temp;

end segui;

begin

    saida <= segui (entrada);

end Behavioral;
```

Como é possível observar neste bloco, a função chamada tem como valor de entrada o da porta de entrada. No início da função existem duas variáveis que são

colocadas a zero. A variável *t* conta os uns que o vector possui contudo, sempre que um elemento do vector é diferente de um esta é colocada a zero. A variável *temp* tem como função guardar o número máximo de *t* ou seja o número máximo de uns seguidos que surgiram até ao momento. Para ser possível contar os uns existentes no vector é utilizado um ciclo *for*, visto que este percorre todo o vector permitindo assim, a pesquisa de todos os seus elementos.

Por fim, a saída do bloco é o resultado da função.

## III. IMPLEMENTAÇÃO

O projecto consiste essencialmente em ler valores do *touch panel*, guardá-los e depois de ter realizado uma operação apresentar o resultado.

Quando o *touch panel* envia um valor para a placa, este é validado e dependendo se pertence a um vector ou a uma operação tem tratamentos diferentes. Se o valor lido pertencer a um vector, o modo como este é guardado é muito importante, porque pode permitir calcular o resultado das operações facilmente. O sistema utilizado para guardar os valores introduzidos pelo utilizador consiste na utilização de dois registos e assim dependendo dos valores de entrada cada registo vai ter um valor distinto (ver Tabela 1).

Valores introduzidos	0	1	-
1ª registo (original)	0	1	0
2ª registo (inverso)	1	0	0

Tabela 1 – Processo utilizado para guardar valores

Se o valor lido corresponder a uma operação então, dependendo do resultado obtido no bloco correspondente a essa operação, vai ser escrita no *touch panel* uma resposta podendo esta ser positiva, negativa ou um número.

## IV. MODO DE UTILIZAÇÃO

Ao realizar este projecto uma preocupação constante foi tentar que o interface de comunicação com o utilizador fosse intuitivo, para que deste modo, o utilizador pudesse utilizar o sistema sem dificuldade.

Assim, depois do circuito estar carregado na FPGA e de o interface ter sido apresentado no *touch panel*, o utilizador pode inicializar a sua utilização.

### A. Introdução de valores

Para introduzir os valores do primeiro vector é necessário pressionar no *touch panel* os quadrados da área que esta identificada com a letra *A* na fig. 3. Para introduzir o segundo vector o processo é idêntico, basta o utilizador pressionar os quadrados da área que está representada pela letra *B* na mesma figura, que contém os diferentes valores para o segundo vector. É necessário ter em atenção que ambos os vectores estão limitados por um

número máximo de vinte valores. Quando é introduzido um valor que faz com que a dimensão máxima (20 valores) do vector seja ultrapassada, esse valor não é considerado.

### B. Escolha da operação

Como já foi referido anteriormente o utilizador tem cinco operações disponíveis, estas estão representadas no *touch panel* pela área D da fig. 3. Assim para escolher uma operação o utilizador tem que pressionar o quadrado identificativo da operação que pretende realizar. Para as operações 2, 3, 4 (verificar número de uns seguidos) e Cont (contar o número de uns do primeiro vector) não é necessário que sejam introduzidos os dois vectores pois as operações são realizadas sobre o primeiro vector.

### C. Resultado

O resultado das operações pode surgir em dois locais diferentes. Assim, para as operações 2, 3, 4 e Ort a resposta é Sim ou Não (consoante se verifica ou não) como se pode visualizar no *touch panel* da fig. 4. Mas se a operação escolhida for Cont, o resultado irá surgir na área identificada pela letra E da fig. 3.

### D. Recomeçar

Os valores escolhidos pelo utilizador surgem no *touch panel* à medida que vão sendo introduzidos (fig. 3, área C). Ao escolher os valores para os vectores o utilizador pode se enganar e caso isso aconteça, este só tem que pressionar a área do *touch panel* que diz Anular (fig. 3) e assim o *touch panel* volta ao estado inicial e os vectores serão apagados.

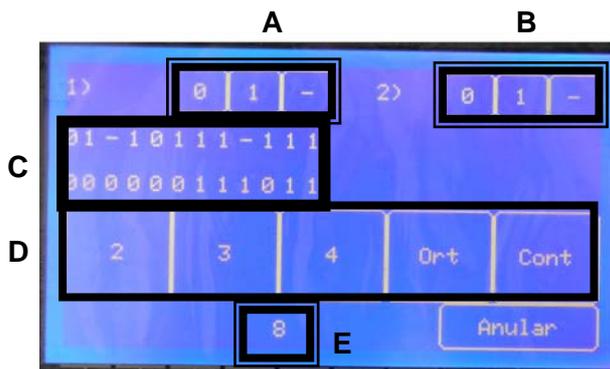


Fig. 3 – Exemplo de utilização inserindo os dois vectores e escolhendo a operação contar uns



Fig. 4 – Exemplo de utilização inserindo dois vectores e escolhendo a operação ORT

## V. CONCLUSÃO

Este trabalho tornou-se bastante enriquecedor, na medida em que inicialmente foram verificadas algumas dificuldades, quer a nível da linguagem VHDL, assim com alguma inexperiência em programar componentes reconfiguráveis. Contudo, estas dificuldades foram ultrapassadas ao estudar os tutoriais [1] disponíveis e com a matéria leccionada nas aulas de CR. As principais dificuldades na realização deste trabalho foram a leitura e escrita de dados no *touch panel*, e o facto da programação não ser sequencial, visto que até esta disciplina este foi o único tipo de programação desenvolvido.

Assim, tendo em conta que o trabalho desenvolvido corresponde aos requisitos definidos inicialmente é possível afirmar que o projecto foi realizado com sucesso.

## REFERÊNCIAS

- [1] <http://webct.ua.pt>, "2º Semestre", disciplina "Computação Reconfigurável".
- [2] <http://www.trenz-electronic.de/prod/proden12.htm>
- [3] Valery Sklyarov, Iouliia Skliarova, "Ferramentas para desenvolvimento de sistemas digitais reconfiguráveis", Electrónica e Telecomunicações, Vol. 3, Nº 8, Jan. 2003, pp. 743-764.
- [4] <http://www.qscomp.cz/Pdf/kit2407e.pdf>