

Desenvolvimento de uma calculadora baseada numa FPGA e num VGA

Christophe Pires

Resumo - Este artigo descreve uma calculadora baseada numa FPGA da família *Spartan-III* da *Xilinx* que implementa operações aritméticas elementares. Os valores dos operandos e a operação são seleccionados num teclado virtual no VGA através dos botões da placa de protótipo TE-XC2Se da *Trenz Electronic*.

Abstract - This paper describes an FPGA-based implementation of a simple calculator for executing elementary arithmetic operations over 3-digits operands. The values of operands and operation are selected with the aid of push-buttons available on the FPGA prototyping board TE-XC2Se, which was used for experiments. The results are displayed on a VGA monitor screen.

I. INTRODUÇÃO

O trabalho descrito neste artigo foi realizado na disciplina de Computação Reconfigurável leccionada à licenciatura de Engenharia de Computadores e Telemática no departamento de Electrónica e Telecomunicações da Universidade de Aveiro no segundo semestre do ano lectivo 2004/2005 e baseou-se no trabalho da aula prática 5 [1] bem como em trabalhos dos anos anteriores [2].

A. Objectivos

O objectivo do trabalho em causa foi desenvolver uma calculadora que permitisse a realização das operações adição e subtracção sobre dois operandos de três algarismos cada. Os operandos e o resultado obtido são apresentados em notação decimal. Os operandos e as operações são seleccionados através de um teclado virtual visualizado no monitor VGA. Os botões da placa de protótipo TE-XC2Se [3] que contém a FPGA são utilizados para mover o cursor do teclado virtual, seleccionar um símbolo deste conforme ilustrado na figura 1 e apresentar o resultado no monitor VGA.

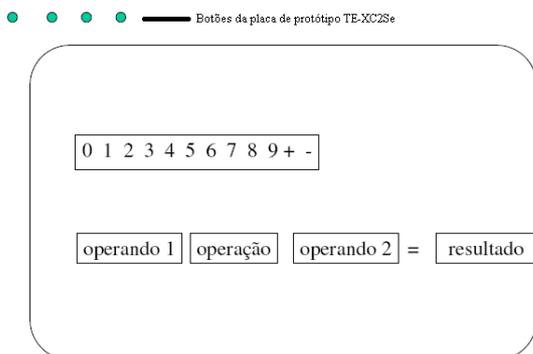


Fig. 1 — Interface do projecto e botões da placa de TE-XC2Se

II. Especificação

A. Descrição das funções dos botões

As funções dos botões (numerando os botões por ordem crescente da esquerda para a direita a partir de 0) são as seguintes:

- Botão 0: deslocar o cursor para a esquerda;
- Botão 1: deslocar o cursor para a direita;
- Botão 2: seleccionar o símbolo do teclado virtual onde se encontra o cursor;
- Botão 3: calcular o resultado.

B. Descrição por módulos

A seguir descrevem-se os diferentes módulos apresentados na figura 2 e que foram utilizados na implementação da calculadora.

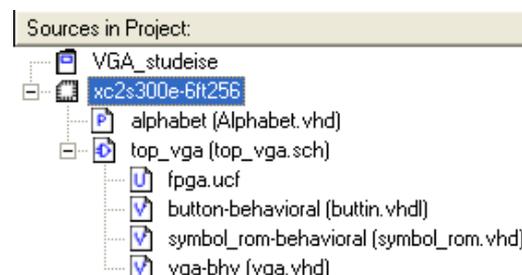


Fig. 2 — Módulos do projecto [4].

1. PACOTE ALPHABET.VHD: Neste pacote definem-se os símbolos gráficos dos algarismos de 0 a 9 e os símbolos '+', '-', '^', '#' e '.'. Os algarismos de 0 a 9 podem ser seleccionados para os algarismos dos operandos e os símbolos '+' e '-' como operação aritmética a executar. O símbolo '^' indica a posição do cursor do teclado virtual e o símbolo '#' é utilizado para delimitar a localização do teclado virtual, dos operandos, da operação e do resultado no monitor VGA;
2. MÓDULO TOP_VGA.SCH: Esquemático global do projecto apresentado na figura 3. O esquemático indica como os diferentes módulos interagem;
3. MÓDULO FPGA.UCF: Indica o modo como os pinos da FPGA são atribuídos assim como as definições a nível de sinais de relógio;

4. MÓDULO BUTTON.VHD: Este módulo retorna os valores dos botões da placa de protótipo TE-XC2Se. Estes valores são guardados num array de 4 posições, sendo o valor de cada posição alterado sempre que o botão correspondente é pressionado ou sempre que deixa de ser pressionado;

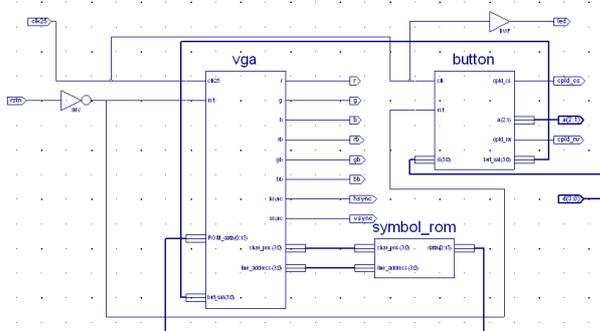


Fig. 2 — Esquemático global do projecto.

5. MÓDULO SYMBOL_ROM.VHD: ROM para os símbolos gráficos utilizados, no qual é feito o mapeamento destes em função da sua posição no pacote ALPHABET.VHD;
6. MÓDULO VGA.VHD: Interface que permite apresentar o teclado virtual, os operandos, a operação e o resultado no monitor VGA. Neste módulo são também realizadas as operações aritméticas.

C. Descrição de processos

Esta descrição cingir-se-á apenas aos processos de maior relevância para o projecto do módulo vga.vhd.

1. Processo print: este processo é responsável pela impressão do teclado virtual, dos operandos, do sinal da operação, do resultado e do cursor no monitor VGA.
2. Processo atrib: este processo é responsável pela atribuição dos valores seleccionados com o cursor aos operandos e à operação. É importante referir que não podem ser atribuídos os símbolos '+' e '-' aos operandos e à operação apenas podem ser atribuídos os símbolos '+' e '-'. A atribuição dos valores faz-se de um modo sequencial, em primeiro lugar atribuem-se os valores aos algarismos do operando1 (do mais significativo para o menos significativo), depois à operação e finalmente ao operando2 (novamente do mais significativo para o menos significativo). A atribuição é feita através do botão 2.
3. Processo Calculadora: Caso a operação escolhida seja a adição, efectua a adição dígito a dígito transportando o carry para o dígito seguinte. Não ocorre overflow, pois o resultado tem quatro dígitos e os operandos apenas têm três dígitos. Em relação à subtração adoptou-se uma estratégia diferente, assim, verifica-se qual deles é o maior e subtrai-se o

menor ao maior. Neste ponto também se define o sinal do resultado. Para obter cada um dos dígitos do resultado no código BCD (*Binary Code Decimal*), para sua posterior visualização no monitor VGA, foram tentadas várias estratégias:

- Tentou-se obter os vários dígitos por divisão inteira, mas tal não foi possível pois o VHDL sintetizável exige que os operandos sejam potência de 2;
- Tentou-se obter os vários dígitos através de iterações sucessivas em que em cada uma era subtraído um valor definido (100 ou 10). O número de iterações era guardado e no final correspondia ao valor do dígito. Esta solução verificou-se viável.

Para implementar a segunda estratégia apresentada relativa à subtração optou-se por utilizar ciclos. Mais uma vez surgiram alguns problemas com o VHDL sintetizável. Os ciclos de VHDL "loop...end loop;" e "while...loop...end loop;" não puderam ser aplicados devido a problemas relativos ao número de iterações que iriam efectuar. O erro apresentado indicava que esses ciclos não podiam efectuar mais do que 64 iterações, o que é muito estranho pois neste caso iriam efectuar no máximo 9 iterações. Assim, dada a impossibilidade de recorrer a esses ciclos, recorreu-se ao ciclo "for...end loop". O número máximo de iterações para esse ciclo é 9. De seguida apresentam-se os excertos de código onde se obtêm os algarismos do resultado [5].-

```
int3 := 0;
for i in 1 to 9 loop -- obtenção do algarismo
    --das centenas
    if(tempres > 99) then
        tempres := tempres -100;
        int3 := int3 +1;
    end if;
end loop;
int2 := 0;
for i in 1 to 9 loop -- obtenção do algarismo
    --das dezenas
    if(tempres > 9) then
        tempres := tempres -10;
        int2 := int2 +1;
    end if;
end loop;
int1 := tempres; -- obtenção do algarismo das
--unidades
```

4. Processo posição: este processo é responsável pela mudança de posição do cursor. O processo tem

associado um contador que é incrementado ou decrementado sempre que o botão 1 ou o botão 0 são premidos, respectivamente. Esse contador dá a posição lógica do cursor.

5. Processo relógio: processo de geração do relógio local utilizado nos processos calculadora, atrib, posicao e debouncing;
6. Processo debouncing: processo de debouncing utilizado para evitar o retransmitir da informação associada a uma único pressionar de botão;

V. CONCLUSÃO

A implementação desta calculadora simples em que os operandos são introduzidos e a operação aritmética é escolhida através de um teclado virtual foi interessante pois possibilitou um aprofundar das potencialidades de interacção das FPGA's com o monitor VGA, tal como

também permitiu consolidar os conhecimentos de VHDL adquiridos.

Os problemas que foram surgindo aquando da realização do projecto foram ultrapassados sem muita dificuldade, pois a maioria foi devida à utilização de ciclos e já tinham sido dificuldades semelhantes nas aulas práticas.

REFERÊNCIAS

- [1] Página da disciplina Computação Reconfigurável, <http://www.ieeta.pt/~iouliia/Courses/CR>.
- [2] <http://elearning.ua.pt/>, disciplina Sistemas Digitais Reconfiguráveis.
- [3] <http://www.trenz-electronic.de/>.
- [4] <http://www.xilinx.com/>.
- [5] V. Pedroni, Circuit Design with VHDL, MIT Press, 2004.