

Desenvolvimento de uma placa protótipo baseada numa FPGA

Manuel Almeida

Resumo – Este artigo descreve a implementação de uma placa protótipo tendo por base uma FPGA (Field Programmable Gate Array) XC3S400-4PQ208 da família Spartan-3, como componente reconfigurável. Para além de uma descrição dos elementos mais importantes também é referido as ferramentas e aplicações necessárias para a funcionalidade da placa.

Abstract - This paper describes the implementation of a prototyping board designed on the basis of the FPGA (Field Programmable Gate Array) XC3S400-4PQ208 from the family Spartan-3 of Xilinx. It characterizes all major elements of the board, considers possible development tools and demonstrates feasible practical applications.

I. INTRODUÇÃO

O hardware reconfigurável permite acelerar significativamente a resolução de problemas em várias áreas, tais como tratamento de imagens, optimização combinatória, multimédia, etc. É de salientar contudo que este possui muitas restrições que impedem a realização de operações apenas em hardware. Para ultrapassar esta desvantagem recorre-se às técnicas baseadas em partição software/hardware ou à reconfiguração dinâmica. Como resultado, problemas complexos devem ser decompostos em sub-sistemas implementados em software e hardware. Neste caso a interacção entre os sub-sistemas interligados que processam várias partes do mesmo problema, torna-se muito importante. O projecto proposto foi dedicado a esta área e consistiu na criação de uma placa protótipo, tendo como componente reconfigurável principal uma FPGA.

A FPGA escolhida foi a XC3S400-4PQ208 da família Spartan-3 [1] da Xilinx, que é baseada na tecnologia 90nm. Com esta tecnologia, a Xilinx, aumentou a eficiência de fabrico e diminuiu até 80% o tamanho da pastilha, e consequentemente a diminuição dos preços dos componentes.

A placa protótipo tem uma base simples ao nível da sua constituição e um baixo custo associado. Ela permite o controle de vários periféricos, ou funcione apenas como o núcleo de um sistema reconfigurável. Esta placa pode ser utilizada no processo educativo e por empresas.

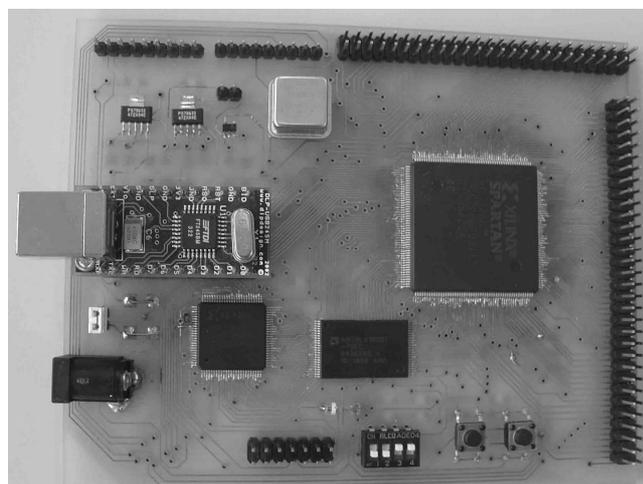


Fig. 1 – Protótipo da placa intermédia

II. ESPECIFICAÇÃO E DESCRIÇÃO DOS DISPOSITIVOS INTEGRANTES DA PLACA

A Fig. 1 mostra uma placa intermédia, criada com o intuito de efectuar testes ao nível do hardware e software. Inicialmente todos os dispositivos foram testados de forma individual, tendo-se avançado de seguida para a interligação entre eles.

A existência de dispositivos com limitações ao nível da frequência de funcionamento, levou a um processo de testes mais pormenorizado, para optimização da frequência de trabalho entre dispositivos. É de salientar que a utilização de uma frequência mais baixa por parte de um dispositivo não leva a que todo o sistema funcione a essa frequência. A interligação entre dispositivos dá-se sempre à frequência do dispositivo que suporta a menor frequência.

Após os vários testes efectuados, a placa protótipo final apresentará umas ligeiras alterações ao nível da sua aparência, já que ela, ao contrário da apresentada na Fig. 1, vai ser fabricada com um maior número de *layers* conseguindo-se assim uma placa protótipo de dimensões mais reduzidas.

A placa protótipo é constituída por um conjunto de elementos (ver Fig. 2) que tem como objectivo a funcionalidade e configuração do elemento principal, que é a FPGA. Nesta secção descreve-se os elementos principais que integram a placa.

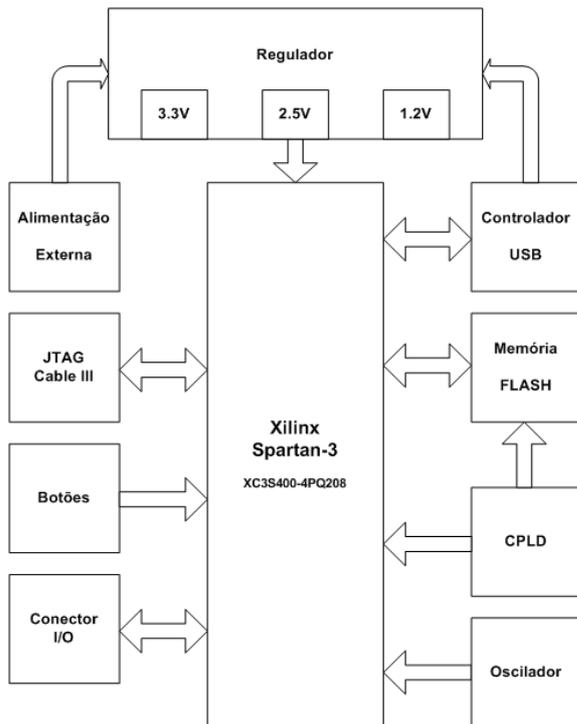


Fig. 2 – Diagrama de blocos da placa

A. FPGA XC3S400 da família Spartan-3

A FPGA utilizada como componente reconfigurável, contém 400 mil portas de sistema, 56Kb de RAM distribuída, 16 multiplicadores, 141 pinos de entrada e saída, etc. Esta é uma FPGA recente e utiliza as mais recentes tecnologias.

A FPGA possui 3 tipos de alimentação : V_{CCO} a 3.3V, V_{CCAUX} a 2.5V e V_{CCINT} a 1.2V. A alimentação da placa é fornecida através da porta USB do computador, à qual a placa está ligada, ou externamente, conseguindo-se as respectivas tensões através de reguladores de tensão.

A configuração da FPGA é efectuada pela memória *flash* e pela CPLD (Complex Programmable Logic Device) [2]. A FPGA também suporta a configuração através de JTAG (ver Fig. 2), sendo utilizado para o efeito o cabo paralelo III, IV ou multilix e a ferramenta IMPACT da Xilinx. Desta forma, o *bitstream* será colocado na FPGA directamente.

A FPGA utilizada trabalha a uma frequência que pode ir até aos 326 MHz. Portanto, o oscilador que integra a placa protótipo pode variar, mas neste caso é utilizado um oscilador de 80 MHz.

B. Controlador USB

A placa protótipo contém um módulo concebido pela empresa DLP Design [3] que utiliza o dispositivo FT245BM da FTDI [4]. Este módulo é constituído por alguns blocos, tais como, *transceiver* USB, oscilador de 6 MHz, multiplicador de relógio de 8 vezes, *buffers* do tipo

FIFO, bloco que implementa uma interface série, bloco que implementa o protocolo USB e uma EEPROM, que permite programar os parâmetros VID e PID, que é respectivamente a informação do produto do fabricante e o número de série. Esta funcionalidade, permite a identificação da placa no PC anfitrião sempre que a placa seja conectada à porta USB.

A interface (ver Fig. 3) entre este módulo USB-paralelo e um microprocessador ou microcontrolador, é facilitado devido à simplicidade do módulo em causa. Este módulo contém um conector USB do tipo 'B', um barramento de dados bidireccional de 8 bits, 4 pinos principais de controle dos dados, tal como esta representado na Fig. 3, e contém mais alguns pinos para controle quer ao nível dos dados quer ao nível das tensões de funcionamento do módulo.

Neste caso, o módulo permite a transferência de dados entre a FPGA e um computador anfitrião.

A placa protótipo por defeito é alimentada através da porta USB, deixando-a de ser apenas no momento em que se liga uma alimentação externa no *jack*.

Este módulo é compatível com o USB 1.1 e USB 2.0.

O módulo acima descrito é de vital importância para o funcionamento da placa protótipo. Ele permite a alimentação da placa, o envio do ficheiro de configuração da FPGA, o envio de dados de utilizador do computador anfitrião para a placa protótipo e vice-versa.

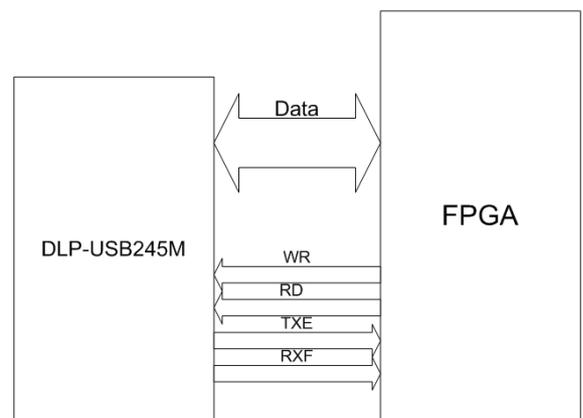


Fig. 3 – Ligações entre módulo DLP-USB245M e FPGA

C. Descrição e funcionamento da CPLD

CPLD é um dispositivo que integra no seu interior várias PLDs, interligadas por uma estrutura programável. A CPLD utilizada na placa protótipo pertence à família XC95xx da Xilinx, sendo a sua arquitectura composta por um conjunto de blocos funcionais e blocos de entrada/saída interligados por uma matriz de comutação que encaminha as entradas e saídas para os blocos funcionais.

A CPLD utilizada foi a XC9572XL, e tem como finalidade controlar o processo de configuração da FPGA.

O ficheiro de configuração encontra-se guardado na memória *flash*, e para se conseguir enviar os respectivos dados para a FPGA é necessário ter um elemento que controle a memória *flash*. De um modo mais simples, a CPLD tem a seu cargo o controle do barramento de endereços e pinos de controle da *flash*.

A especificação do projecto a implementar na CPLD é realizada na linguagem de descrição de hardware VHDL e a configuração é efectuada através da porta JTAG, utilizando a ferramenta IMPACT da Xilinx.

D. Memória Flash

A placa protótipo contém uma memória *flash* [5] de 16 Mbit, organizada em 2,097,152 bytes ou 1,048,576 palavras. A memória *flash* encontra-se dividida em sectores de 64Kbytes com excepção dos últimos quatro sectores. Para uma organização mais fácil, dividiu-se a *flash* em 3 sectores principais (ver Fig. 4). A primeira secção corresponde à área onde se guarda a *bitstream* de suporte à placa, que possibilita o controle de dados do USB e memória *flash*. A segunda secção, permite guardar a *bitstream* do utilizador. A terceira secção, possibilita o armazenamento de várias *bitstream* ou dados do utilizador. Esta terceira secção permite guardar um grande número de dados de utilizador, quer pela FPGA, quando este está a realizar um dado trabalho, quer directamente pelo utilizador, através da aplicação a correr no computador. Permite ainda guardar 6 ficheiros de configuração (*bitstream*). A escolha de um destes ficheiros é feita através de interruptores ligados ao barramento da CPLD. Existe ainda a possibilidade desta escolha ser realizada através da FPGA, permitindo uma reconfiguração dinâmica.

Os ficheiros de configuração guardados na memória *flash* permite a utilização da placa protótipo como um dispositivo autónomo.

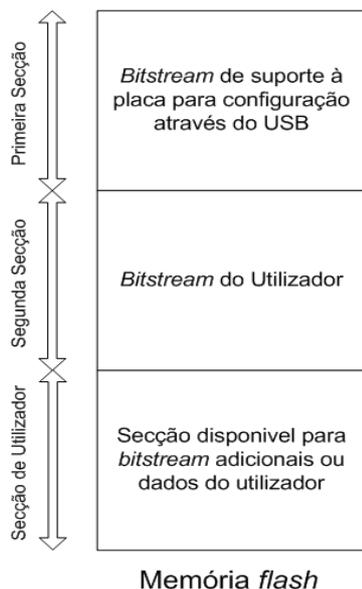


Fig. 4 – Estrutura funcional da memória *flash*

III. MODOS DE CONFIGURAÇÃO DA FPGA

A FPGA utilizada permite 5 modos de configuração que são : *master/slave serial*, *master/slave parallel* e *boundary scan (JTAG)*.

O modo de configuração utilizado pela placa protótipo é o *slave parallel*. A utilização deste modo deve-se ao facto de se conseguir tempos de configuração muito baixos. A FPGA a funcionar em modo *slave*, em vez do modo *master*, apenas indica que na configuração, o sinal de relógio necessário para a sincronização dos dados é fornecido por um dispositivo externo à FPGA, neste caso a CPLD. Em modo *master* seria a FPGA a fornecer o sinal de relógio.

A placa protótipo permite ainda outra forma de se configurar a FPGA, *boundary scan (JTAG)* (ver Fig. 5). Este modo permite de uma forma directa programar a FPGA. Para isso é necessário o cabo III, IV ou multilinx de JTAG e a ferramenta IMPACT, ambos da Xilinx. Tal como foi referido acima, é através deste modo de configuração que se configura a CPLD. Sempre que se utilize este modo de configuração na placa protótipo, aparece na ferramenta IMPACT uma corrente onde aparece primeiro a CPLD seguido da FPGA. Isto é, ao configurar através do JTAG é possível fazer a configuração dos dois dispositivos, não sendo obrigatório configurar ambos. De notar que esta sequência, primeiro a CPLD seguido da FPGA, não tem obrigatoriamente de ser desta forma.

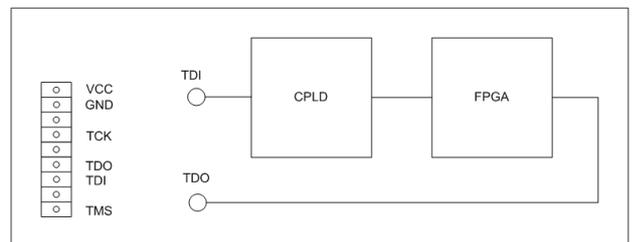


Fig. 5 – Configuração via JTAG

IV. ARQUITECTURA DA PLACA PROTÓTIPO

A arquitectura da placa protótipo (ver Fig. 6) foi projectada de modo a facilitar a sua configuração e utilização. Quanto à configuração, tal como referido na secção anterior, o modo paralelo permite uma configuração mais rápida mas ao mesmo tempo leva à utilização de mais pinos, aumentando a complexidade do circuito. Apesar deste aumento a placa protótipo fica a ganhar com esta opção. No que respeita à sua utilização, esta torna-se muito importante tendo em consideração os utilizadores alvo. Numa primeira fase, esta placa será utilizada no processo educativo fazendo com que se opte por uma forma mais simples, já que não é necessário a utilização de muitas opções por parte dos utilizadores.

A configuração é realizada através da porta USB do PC, onde corre uma interface gráfica que possibilita uma fácil utilização.

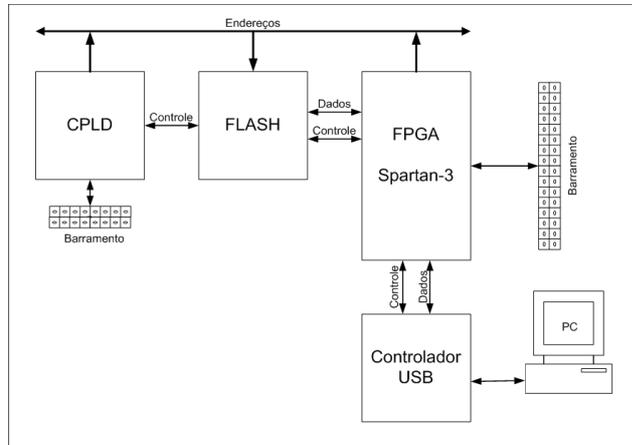


Fig. 6 - Arquitectura da placa protótipo

A placa protótipo disponibiliza uma grande quantidade de pinos de entrada/saída, possibilitando a ligação de vários dispositivos externos. Esta placa por defeito não inclui dispositivos tais como, porta série, VGA, Bluetooth, etc., já que o objectivo era projectar uma placa protótipo básica com a possibilidade de integração de outros módulos dependendo da necessidade do utilizador. Implementar uma placa com vários dispositivos e não os utilizar, é desperdiçar recursos.

V. SOFTWARE DE SUPORTE À PLACA PROTÓTIPO

Para controlar a placa protótipo é necessário um conjunto de aplicações e ferramentas. Uma das ferramentas utilizadas é o ISE (Integrated Software Environment) da Xilinx, que é um sistema de desenvolvimento assistido por computador. Esta ferramenta permite a criação de módulos em *VHDL*, criação de módulos no editor esquemático, implementação da ferramenta *Core Generator*, *IMPACT*, etc. Sobre esta ferramenta de software ISE da Xilinx pode consultar o artigo [6], onde se encontra descrito todos os passos necessários para o desenvolvimento e a implementação de circuitos digitais com base em FPGA. Os documentos [7] contêm todos os dados sobre as ferramentas da Xilinx para ISE 6.x.

Na secção anterior foi referido a existência de macros para o controle de alguns elementos. Criou-se uma macro em *VHDL* que controla o módulo USB. Para isso é necessário controlar alguns sinais (ver Fig. 3) que têm de ser controlados segundo um diagrama temporal fornecido pelo fabricante. É necessário ter em atenção a frequência de funcionamento da FPGA, já que o módulo em algumas situações, não suporta tempos de controlo muito baixos. Isto significa que para frequências elevadas, acima dos 40 MHz, seja necessário baixar a frequência de

funcionamento da FPGA, conseguindo-se através de uma macro implementada em *VHDL* que divide a frequência por 2^N , sendo N o valor possível de alterar.

Uma outra macro foi criada com o objectivo de controlar a memória *flash*. Esta macro, bastante mais complexa do que a anterior, controla uma grande quantidade de sinais. Os sinais a controlar são o barramento de dados (16 pinos), barramento de endereços (20 pinos) e 6 pinos de controle. Esta memória *flash* possui alguns comandos que para serem implementados é necessário seguir uma dada ordem. Por exemplo, para a escrita de dados é necessário apagar os sectores para o qual queremos escrever, sendo necessário 6 ciclos, e posteriormente são necessários mais 4 ciclos para a escrita. Cada ciclo respeita um diagrama temporal fornecido pelo fabricante. Dependendo da operação a executar, cada ciclo aqui referido pode corresponder a 3 ciclos de relógio usado pela FPGA.

O envio de dados por parte de um computador anfitrião para a placa necessita de uma aplicação que seja fácil a sua utilização. Esta aplicação desenvolvida em C# tem como objectivo o envio de um *bitstream*, que é um ficheiro binário que contém a especificação das funções e as ligações necessárias, a fim de construir o dispositivo físico que poderá ser testado em hardware. Esta aplicação também permite receber dados que são enviados pela placa protótipo. Esta possibilidade é muito importante ao nível da obtenção de resultados e ao nível do *debug* do programa a correr.

A FTDI fornece o *driver* necessário para o controlo da porta USB.

VI. INTERLIGAÇÃO DA PLACA PROTÓTIPO COM DISPOSITIVOS EXTERNOS

Módulos em hardware podem ser criados, permitindo à FPGA ser ligada a vários dispositivos externos. Estes módulos podem ser considerados como componentes de uma biblioteca que permite o desenvolvimento de sistemas baseados em FPGA, comunicando com computadores, memória externa e dispositivos periféricos. Isto permite utilizações muito úteis, tal como, o apoio para várias interfaces, interacção com painel de toque, textos e gráficos em LCD, comunicações com um rato, um teclado, monitores VGA, troca de dados entre placas protótipos, etc. Na projecção/realização de placas protótipo deve-se concentrar os esforços para o problema particular que é a resolução e envolvimento dos componentes acima mencionados, fornecendo somente benefícios úteis como a troca de dados, visualização de resultados intermédios e finais, depuração, etc.

Para um melhor entendimento ver o artigo [8], que contém, além de outras coisas, módulos reutilizáveis de interface para interacção entre FPGAs e dispositivos externos e alguns exemplos de projectos práticos que ilustra a aplicabilidade e eficácia de componentes reutilizáveis.

A. Controlo de um LCD

Na Fig. 7 é possível verificar um sistema constituído pela placa protótipo e um dispositivo externo que neste caso é um LCD (Liquid Crystal Display), O LCD usado contém quatro linhas de 20 caracteres cada. Com este dispositivo é possível visualizar um conjunto de informação enviada pela FPGA.

Uma aplicação básica do sistema da Fig. 7 é o do envio de dados por parte da FPGA e para além da visualização dos dados enviados também é possível manipular a sua visualização. Isto é, como o LCD tem um conjunto limitado de linhas e caracteres então pode-se, em caso de necessidade, manipular os dados enviados pela FPGA e implementar um conjunto de operações sobre o LCD. Essas operações pode ser o deslocamento de dados numa linha, o deslocamento de dados de uma linha para a outra, e em caso de necessidade de chamar a atenção de quem está a visualizar pode-se pôr a piscar durante alguns segundos o LCD. Isto é possível com uma máquina de estados finitos reprogramável. A abordagem aqui feita tenta apenas dar uma informação geral da constituição de sistemas e suas potencialidades. Em [8] está uma apresentação e explicação mais pormenorizada sobre este problema.

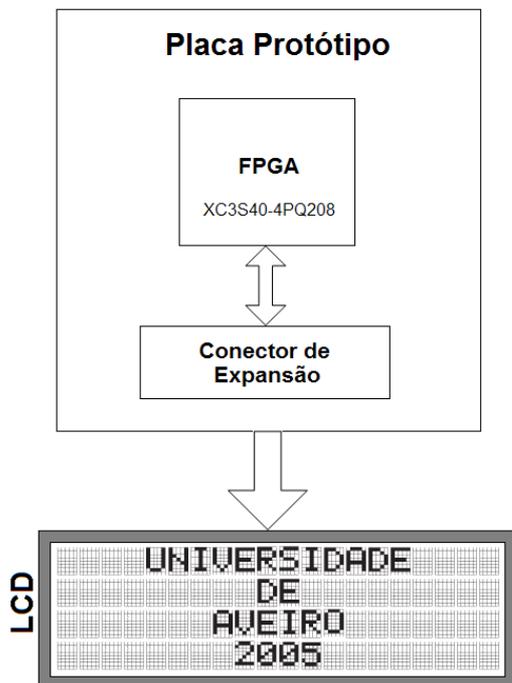


Fig. 7 – Experiência com a placa protótipo a controlar um LCD

Este tipo de dispositivo externo é bastante útil na depuração de circuitos.

Outro exemplo concreto é de um circuito que realiza cálculos aritméticos simples como a soma, subtração, multiplicação e divisão. Com este sistema e através de um módulo implementado em VHDL, pode-se visualizar os operadores, a operação a efectuar e o respectivo resultado.

Com este exemplo simples apenas se tenta demonstrar a utilidade dos dispositivos externos que se pode ligar à placa e não sendo estes dispositivos o elemento central do problema, em muitas ocasiões se tornam um auxílio muito importante.

B. Processador combinatório

Tal como foi referido anteriormente, esta placa permite acelerar significativamente a resolução de problemas em várias áreas. Na área da computação reconfigurável existe um grande número e variedade de problemas possíveis de resolver com esta placa protótipo. Existem muitos problemas práticos que podem ser formulados sobre modelos matemáticos tal como matrizes discretas, equações Booleanas, etc. A maioria destes modelos estão interligados, isto é, qualquer um destes modelos pode ser utilizado para o mesmo objectivo. Os problemas combinatórios podem ser resolvidos com a ajuda de algoritmos tais como satisfiabilidade booleana, cobertura mínima de matrizes, etc.

O projecto descrito a seguir implementa o algoritmo da cobertura mínima de matrizes [8,9,10]. Estes algoritmos utilizam muitos recursos, assim como, a utilização de muito tempo quando implementados em software. Para combater estes problemas, este algoritmo é implementado em hardware utilizando FPGAs.

Este algoritmo tem por base a aplicação de regras de redução de matrizes.

	A	B	C	D	E	F	G	H	I
1	1	0	0	0	0	1	0	1	1
2	0	1	1	0	0	0	1	0	0
3	0	0	1	1	0	1	0	0	1
4	1	0	0	1	1	0	0	0	0
5	0	0	1	0	0	0	1	1	1
6	0	1	0	0	0	1	0	0	0
7	0	1	0	0	1	0	1	0	1
8	1	0	1	1	0	0	0	1	0
9	0	0	0	0	0	1	0	1	0
10	0	0	0	1	0	0	0	0	1

Fig. 8 – Matriz binária 10 X 9

Considerando uma matriz binária, como por exemplo a da Fig. 8, com M linhas por N colunas, para encontrar a cobertura mínima desta matriz, é necessário encontrar o menor número de colunas por forma a que estas contenham pelo menos um '1' em cada linha. Para tal são utilizadas as seguintes regras que permitem simplificar a matriz:

Para $i \neq j$ se linha _{i} & linha _{j} = linha _{j} então a linha _{i} pode ser removida da matriz.

Para $i \neq j$ se coluna _{i} & coluna _{j} = coluna _{i} então a coluna _{j} pode ser removida da matriz.

Se uma coluna tiver todos os seus elementos iguais a '0', esta tem de ser removida da matriz.

Se uma linha não tiver nenhum elemento igual a '1', então não será possível encontrar uma cobertura para a matriz.

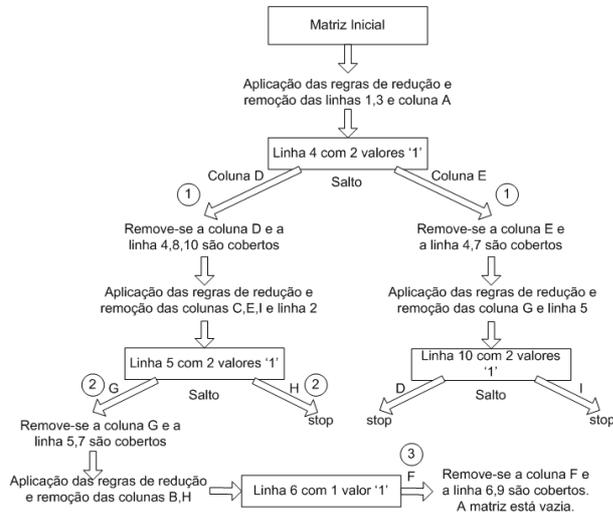


Fig. 9– Diagrama que encontra a solução para a matriz da Fig. 8

Aplicando estas regras até não ser possível simplificar mais a matriz, se todas as linhas tiverem mais do que um elemento igual a "1" então, para a primeira linha que contenha o mínimo número de elementos iguais a '1' é necessário analisar todas as soluções que contenham as colunas correspondentes aos elementos iguais a '1' desta linha, por forma a escolher a solução que utilize o menor número de colunas.

Na Fig. 9 encontra-se um diagrama que exemplifica, passo a passo, o algoritmo da cobertura mínima de matrizes. O exemplo referido é aplicado à matriz da Fig. 8 sendo uma das soluções as colunas DGF.

VII. CONCLUSÕES

Durante o projecto, este apresentou uma complexidade na interligação entre os vários elementos que compõem a placa, que inicialmente não era previsível.

Pode considerar-se esta placa protótipo como a base de um sistema que suporta um componente reconfigurável poderoso como é a FPGA. Este projecto pode ser o início, ao nível do processo educativo, para a criação de placas protótipo mais elaboradas. É de salientar o constante aparecimento de novas FPGAs no mercado, mas com este trabalho realizado de base, a integração dessas mesmas FPGAs torna-se mais simples.

Ao nível do utilizador, esta placa torna-se simples já que necessita apenas de um cabo para a alimentação da placa e configuração da mesma.

Depois de desligada a alimentação da placa ela não perde a configuração que tinha anteriormente não sujeitando o utilizador a uma nova configuração.

Um factor importante que convém salientar, é o baixo custo desta placa. Apesar disso, esta placa contém componentes que utiliza a última tecnologia e que se encontra facilmente no mercado.

A placa protótipo foi projectada tendo em conta a simplicidade e o baixo custo, sem descorar a tecnologia actual.

AGRADECIMENTOS

O autor agradece ao Professor Valery Sklyarov pela ajuda prestada na elaboração deste artigo.

REFERÊNCIAS

- [1] Spartan-3 FPGA Family, 2003 : <http://www.xilinx.com/>
- [2] CPLD XC9572XL High Performance, 2004 : <http://direct.xilinx.com/bvdocs/publications/ds057.pdf>
- [3] DLP-USB245M User Manual : <http://dlpdesign.com/>
- [4] FT245BM USB FIFO : <http://www.ftdichip.com/>
- [5] AM29LV160D Data Sheet : <http://www.amd.com/>
- [6] V. Sklyarov, I. Skliarova, "Ferramentas para desenvolvimento de sistemas digitais reconfiguráveis", E&T, Vol. 2, Nº 6, Set. 2002.
- [7] http://www.xilinx.com/support/sw_manuals/xilinx6/download/
- [8] V. Sklyarov, I. Skliarova, P. Almeida, M. Almeida. Design Tools and Reusable Libraries for FPGA-Based Digital Circuits. Proceedings of EUROMICRO Symposium on Digital Systems Design – DSD'2003. Belek Turkey, September, 2003. IEEE Computer Society, 2003, pp. 255-263
- [9] M. Almeida, P. Almeida, "Ferramentas de software para experiências com FPGA", projecto final de curso disponível na página <http://webct.ua.pt>, "2º Semestre", a disciplina "Computação Reconfigurável"
- [10] V. Sklyarov and I. Skliarova, "Architecture of Reconfigurable Processor for Implementing Search Algorithms over Discrete Matrices", Proceedings of ERSA'2003, Las Vegas, USA, 2003