

RELATORIO  
DO PROJECTO FINAL VHDL  
DA CADEIRA  
**SISTEMAS DIGITAIS  
RECONFIGURAVEL**

Data: 10.12.2006

BARTLOMIEJ WILKOWSKI – a38106

## Introducao

O objectivo deste projecto foi desenhar um sistema que pode encontrar numa matriz binária (gerada aleatoriamente com tamanho 20x20 e gravada numa memória do tipo *block*) todas as linhas que contêm um vector especificado de 8 bits.

## Descricao

O sistema consta das 8 modulos. Abaixo presenta-se as descricoes de cada um.

### 1. Modulo DCM.

Usa-se para dividir o clock e para obter sinais de clock e reset.

### 2. Modulo Vector.

Esta a gerar um vector de 8 bits que depois vai estar procurado na matriz binaria (20x20). Tem duas entradas, uma para o clock e o outro para o reset. Como a saida tem o vector de oito bits (*vector(7:0)*).

### 3. Modulo MatrixGenerator

Este modulo vai gerar a matriz binaria 20x20 de forma aleatoria. Como entradas tem: clk e reset, para controlar os processos, e vector de oito bits (*vector(7:0)*) – saida do modulo Vector. Como saidas tem um signal de um bit (*ready(0:0)*) para informar outros modulos se processo de generacao da matriz ja esta acabado. Tem valor ‘0’ se a matriz ainda nao foi gerada completamente, e ‘1’ quando o modulo acabou funcionar. Outra saida, *line(27:0)* e uma linha da matriz 20x20 (os primeiros 20 bits (19:0)) e depois ultimos 8 bits (27:20) e um numero BCD que representa numero de vectores encontrados nesta linha. Ultima saida e o *address(7:0)* para escrever as linhas numa memoria RAM.

Para gerar matriz numa forma aleatoria tem dois signals: um de oito bits e um de quatro. Os valores destes sao inicialmente: “00000000” e “1100”. Com cada *rising\_edge* de clock, estes valores se mudam. Depois a linha esta gerada com umas operacoes binarias entre estes vectores e o vector de entrada. Estao usados aqui os operacoes **xor** e **and**. Todo o processo esta controlado pelo um contador.

### 4. Modulo MatrixRAM

Este modulo usa-se para guardar a matriz binaria. Consta de dois memorias tipu block (RAMB4\_S16\_S16). Como entradas tem alem de clock e reset, dois addresses: *RAM\_address\_write* e *RAM\_address\_read* para escrever e ler destas memorias. A ultima entrada e um vector de 28 bits (*write\_line(27:0)*), que consta duma linha da matriz (20 bits) e 8 bits como um numero BCD – numero dos vectores enctontrados. Como a saida, tem um vector de 28 bits lito com *RAM\_address\_read*.

Neste modulo, usa-se duas memorias para guardar primeiros 16 bit na primeira e

outros bits na segunda. Os addresses sao mesmos para as duas memorias para facilitar a leitura depois.

As entradas `RAM_address_write` e `write_line` vem do modulo anterior – MatrixGenerator.

#### 5. Modulo Processor.

Este modulo esta a ler as linhas da memoria RAM (modulo anterior) e depois mandando cada elemento (bit) para guarda-los na memoria de ecran. Os elementos mandados sao convertidos a valores hexadecimal. Este modulo comeca funcionar quando o modulo MatrixGenerator acaba escrever todas as linhas para a memoria RAM. Neste momento o signal *active* muda-se a '1' e o processador vai ler uma linha da memoria e depois processar os bits e manda-los para memoria do ecran.

#### 6. Modulos VGA\_S3, S\_ROM e C\_RAM

Sao os modulos desenvolvidos durante as aulas practicas e sao necessarios para mostrar os bits no ecran. No C\_RAM ha 3 blocos de memoria RAM, S\_ROM e VGA\_S3 comunicam-se e pintam os elementos no ecran usando um alfabeto feito (mapping das letras e numeros).

Quaisquer perguntas sobre o projecto pode-se fazer pelo mail: [wilkowskib@gmail.com](mailto:wilkowskib@gmail.com).