Pac-Man

Relatório da Implementação em VHDL

Duarte Ferreira Dias

71685

MIEET

DETI

Universidade de Aveiro

Índice

[Introdução 2](#_Toc421484642)

[Arquitetura 3](#_Toc421484643)

[Motor Gráfico 4](#_Toc421484644)

[Controlador do Jogo 4](#_Toc421484645)

[*Co*ntrolador de VGA 5](#_Toc421484646)

[Controlador do Pacman 6](#_Toc421484647)

[Controlador dos Fantasmas 7](#_Toc421484648)

[Conclusão 8](#_Toc421484649)

# Introdução

A implementação do jogo PAC-MAN em VHDL surgiu no âmbito da disciplina de Laboratórios de Sistemas Digitais.

O PAC-MAN é um jogo criado pela namco, que consiste em recolher o máximo número de pontos antes de ser apanhado, por um dos quatro fantasmas.

Este projeto foi implementado em duas placas de desenvolvimento com FPGA's das duas marcas líderes de mercado, a primeira, um kit da *terasic* (DE2-115) equipada com uma FPGA *CycloneIV* da *Altera* , a outra uma placa fabricada pela *Digilent* que tem o nome da FPGA da *Xilinx* que a equipa (Spartan-3e Starter Kit).

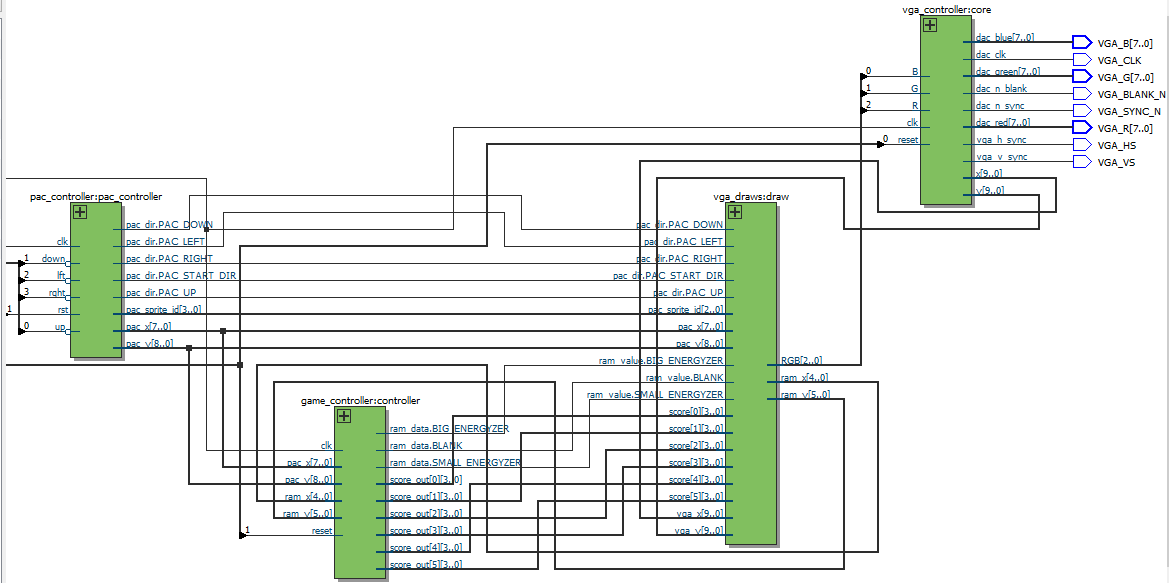
O IDE mais utilizado para a escrita e compilação deste projecto foi o ISE da *Xilinx*, tendo o *Altera* *Quartus* sido utilizado apenas para compilar o .sof final.

Comecei a implementar este projecto no inicio do 2º semestre deste ano, como tal, o controlador de VGA foi feito totalmente por mim recorrendo às fichas técnicas da placa da *terasic* e da *Diligent.*

Como não foi um projeto desenvolvido a tempo inteiro e não tinha grande experiência para fazer debugging o avanço do projeto foi algo lento e não consegui fazer uma réplica exata do jogo, ficando feito o motor gráfico do jogo e controlo do pacman que atualmente come os pontos do mapa fazendo-os desaparecer e incrementando um contador.

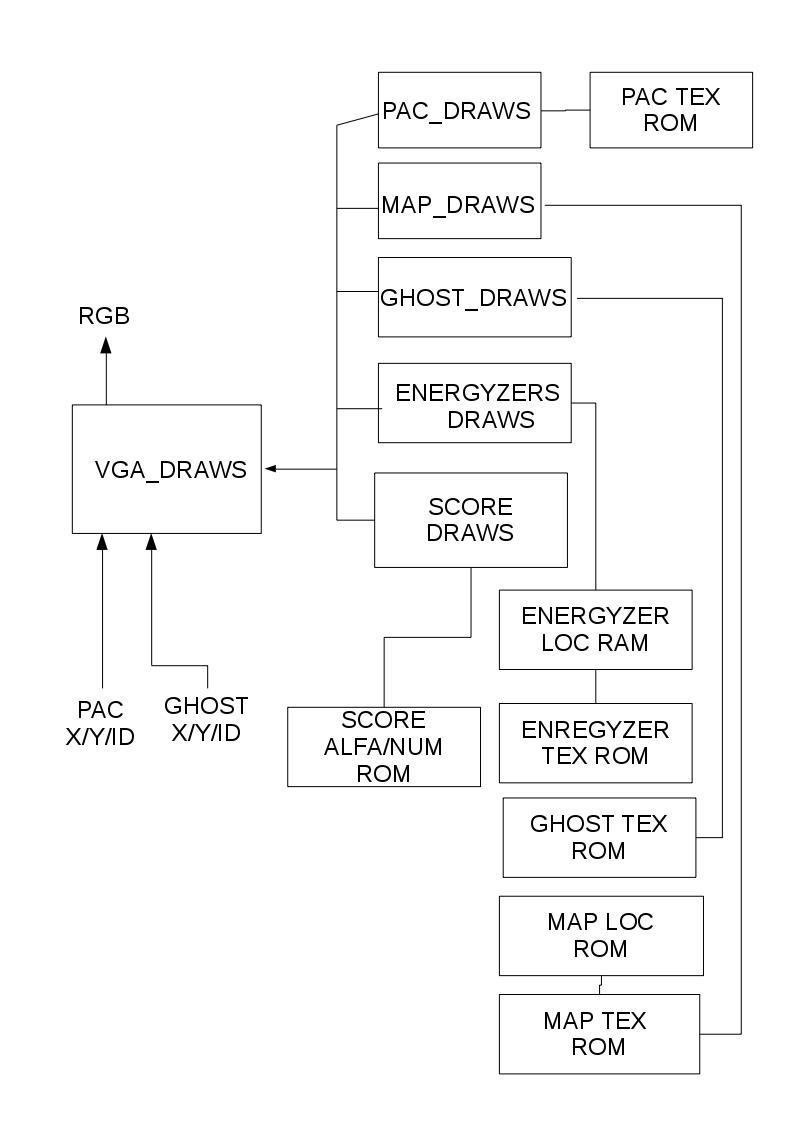
# Arquitetura

O pacman reconhece quatro blocos principais, o vga\_controller, o vga\_draws, o game\_controller, o pac\_controller e o ghosts controller.



## Motor Gráfico

O motor gráfico tem como bloco principal o vga\_draws, que é responsável por assegurar a prioridade de desenho das várias, para isso foi adicionado um bit adicional de cores ao vector de valores lógicos RGB, assim como de receber as informações posicionais dos elementos gráficos do jogo e distribui-los pelos vários blocos de desenho .



É no módulo vga\_draws que se encontram outros módulos como o map\_draws, o energyzers\_draws, ghost\_draws, pac\_draws. Todas estes módulos estão ligados a roms com as texturas a desenhar para uma dada posição de x e y ou tx e ty, sendo o tx e ty a divisão em quadrados 8 por 8 sendo 28 destes(tx) por cada 36 linhas(ty).

A exceção é o bloco energyzers\_draws que para saber qual textura desenhar recorre a uma ram que vai ser alterada conforme a passagem do pacman nos diversos tiles(necessário para que os pontos no mapa desapareçam).

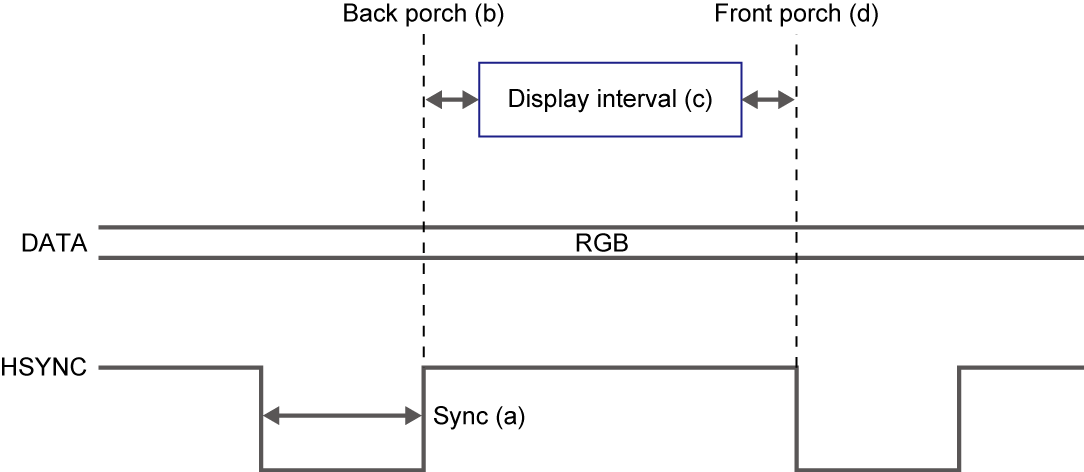
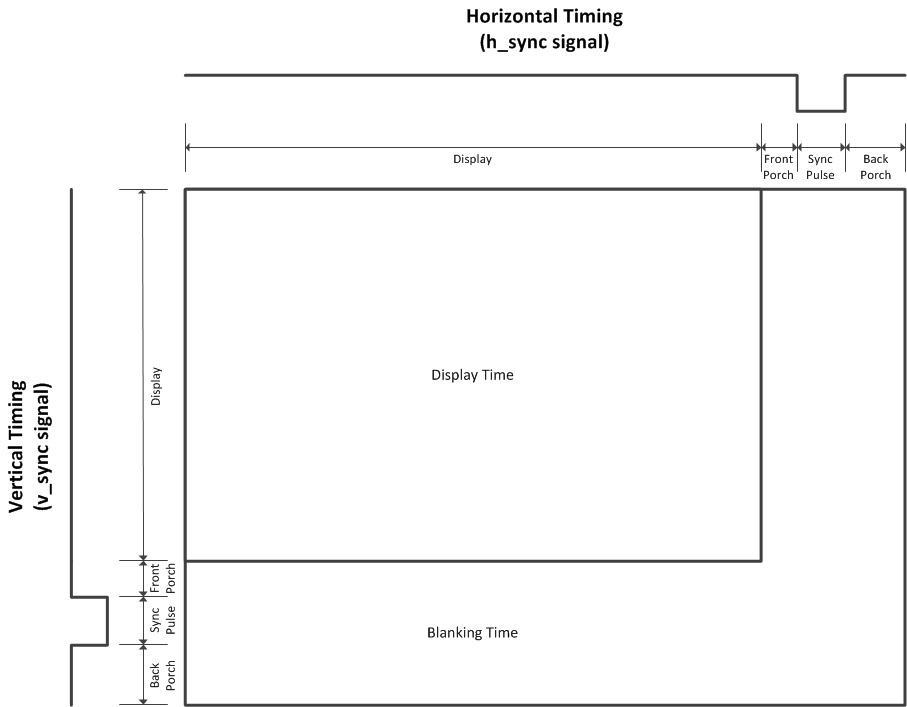
## Controlador do Jogo

O controlador de jogo é o bloco que controla toda a lógica de jogo, porém, este não ficou completo, a função deste bloco é controlar o estado em que o jogo está, isto é, se está em modo run, hold ou game-over, assim como incrementar o contador de pontuação, gerir as vidas detectando as colisões do pacman com os ghosts.

## Controlador de VGA

É no controlador de VGA que existem as únicas diferenças da implementação feita nas duas placas, o sinal de RGB que a placa da *terasic* processa tem oito bits para cada componente de som isto deve-se à existência de uma DAC dedicada para o propósito, enquanto que na placa da *Digilent* a cor é enviada diretamente de um pino da FPGA, tendo cada componente de cor apenas um bit. Embora os oito bits para cada componente permitam maior flexibilidade em termos de cor, foi utilizado o modelo de três bits, tendo sido feita saturação de bits para a Altera.

A resolução escolhida foi de 800 x 600 @ 50 MHz. Para que vá imagem para o monitor é necessário enviar certas informações em determinados instantes para que o ecrã mostre as imagens corretamente, no guia de utilizador da DE2-115 é explicado como funciona o processo de desenho de imagens no monitor através das seguintes imagens:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eixo |  |  | Display Interval |  |
| Horizontal | 1.6 us X\_A | 3.2 us X\_B | 16.2 us X\_C | 0.3 us X\_D |
| Vertical | 3 linhas Y\_A | 21 linhas Y\_B | 600 linhas Y\_C | 1 linhas Y\_D |

O funcionamento do controlador de vga pode-se resumir da seguinte maneira:

Temos dois contadores, um deles incrementa a cada rising\_edge do vga\_clock gerindo assim o sincronismo do sinal h\_sync, o valor máximo do contador vai ter o valor de colunas definidas pela resolução, no meu caso 800. Quando esse contador chega a esse número este vai passar a 0 e incrementar o outro contador que regista o número de linhas desenhado controlando, assim, o v\_sync. É a combinação dos valores destes dois contadores que nos devolve as coordenadas que estão prontas a receber informação de cor.

Quanto a timings interessa referir que existem dois principais estados, o *blanking time* e o intervalo de tempo em que a imagem é mostrada o *DIsplayTime* .

## Controlador do Pacman

O controlador do Pacman recebe os input’s das keys que são processados por uma máquina de estados que em função do input (up, down, rifght, left), faz a alteração da posição do pacman com a frequência de 20Hz. Para além disto este bloco está ligado a uma rom que tem codificadas as restrições de circulação do pacman para cada posição num vetor de quatro bits cada um dedicado para cada direção que o pacman pode circular. (map\_phys)

## Controlador dos Fantasmas

Este controlador é considerada a AI do jogo, mas ao contrário do que possa parecer o modo dos fantasmas se moverem é bastante determinístico, este bloco apenas se limita a calcular a direção que os ghosts vão tomar em função da posição atual do pacman.

Olhando com pormenor é possível reparar que os ghosts ciclicamente perseguem o pacman como de repente estão a ir para o seus cantos do mapa.

Eu tentei implementar o fantasma vermelho por pensar ser o mais fácil de implementar pois este calcula o seu movimento diretamente pelas coordenadas do pacman.

# Conclusão

Com a realização deste projeto reforcei ainda mais o meu gosto pela programação em VHDL e a implentação do mesmo em FPGA’s, penso que é uma linguagem de descrição de hardware que se aprende com alguma facilidade.

Ao implementar o projeto em duas placas entendi o quão abrangente é a linguagem pois independente da marca consegui implementar o jogo sem qualquer tipo de problemas, embora preferindo as ferramentas da Xillinx, quer pela sua rapidez e simplicidade de utilização. É claro que o facto de as placas terem quase as mesmas especificações ajudou.

Foi um trabalho muito extenso pelo que me ajudou a ganhar um grande à vontade e noções de sistemas digitais e VHDL muito grandes, para além do alcance da cadeira.

Embora não tenha concluído o pac-man em si, as bases estão lá e mais tarde poderá até ser possível terminá-lo.