Relatório - Fractal

Fernando Henrique Ratusznei Caetano

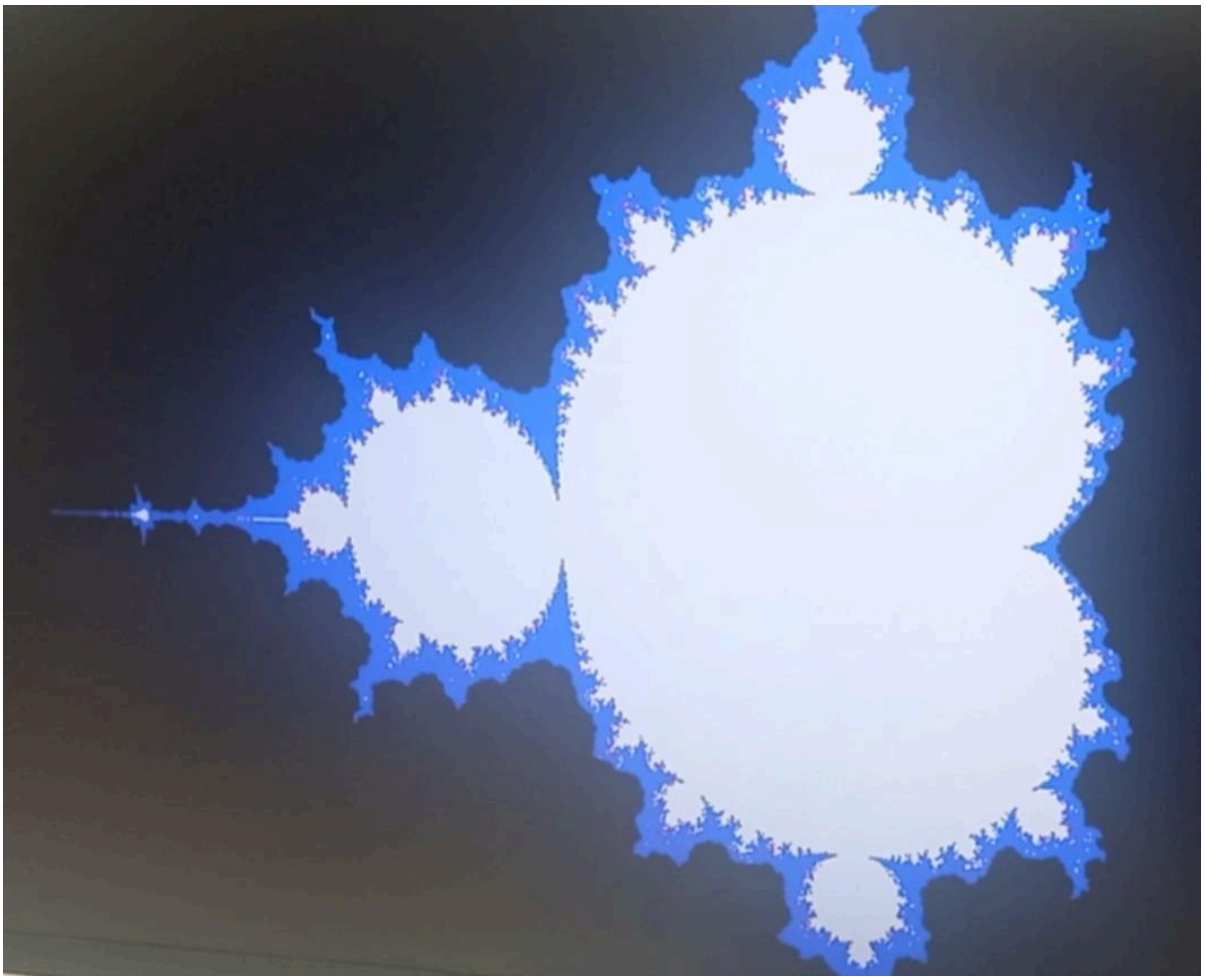

Fig 1. Foto no Conjunto de Mandelbrot, gerado pela placa DE10-Lite.

Descrição do problema:

O Conjunto de Mandelbrot é definido como o conjunto de todos os pontos c no plano convexo, tal que para sucessivas interações com z inicial zero, a função de recorrência a seguir não diverge:

$$z_{n+1} = z_n^2 + c.$$

Podemos desenhar o fractal na tela contando o número de iterações i para que z atinja uma certa magnitude a partir da qual podemos ter certeza que a relação diverge. Cada i então é associada a uma cor para ser desenhada na tela.

Blocos de ponto flutuante:

Foram utilizados dois blocos de ponto flutuante da biblioteca de IP para implementação, um para realizar a adição e outro para realizar a multiplicação de números complexos. Esses blocos utilizam uma pipeline interna, adicionando uma latência de 11 ciclos para adição e 19 ciclos para a multiplicação.

A captura de tela a seguir apresenta a tela de configuração do multiplicador. É possível configurar o pipeline, variando o número de estágios e frequência de operação. Foi escolhida uma frequência de operação de 200MHz e pipeline de 19 ciclos pois essa configuração apresenta a menor latência – considerando que a latência segue a fórmula: período de clock * número de estágio. Foram encontrados alguns problemas de instabilidade devido ao uso dessa alta frequência de clock. Dentre eles: valores imprevisíveis visualizados no *signal tapper* que desapareceram com redução de frequência; e a sincronização entre os clocks do VGA, em 50MHz, e o clock de aritmética, em 200MHz.

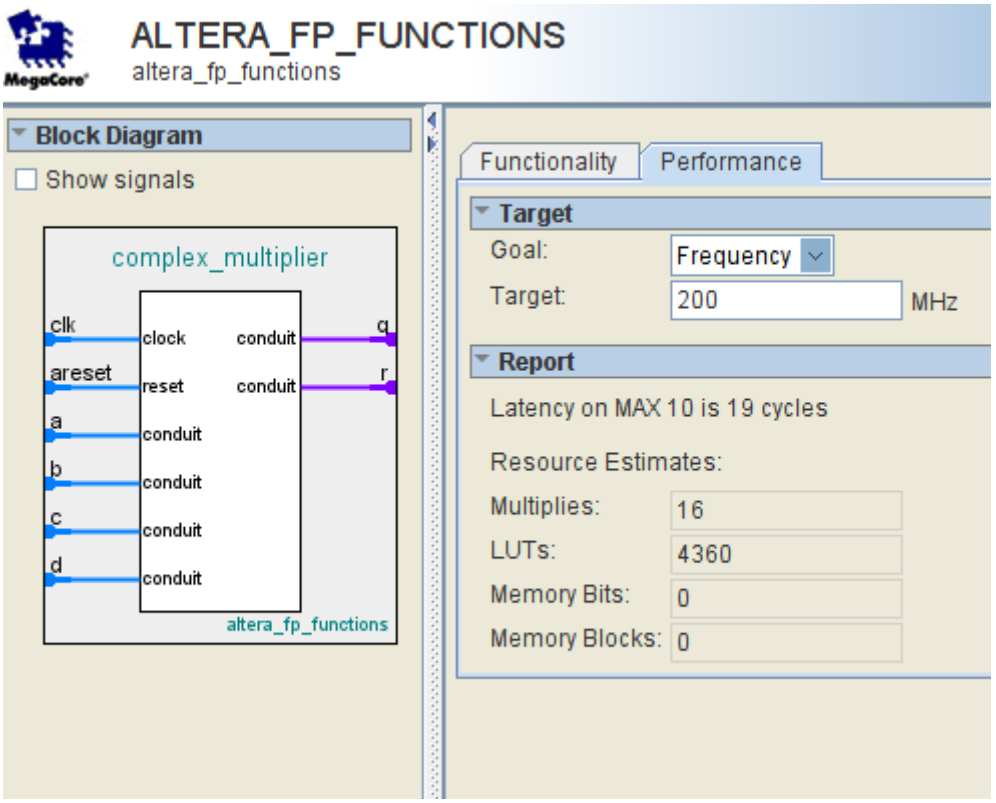

Figura: Configuração do bloco multiplicador complexo

Clocks de aritmética e VGA:

O sistema utiliza dois sinais de clocks diferentes. O singal *clk_vga*, é um sinal de 25MHz que ativa o driver VGA. E o sinal *clk_arith*, é um sinal de 200MHz que ativa as operações aritméticas que implementam a recorrência do fractal.

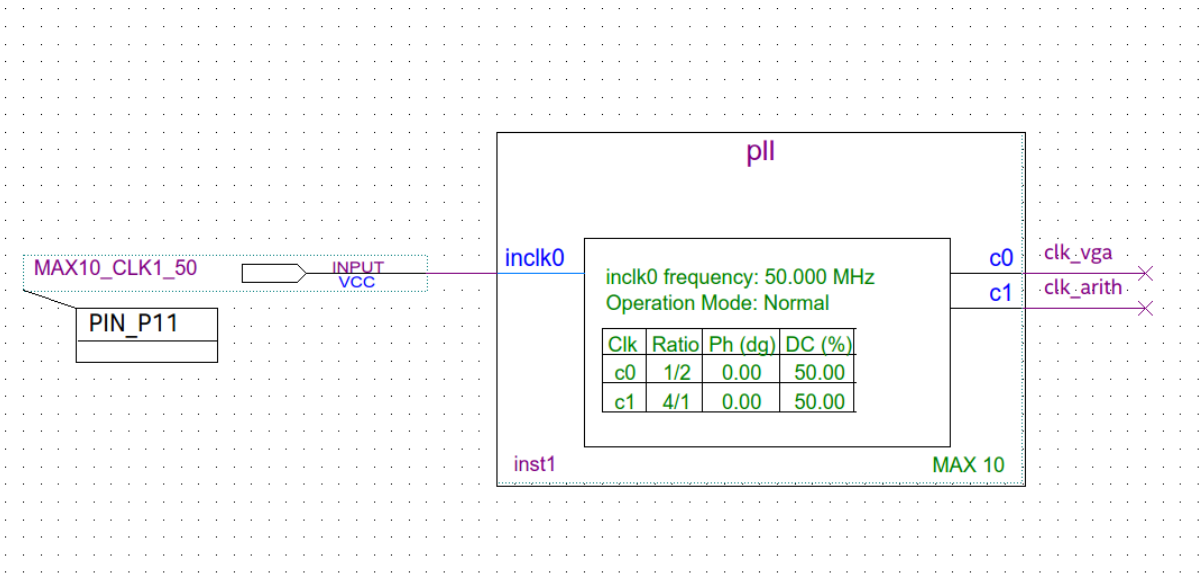

Figura: Geração dos clock pelo uso de uma PLL

Hardware de vídeo:

O hardware de vídeo é apresentado na figura a seguir. O hardware é similar ao utilizado em laboratórios anteriores na disciplina.

Era desejado que os quadros apresentados no monitor fossem quadros completos, isto é, quadros parcialmente desenhados não deveriam ser apresentados. A solução para esse problema foi um buffer duplo, enquanto é escrito em um dos buffers o outro é utilizado para desenho na tela. A implementação do buffer duplo se dá pelo uso de uma segunda memória de vídeo e de um mux para selecionar entre elas.

Um ponto negativo dessa solução é que os requisitos de memória dobram e a placa não possui memória o suficiente para criar dois buffers. Foi necessário compactar a representação de cores, no lugar de gravar um número de 12 bits por posição de memória é gravado um número de 2 bits. Esse número então é indexado em uma tabela de consultas (LUT) para obtermos o valor de 12 bits que representa a cor a ser desenhada.

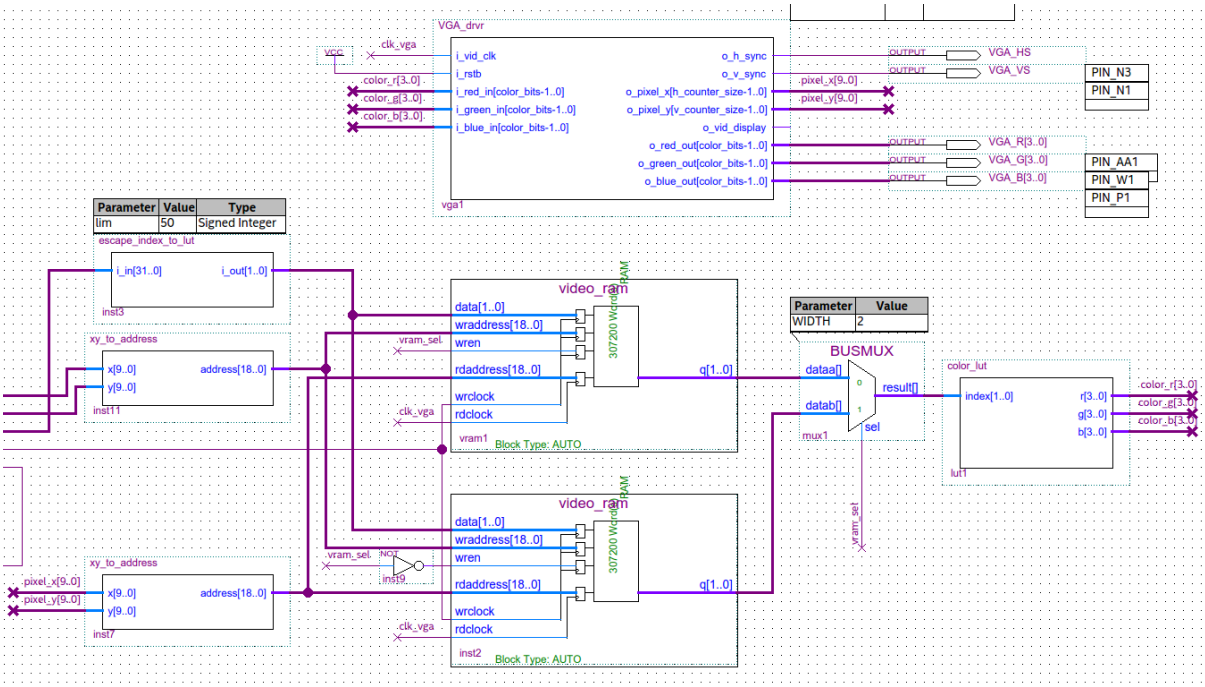

Figura: Hardware de vídeo

Unidade de Controle e cálculo das iterações:

O arquivo *fractal_gen.vhd* descreve o hardware do componente que realiza os cálculos da recorrência. Internamente ele possui uma máquina de estados (a unidade de controle) e o hardware necessário para cálculos (registradores, somadores e multiplicadores). O projeto original utilizaria de uma unidade de controle controlando vários blocos idênticos contendo o hardware para cálculos, porém sincronizar todos esses blocos se provou uma tarefa complicada e o hardware de cálculos e a unidade de controle foram combinados em uma única entidade.

É implementada uma máquina de estados com os estados descritos a seguir:

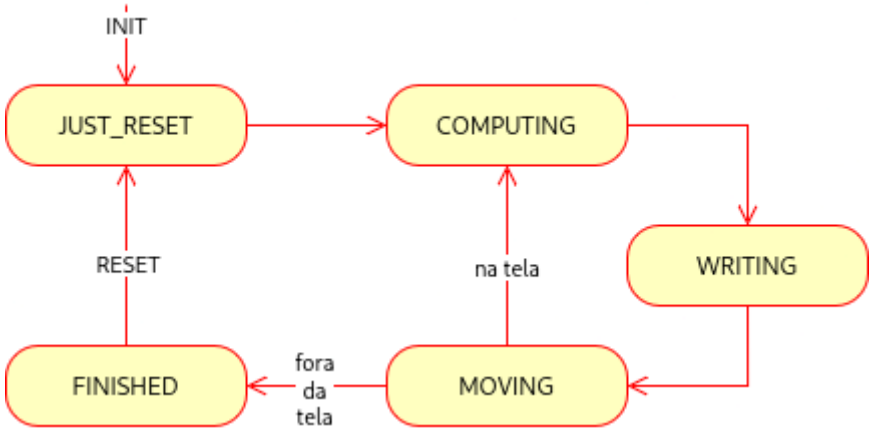

Figura: Máquina de estados da unidade de controle

- **JUST_RESET**: Inicializa registradores e preenche os estágios das pipelines dos blocos de ponto flutuante. Esse estado sempre transiciona para **COMPUTING;**
- **COMPUTING**: Implementa a recorrência e preenche estágios das pipelines dos blocos de ponto flutuante. Esse estado sempre transiciona para **WRITING.** Quando ocorre uma transição saindo desse estado é garantido que o resultado na saída do bloco é válido;
- **WRITING:** Pulsa um sinal no clock de escrita da memória. Esse estado sempre transiciona para **MOVING;**
- **MOVING:** Incrementa registradores que representam a posição no plano complexo e na tela. Se é última posição na tela, transiciona para **FINISHED;** se é final de linha, zera os registradores da parte real e de coluna, incrementa os registradores da parte imaginária e de linha, e transiciona para **COMPUTING**; Se é meio de linha, incrementa os registradores da parte real e de coluna, e transiciona para **COMPUTING.**
- **FINISHED:** Seta o bit *done* e permanece nesse estado até sinal de reset.

Durante o estado **COMPUTING**, a recorrência é recalculada até que se possa ter certeza de que ela irá divergir, e também é imposta uma quantidade máxima de iterações. Quando o valor absoluto de z é maior ou igual a 2 podemos ter certeza que ele diverge.

Na implementação essa verificação foi simplificada para se z está fora de um quadrado de lado 4 centrado na origem, ou seja, se o valor absoluto de parte real for maior ou igual a 2, ou o valor absoluto da parte imaginária for maior ou igual a 2. Para números no padrão IEEE 754 basta verificarmos o valor do expoente. Se o expoente for maior ou igual a 1 podemos ter certeza que o valor codificado é maior ou igual a 2. Os valores bem comportados do padrão podem ser calculados a partir da representação binária pela fórmula:

$$N = (-1)^{BIT\ SINAL} * 2^{BYTE\ DO\ EXPOENTE-127} * 1.BITS\ DA\ MANTISSA_2$$

Quando o byte do expoente for maior ou igual a 128, o expoente será maior ou igual a 1. Como na codificação binária todos os números maiores ou iguais a 128 possuem o bit mais significativo setado, basta verificarmos apenas esse bit para verificarmos que o valor diverge.

Sincronização dos clocks:

O único momento em que os dois clocks operam em um mesmo bloco é durante a troca do buffer duplo. Durante esse momento o bloco *fractal_gen* recebe um sinal de reset assíncrono, o flip flop T que armazena o bit seletor do buffer duplo recebe um sinal de clock e o gerador de posições recebe um clock passando para a próxima posição na animação.

A porta E do circuito na figura a seguir garante que a troca de buffers apenas ocorra quando *fractal_gen* entrou no estado **FINISHED** e o driver VGA acabou de terminar de desenhar um quadro.

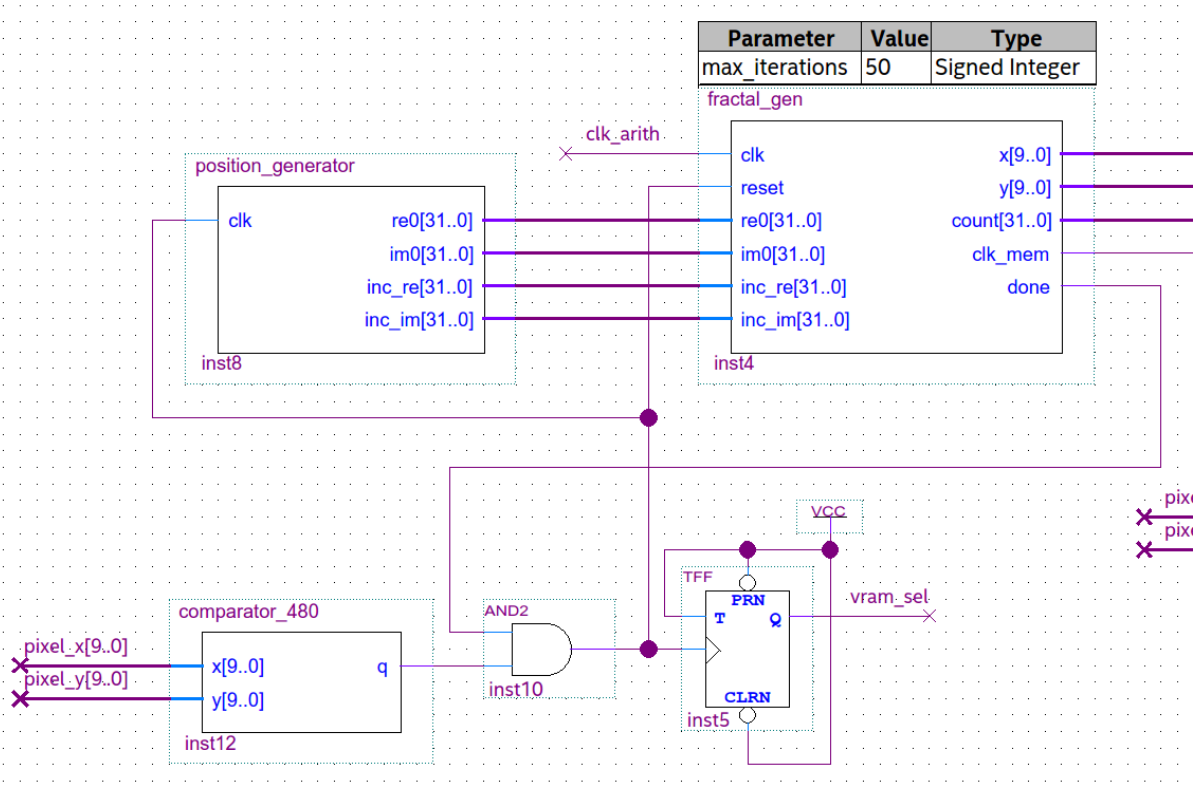

Figura: Circuito que realiza a troca do buffer duplo

Conclusões

Durante o desenvolvimento do foi utilizada uma abordagem utilizando das várias ferramentas estudadas durante a disciplina. VHDL e esquemáticos, que foram estudados em outras disciplinas, foram as principais técnicas utilizadas. Porém o uso da biblioteca de IP – estudada durante o semestre – com a PLL e os blocos de ponto flutuante foram essenciais para o desenvolvimento. Também foram reaproveitados códigos e blocos funcionais, como o driver de VGA.

As maiores dificuldades encontradas surgiram durante a sincronização da lógica de controle com a aritmética dos cálculos, onde precisavam ser levados em conta fatores como o tempo variável para computar cada ponto e os clocks necessários para preencher os pipelines.

Em futuros trabalhos, o sistema ainda pode ser expandido para realizar cálculos com outros fractais ou implementar a paralelização dos cálculos.