

Batalha Naval

Projeto - MC613

Felipe Pessina e Rafael Figueiredo Prudencio

7 de Maio de 2018

Conteúdo

1	Diagrama de Blocos	3
2	Descrição Funcional	4
2.1	Interfaces UART e USB	4
2.2	File	5
2.3	Memória	5
2.4	Desenhar Mapa	5
2.5	Input/Output	6
2.5.1	Controle Mouse	6
2.5.2	Controle VGA	6
2.6	Jogador	6
3	Referências	6

1 Diagrama de Blocos

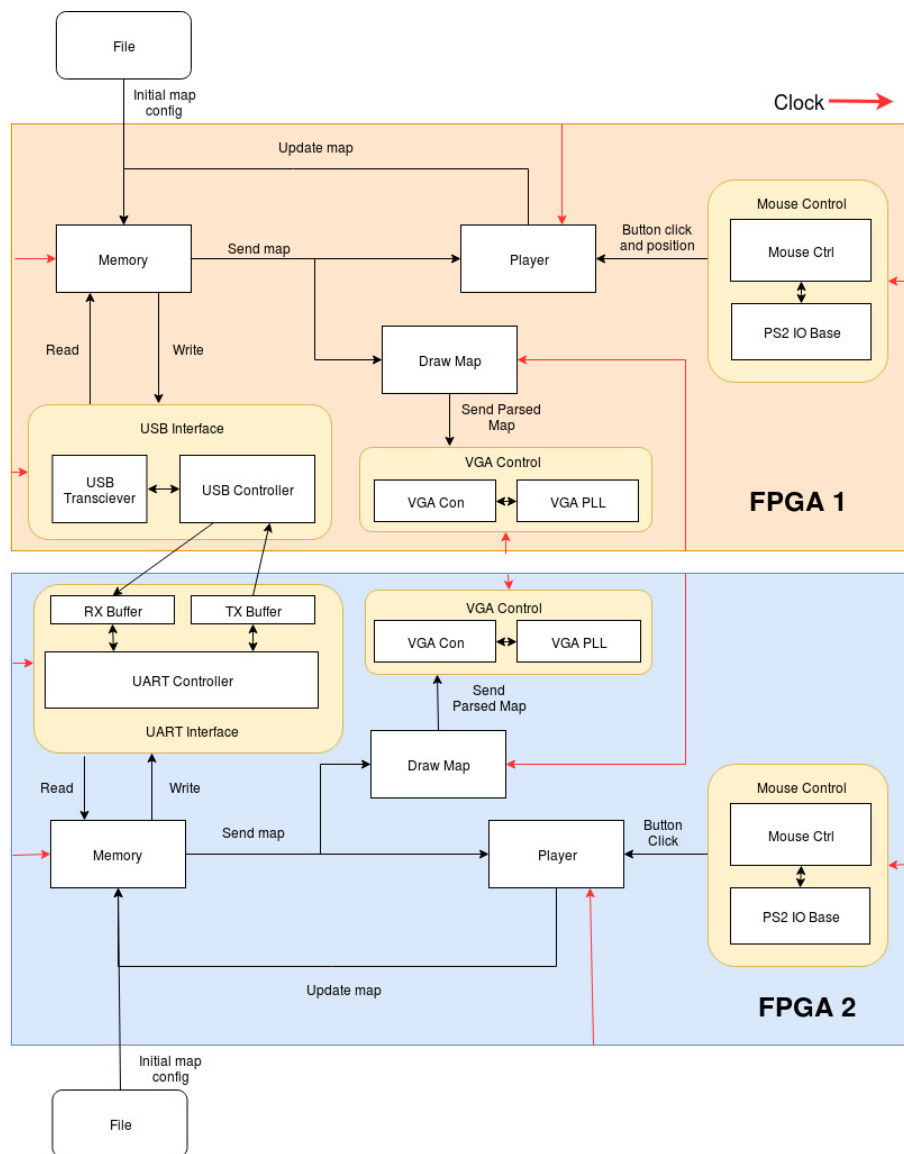


Figura 1: Diagrama de blocos de alto nível para uma implementação de batalha naval.

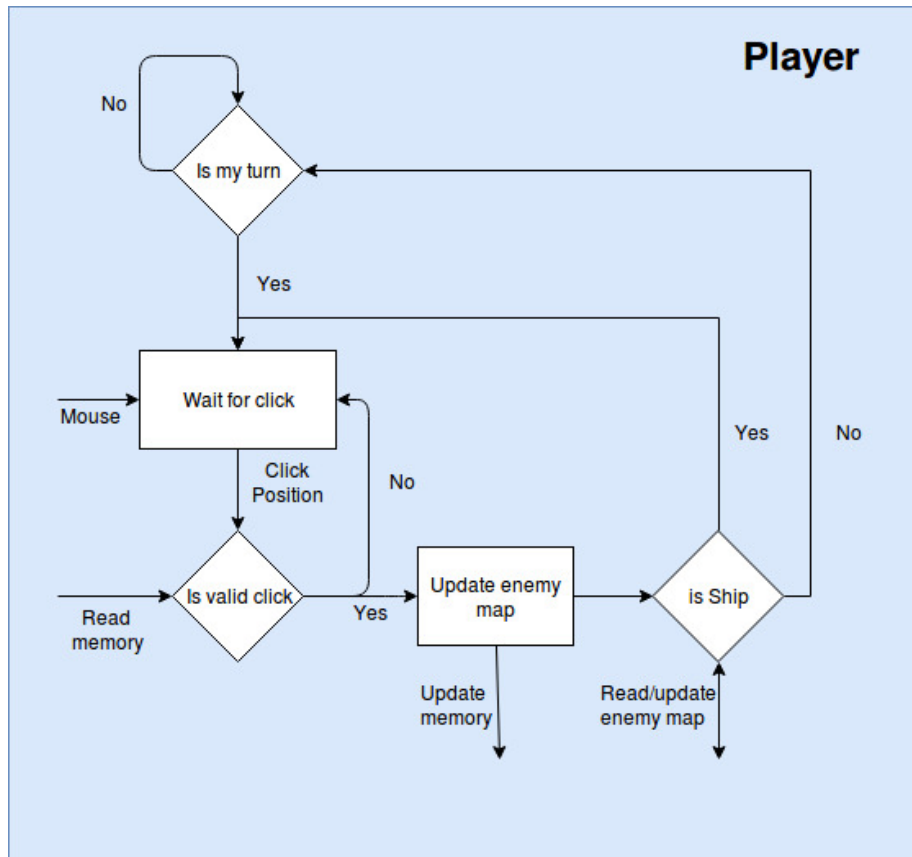


Figura 2: Máquina de estados para a entidade Player.

2 Descrição Funcional

2.1 Interfaces UART e USB

A FPGA1 realiza a transferência de dados para a FPGA2 através da interface USB e, por sua vez, a FPGA2 realiza a transferência de dados para a FPGA1 através da interface UART. A placa Cyclone V SoC da Altera possui uma porta USB-UART, em particular, um conector USB com uma ponte USB-UART para um terminal RS-232 [1]. O UART (Universal Asynchronous Receiver/Transmitter) e o USB (Universal Serial Bus) são duas interfaces de comunicação serial. A universalidade das duas interfaces advém do suporte a inúmeros protocolos seriais. A interface UART implica em no mínimo uma linha TX e RX, que envia uma stream de dados seriais e recebe uma stream de dados seriais, respectivamente [2]. Junto com o padrão RS-232 (Recommended Standard 232), que define os nomes dos sinais, propósito e níveis de voltagem entre

dois dispositivos, a interface UART permite enviar e receber dados. A interface USB é mais moderna e complexa, mas através de um adaptador permite a comunicação com a interface UART.

No diagrama da Figura 1, as linhas vermelhas são os clocks da placa usados como controle por meio de uma entidade que divide o clock na frequência necessária (e.g. `clk_div`). Para realizar a comunicação entre as duas placas será necessário tomar um cuidado especial com a sincronia dos clocks de cada interface serial.

2.2 File

O bloco File da Figura 1 representa um arquivo com a configuração inicial do mapa de cada jogador e que deve ser carregado na memória estaticamente, ou seja, em tempo de compilação.

2.3 Memória

Para cada placa FPGA, há uma instância da entidade Memory da Figura 1. Cada entidade armazena os dois mapas do arquivo na memória e permite com que seja feita a leitura e a escrita na memória. Uma escrita na memória deve trigger um envio de dados para a outra placa, para que ambas tenham uma versão atualizada dos mapas. Na FPGA1 isso será feito através da interface USB, e na FPGA2 isso pode ser feito pela interface UART.

Serão mantidos dois mapas, um para cada jogador. Cada mapa é representado por uma matriz que terá as informações referentes ao estado de cada posição do mapa. Os estados possíveis são: navio, água, navio descoberto e água descoberta. Dessa forma, um jogador pode ler todos os estados do seu mapa e mantê-lo na memória com todas as posições que já foram atacadas pelo oponente. Além disso, cada jogador também deve ler do mapa do oponente os estados: navio descoberto e água descoberta para manter um mapa na memória do que ele já sabe a respeito do oponente. As duas placas podem solicitar uma leitura ou realizar uma escrita para atualizar o estado dos mapas na respectiva memória, ou enviar da sua memória o mapa do jogo para o arquivo.

2.4 Desenhar Mapa

A entidade Draw Map da Figura 1 é responsável pela leitura do mapa do jogo e escrita dele na memória do monitor. São necessárias duas dessas entidades, uma para cada placa que deve atualizar o monitor de cada jogador. Esta entidade tem como entrada os mapas da memória e como saída os valores dos pixels com os endereços onde devem ser gravados na memória do monitor, comunicando-se diretamente com o controle de VGA. Os valores dos pixels serão determinados a partir do estado de cada posição nos mapas mantidos na memória. Um jogador tem acesso a todos os estados do seu mapa, mas apenas aos estados navio descoberto e água descoberta do mapa do seu oponente.

2.5 Input/Output

2.5.1 Controle Mouse

Cada placa FPGA terá um mouse como dispositivo de entrada. A entidade Play receberá a posição do mouse sempre que houver um click através do bloco de controle do mouse. Para esta parte, iremos reaproveitar o código disponibilizado para o laboratório 7.

2.5.2 Controle VGA

A escrita na memória do monitor é feita através do controlador de VGA, usando as mesmas entidades do laboratório 8. Cada jogador também precisa de um monitor que será atualizado através dessa entidade e manterá uma imagem do estado do jogo. A escrita na memória do monitor será feita pela entidade Draw Map.

2.6 Jogador

A entidade Player na Figura 1 é responsável pelo tratamento das jogadas. Um diagrama mais detalhado da entidade Player está indicado na Figura 2. Cada player tem acesso às informações do seu mouse. O turno de cada jogador será armazenado na memória das duas FPGAs. A entidade Player é uma máquina de estados que primeiro deve verificar se é sua vez para então esperar por clicks do jogador. Após um click, deve-se verificar se é uma jogada válida. Caso não seja, devemos continuar esperando por clicks. Caso seja, devemos atualizar o mapa na memória e verificar se acertamos um navio inimigo ou não. Em caso de acerto, podemos realizar uma nova jogada. Caso contrário, devemos escrever no mapa de memória que é o turno do jogador oponente e passarmos para o estado de esperar nosso turno.

3 Referências

- [1] Altera, “Cyclone v soc development board.” https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/manual/rm_cv_soc_dev_board.pdf, Dec 2017. Acessado em 2018-11-08.
- [2] A. Davis, “Difference between uart and rs-232?.” <https://electronics.stackexchange.com/questions/110478/difference-between-uart-and-rs-232>, May 2014. Acessado em 2018-11-08.