



UNIVERSIDADE FEDERAL DA FRONTEIRA SUL - UFFS
CAMPUS CHAPECÓ
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
SISTEMAS DIGITAIS

ANA CLARA BRUSAMARELLO BARBOSA, 2121101054
RICHARD FACIN SOUZA, 2211101002

TRABALHO FINAL
JOGO DE BATALHA NAVAL

CHAPECÓ,
2023

1.0 INTRODUÇÃO

Este relatório descreve o desenvolvimento e a implementação de um jogo de batalha naval simulado. O objetivo do jogo é afundar os navios do oponente, posicionando estrategicamente os navios e realizando disparos. O jogo ocorre em uma matriz 4x4, representando o "mar", onde cada posição possui um código associado.

Inicialmente, o primeiro jogador deve posicionar seus três navios, sendo que dois deles ocupam uma única posição, e o terceiro ocupa duas posições consecutivas, verticalmente ou horizontalmente. Para posicionar os navios, o dono utiliza quatro chaves (SW) da FPGA (uma por vez), onde são informadas as posições de cada navio de forma codificada, seguindo uma tabela de codificação própria.

Uma vez que os navios estão posicionados, o segundo jogador tem quatro chaves para informar a posição do alvo e uma entrada para solicitar um disparo em direção aos navios. Diferentemente das chaves do primeiro jogador, as chaves do segundo jogador não possuem codificação associada e informam diretamente a posição do disparo (linha e coluna). O segundo jogador tem um total de seis chances de disparo. Se, no último disparo, o segundo jogador afundar um navio, ele ganha uma chance extra, continuando assim sucessivamente. Caso, ao final das tentativas, o segundo jogador não tenha afundado todos os navios, o primeiro jogador vence a partida. O número de disparos realizados deve ser apresentado no display presente na placa, em ordem crescente ou decrescente.

Neste relatório, serão detalhadas as etapas de implementação do jogo de batalha naval simulado, incluindo a descrição das estratégias utilizadas, a configuração das chaves e dos componentes da FPGA, bem como a lógica de controle do jogo e a exibição dos resultados no display.

2.0 IMPLEMENTAÇÃO

O jogo começa com o estado COMECO, onde as configurações iniciais são definidas.

O jogador deve posicionar três navios em diferentes coordenadas usando os interruptores deslizantes. Cada navio é representado por um código binário de 4 bits, convertido em um código de exibição nos displays de sete segmentos.

Cada navio é posicionado em cada estado feito para o posicionamento, N1, N2 e N3, onde cada um verifica se as coordenadas dos navios estão corretas e não repetidas. Caso as coordenadas do 2º navio seja igual à do 1º, o jogo avisa um erro e o jogador poderá posicioná-lo novamente.

Quando as coordenadas dos navios estão corretas, o jogo avança para o estado INICIOJOGO. Neste estado, as variáveis de controle são reinicializadas para começar a jogada.

O jogo continua para o estado HEX, onde os displays de sete segmentos exibem um código específico indicando que é a vez do jogador realizar um tiro, tendo 6 chances e mostra quantos barcos ele acertou.

No estado J, o jogador seleciona as coordenadas do tiro nos interruptores deslizantes. O código verifica se o tiro acertou algum dos navios e atualiza as variáveis de controle correspondentes.

Dependendo dos acertos e do número de jogadas realizadas, o jogo progride para diferentes verificações para atualizar as saídas correspondentes, como coordenadas dos tiros, status do jogo e feedback visual nos LEDs verdes que mostram exatamente qual dos barcos foi acertado.

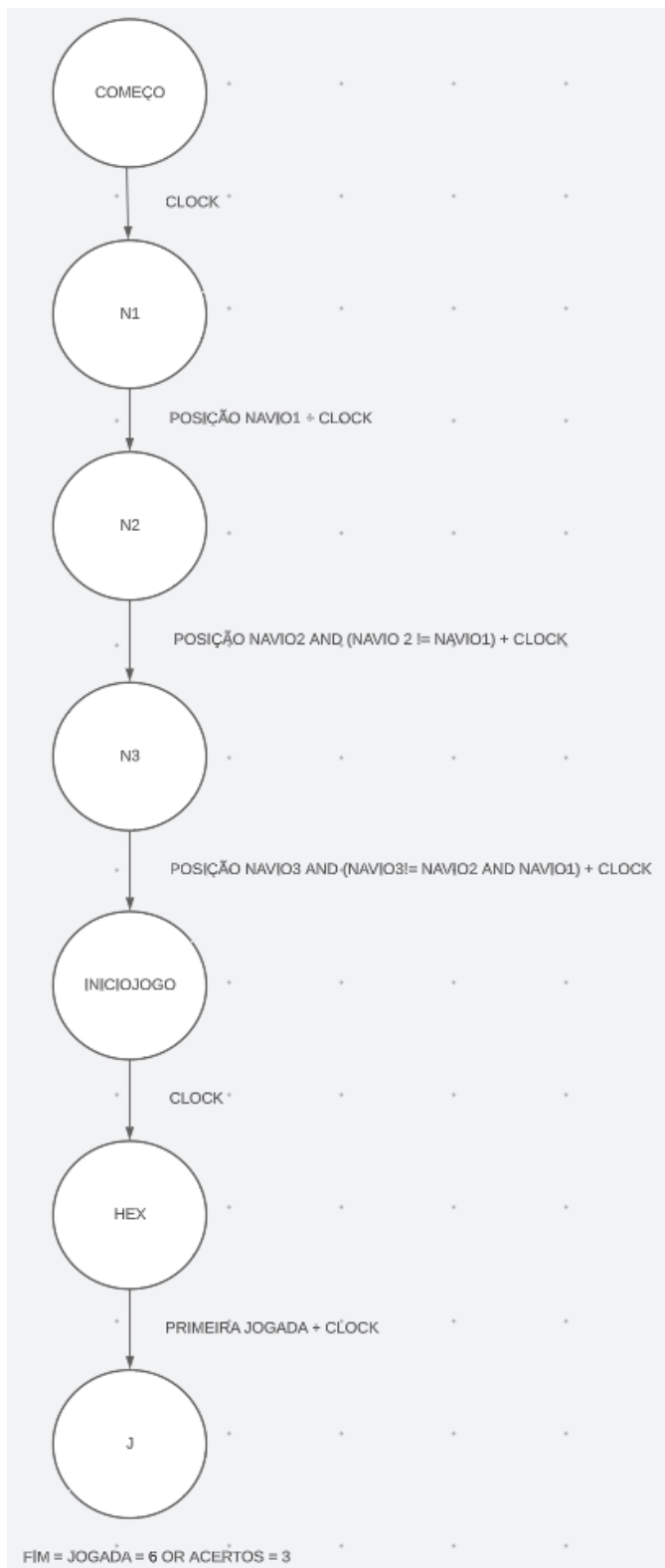
Caso o jogador chegue a 6º jogada e acerta um navio ele ganha uma jogada extra, tendo o limite de 1 erro. O jogo acaba quando todos os navios são acertados ou quando as chances acabam, no primeiro caso todos os LEDs verdes acendem, já no segundo todos os vermelhos acendem.

Observação sobre o Terceiro Navio:

Uma observação importante é que, de acordo com a análise do código VHDL fornecido, o terceiro navio não foi implementado com as duas posições solicitadas. Em vez disso, o terceiro navio possui apenas uma posição definida pelos interruptores deslizantes.

Diagrama de Estados:

A seguir, apresentamos o diagrama de estados que descreve o fluxo de funcionamento do jogo Batalha Naval implementado em VHDL:



O diagrama de estados ilustra as transições entre os diferentes estados do jogo, mostrando como o jogo progride à medida que o jogador realiza as ações e as condições são satisfeitas.

Observações

O código apresenta um “problema” que quando está no estado J o contador de acertos

e jogadas apenas atualiza no ciclo de Clock seguinte, podendo ser ajustado criando um variável auxiliar, assim não dependendo da transição de clock para o display mostrar qual jogada está e quantos acertos tiveram.

3.0 CONCLUSÃO

Nosso código é todo baseado numa estrutura de estados, onde cada estado representa uma etapa diferente do jogo. Primeiro, o primeiro jogador posiciona seus navios usando as chaves da placa. Seguindo a premissa do trabalho, codificamos as posições dos navios, visto que as posições são ‘aleatórias’ e não seguem a ordem correta que indica na linha e coluna.

Após o posicionamento dos navios, o segundo jogador usa as chaves da placa para indicar as coordenadas de onde quer atirar. Comparamos essas coordenadas com as posições dos navios do primeiro jogador e contamos os acertos e erros. Essas informações são mostradas nos displays da placa.

Ao longo do jogo, o código implementa lógica para controlar as transições entre os estados, exibindo os resultados corretamente nos displays e LEDs da placa. Além disso, são consideradas condições especiais, como a possibilidade de o adversário receber uma chance extra caso afunde um navio no último disparo.

Em conclusão, o projeto atingiu seus objetivos de simular um jogo de batalha naval e demonstrou a capacidade de programação para o controle de uma placa FPGA.