

# CAMPUS CHAPECÓ

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**Trabalho Batalha Naval**

**HENRIQUE RIBEIRO RODRIGUES**

**JOÃO VITOR BERNARDON SCHWEIKART**

**RAFAEL CORDEIRO DE ALMEIDA**

**CHAPECÓ 14/12/2023**

# Apresentação

Este relatório descreve o processo de construção de um jogo de batalha naval utilizando codificação de máquina de estados. Basicamente, o jogo consiste na presença de dois navios no “mar”, sendo esse uma matriz codificada de 4x4 posições. Para posicionar os navios (sendo que um deles ocupa apenas uma posição e outro duas posições – verticais ou horizontais), o “dono do mar” utiliza 4 chaves (SW) referente a linha e coluna desejada (de forma codificada), informando assim a posição do seu navio no mar (cada posição da matriz possui um código associado). Da mesma forma, o outro jogador informa onde efetuará o disparo para “derrubar” um dos navios do inimigo através da utilização das 4 chaves (SW) referentes a linha e coluna (desta vez não-codificado) onde se imagina em que um dos navios adversários está localizado.

A partir do código VHDL executado sobre a FPGA, o sistema criado “flutua” sobre 6 diferentes estados, onde cada um deles realiza uma ação específica de forma que o jogo percorra de forma saudável até que alguma das condições pré-definidas retorne o jogo para o estado inicial. Cada estado e a lógica utilizada é explicada no tópico 2 deste relatório.

# Relato da Estratégia Aplicada:

Para a resolução, a lógica do jogo foi dividida em 3 arquivos/componentes diferentes, sendo o arquivo principal o “batalha.vhdl”, que percorre os outros dois arquivos “display.vhdl” e “codificador.vhdl” através da chamada desses componentes na sua arquitetura.

No componente “display” é onde ocorre a validação da quantidade de jogadas para, então, ativar o display de 7 segmentos para mostrar a quantidade restante de jogadas. No compontente “codificador” ocorre a descodificação da entrada que o jogador define a posição de seus navios no mar, para que seja possível posteriormente fazer a comparação com a posição que for definida os tiros pelo outro jogador. Essa descodificação foi definida através da codificação do campo selecionada (tópico 3) e através das simplificações por Karnaught (tópico 4).

No componente principal “batalha” é onde o jogo de fato ocorre. Seguindo a ordem do diagrama de estados descrito no tópico 5 desse relatório, o estado inicial S é utilizado apenas para iniciar/reiniciar as posições dos barcos, led’s e outras variáveis que envolvem o processo do jogo em caso de reset ou clock inicial.

Para posicionarmos as 3 posições de barcos (1 posição do barco menor + 2 posições do barco menor), foi necessário criar um estado para a inserção de cada posição, visto que não haveria chaves suficientes para adicionar todos navios ao mesmo tempo. Portanto, os 3 próximos estados (A, C e C2) servem para definir a posição de cada um dos barcos separadamente, como explicado anteriormente.

Para o estado C2 em específico, fez-se necessário construir uma validação do local onde a segunda posição do barco maior seria salva, visto que o próprio jogador pode escolher se o barco maior ficará posto na horizontal ou vertical através da seleção da chave sw(8). Nessa etapa, o sistema automaticamente posiciona a segunda posição do barco maior abaixo da primeira posição (em caso de vertical) ou coloca ao lado direito da primeira posição (em caso de horizontal). Caso o sistema identifique que a tentativa de inserção não é valida (exemplo: a primeira posição do navio maior já está na borda da direita do mar, então não é possível inserir na direita em caso horizontal), o sistema salva a segunda posição no lado contrário da condição explicada anteriormente. Para cada navio salvo no mar, um led vermelho é aceso.

Após a etapa de salvamento das posições dos navios a partir dos 3 estados anteriores, chegamos aos estados D e E, que é onde ocorre o disparo dos tiros e suas devidas validações. No estado D, basicamente ocorre a comparação da entrada definida pelo usuário que está efetuando os tiros com a posição descodificada de cada um dos barcos. Em caso de acerto do tiro, led’s verdes são acesos e, em caso de queda de algum dos navios, o led vermelho correspondente ao navio é apagado.

Dentro do estado E, é feita a verificação se todos os navios já foram acertados ou se a quantidade de jogadas restantes é igual a zero. Caso alguma dessas afirmações forem verdadeiras, o jogo é resetado e retorna para o estado S. Caso contrário, retorna ao estado D para continuar o disparo dos tiros.

# Codificação do Campo e Tabela Verdade

Foi utilizada a matriz codificada número 22 para a construção do circuito:

|  |  |  |  |
| --- | --- | --- | --- |
| 1001 | 0110 | 1111 | 1101 |
| 0000 | 1011 | 0001 | 0011 |
| 1110 | 0100 | 0010 | 1010 |
| 0111 | 1000 | 1100 | 0101 |

Fonte: descrição do trabalho;

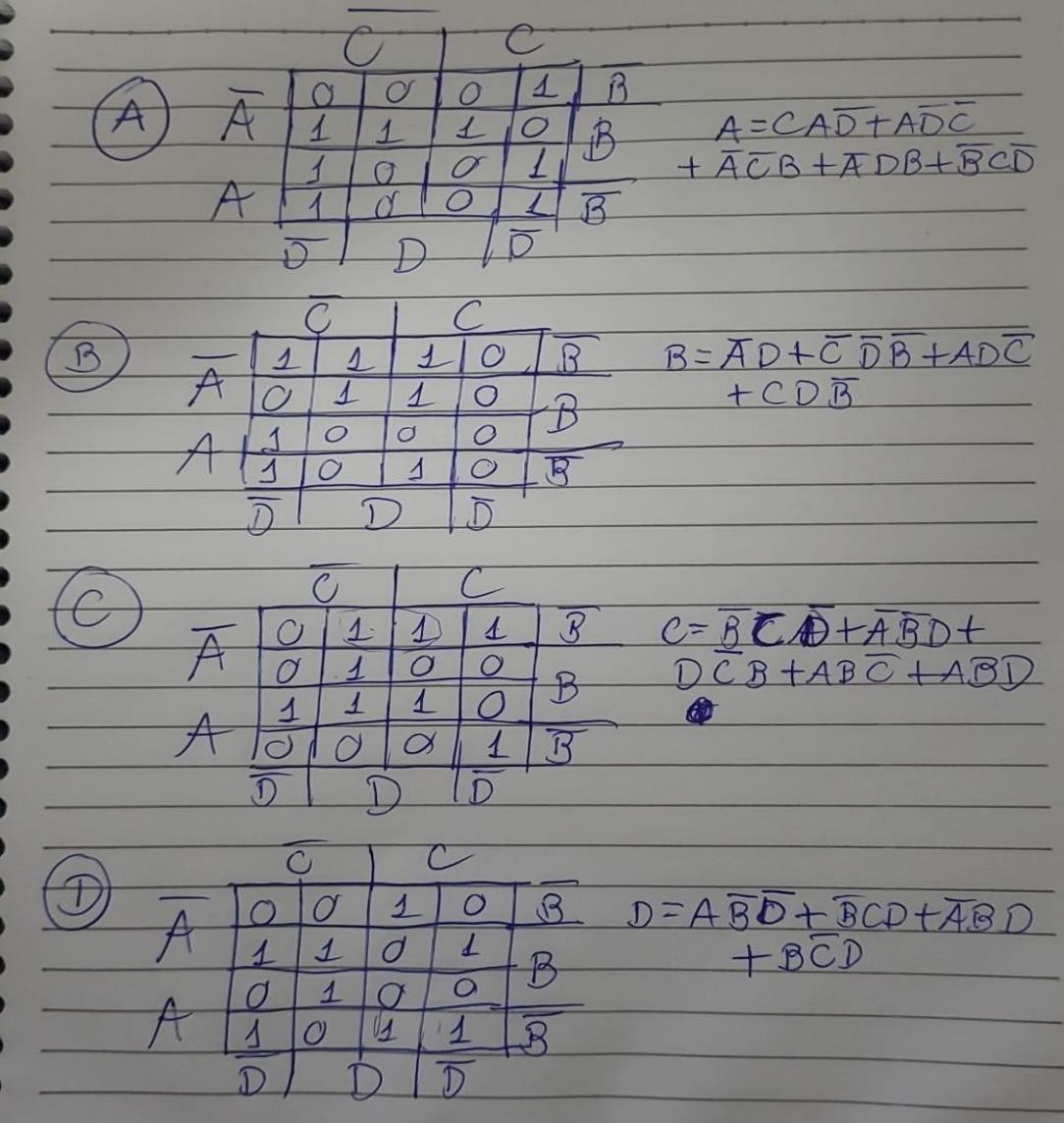
A partir da codificação do campo acima, foi construída a seguinte tabela verdade, transformando o código da posição dos navios em posição da linha e coluna onde o mesmo código se encontra:

|  |  |
| --- | --- |
| **ENTRADA (POSIÇÃO DO**  **NAVIO CODIFICADO)** | **A B C D (POSIÇÃO DO NAVIO EM LINHA E COLUNA DESCODIFICADO)** |
| 0 0 0 0 | 0 1 0 0 |
| 0 0 0 1 | 0 1 1 0 |
| 0 0 1 0 | 1 0 1 0 |
| 0 0 1 1 | 0 1 1 1 |
| 0 1 0 0 | 1 0 0 1 |
| 0 1 0 1 | 1 1 1 1 |
| 0 1 1 0 | 0 0 0 1 |
| 0 1 1 1 | 1 1 0 0 |
| 1 0 0 0 | 1 1 0 1 |
| 1 0 0 1 | 0 0 0 0 |
| 1 0 1 0 | 1 0 1 1 |
| 1 0 1 1 | 0 1 0 1 |
| 1 1 0 0 | 1 1 1 0 |
| 1 1 0 1 | 0 0 1 1 |
| 1 1 1 0 | 1 0 0 0 |
| 1 1 1 1 | 0 0 1 0 |

Fonte: os autores

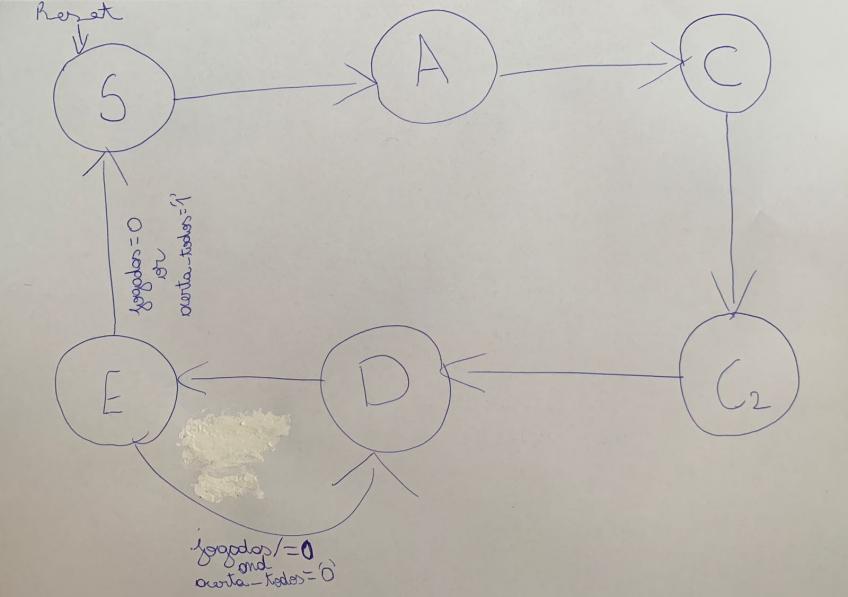
# Simplificações:

Após a construção tabela verdade, foi aplicado Karnaugh em cada uma das 4 saídas obtidas para simplificar o circuito:



Fonte: os autores;

1. **Máquina de Estados**



Fonte: os autores;

**Legenda:**

* O estado S inicializa o jogo limpando os campos;
* Os estados A, C e C2 servem para posicionar os barcos;
* O estado D é onde ocorre o disparo dos tiros;
* O estado E verifica se todos os barcos foram acertados ou se acabaram as jogadas;
* O reset acontece quando o pino sw(9) está ativo.

# Conclusão:

Conclui-se, portanto, que mediante à atividade feita foi possível aplicar muito do conhecimento teórico como a implementação da Máquina de Estados no diagrama e do conhecimento prático no código em VHDL, o qual simula um jogo de Batalha Naval. Dentre as maiores dificuldades, ressalta-se a lógica de codificação das jogadas. Diante disso, a dinâmica permitiu que os conceitos aprendidos fossem aprofundados, a fim de implementar algo do dia a dia, nesse caso, o jogo Batalha Naval.