

Universidade Federal de Santa Catarina  
EEL5105: Circuitos e Técnicas Digitais  
Semestre: 2015/1 – Projeto

## Batalha naval

O projeto final consiste na implementação de um circuito na placa de desenvolvimento *DE2* fazendo uso das estruturas e conhecimentos obtidos durante o curso. O circuito deverá implementar a funcionalidade de um jogo interativo similar ao batalha naval, entre dois jogadores (usuário 1 e usuário 2). O comportamento do jogo está definido a seguir:

- Os usuários definem de forma secreta uma matriz de zeros e uns onde os uns indicam a posição dos barcos. A matriz possui 4 linhas e 14 colunas e, após preenchida, deve ser armazenada em uma memória.
- Os usuários escolhem um nível de dificuldade (velocidade do jogo), com os Switches  $\{SW_{17}, SW_{16}\}$ . O nível de jogo será mostrado no Display HEX2. O Display HEX3 mostra a letra L de ‘*Level*’. A contagem será mostrada nos Displays HEX1 e HEX0.
- Uma vez selecionado o nível do jogo o usuário 1 pressiona INIT (utilizando o botão KEY2) e começa o jogo.
- O usuário deve acertar as posições dos alvos no tempo de jogo usando os Switches  $\{SW_{15}, SW_{14}, \dots, SW_1, SW_0\}$  e o botão KEY3. As coordenadas selecionadas são mostradas nos displays HEX7 e HEX6. Só é permitido um disparo por rodada, por isso uma vez pressionado o botão KEY3, o outro usuário deverá fazer a sua seleção e realizar o seu disparo. Os displays HEX5 e HEX4 apresentam a informação dos turnos, mostrando a letra U de ‘*User*’ no HEX5 e 1 ou 2 no HEX4 dependendo se está jogando o usuário 1 ou 2, respectivamente.
- A contagem de acertos será mostrada nos LEDs verdes  $\{LEDG7, \dots, LEDG0\}$  para o usuário 1 e nos LEDs vermelhos  $\{LEDR7, \dots, LEDR0\}$ , uma vez que na matriz de uns e zeros serão permitidos apenas 8 uns. Caso um dos usuários não realize um disparo no tempo da sua rodada, e a contagem termina (valor 0 é alcançado), começa a rodada do usuário seguinte, e assim sucessivamente até que algum usuário acerte todos os alvos. Nesse momento, o sistema reinicia.
- Um usuário pode a qualquer momento parar o jogo usando o botão KEY3 zerando a contagem de alvos, para assim reiniciar o jogo.

O esquema geral do projeto é mostrado na Figura 1 e inclui cinco blocos diferenciados:

- *Memória*: Duas memórias onde os usuários colocam as matrizes de zeros e uns com os alvos.
- *Temporizador com seletor de nível*: Circuito sequencial encarregado de gerar uma contagem decrescente em segundos de dois dígitos em decimal. O tempo associado a cada rodada é definido por dois switches SW17 e SW16, permitindo 4

possíveis tempos de jogo. Além disso, o temporizador possui um reset assíncrono e botão de início de turno nos botões KEY0 e KEY3, respectivamente. Sempre que o tempo de jogo finaliza, ocorre um reset, ou é realizado um disparo com o botão KEY3, então o usuário é alternado, essa informação é dada pelo bit "USER".

- *Comparador*: Circuito combinacional encarregado de gerar um bit, *MATCH*, que determina se o usuário adivinhou o alvo do inimigo usando os Switches  $\{SW_{15}, \dots, SW_1, SW_0\}$ , onde *MATCH* = '1' se o usuário acerta o valor, e *MATCH* = '0' caso contrário).
- *Contador de pontos*: Circuito sequencial encarregado de acumular os alvos atingidos pelos usuários (número de vezes que *MATCH* = '1').

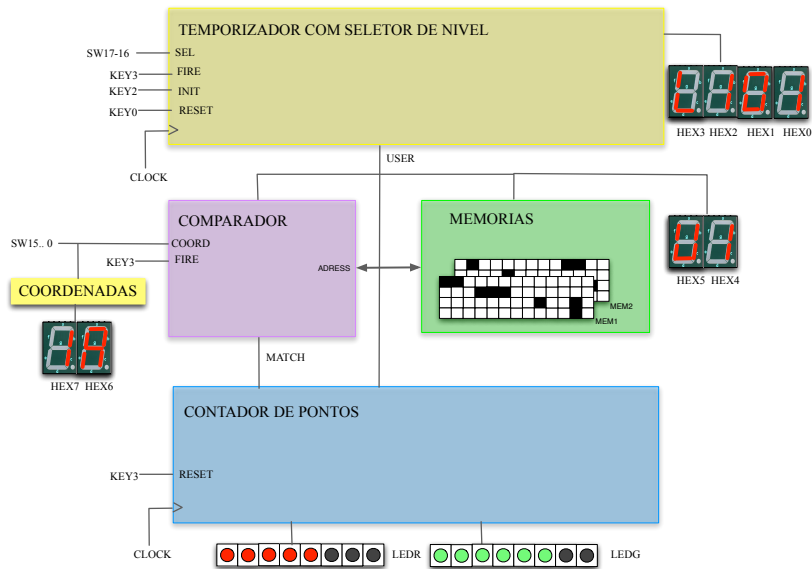


Figura 1: Diagrama de blocos do jogo batalha naval.

**MEMÓRIA** No ambiente Moodle da disciplina é fornecido parte do código de uma memória ROM (ROM.vhd) e um arquivo de texto (planilha.txt) onde os alunos podem anotar a posição dos seus barcos, a qual deve coincidir com a matriz introduzida na ROM. O aluno deve escrever o resto do código da memória usando as dicas fornecidas na aula 10.

Os 4 tipos de barcos que os usuários devem usar são os seguintes: ●●● Porta-aviões (×1). ●● Cruzadores (×2). ● Submarino (×1).

*Dica: Uma vez que as memórias dos usuarios 1 e 2 tenham sido descritas em VHDL, e as matrizes tenham sido introduzidas, usar um MUX 2:1 com USER como entrada de controle.*

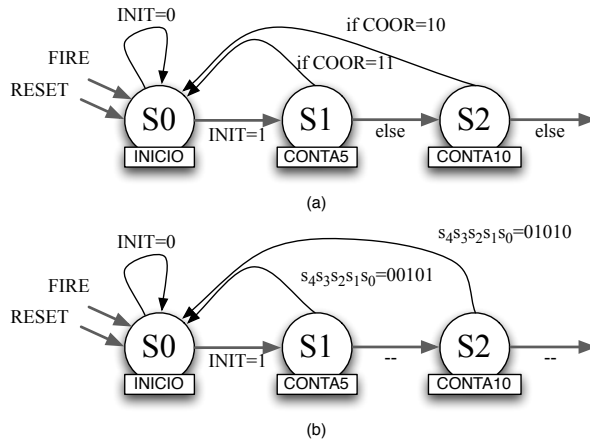


Figura 2: Diagrama de estados parcial do temporizador com seletor de nível, a) para descrição em modo comportamental e b) em modo estrutural.

**TEMPORIZADOR COM SELETOR DE NÍVEL** O temporizador a ser implementado deve permitir 4 tempos de jogo para cada rodada (5, 10, 20 e 30 segundos para os níveis 4, 3, 2 e 1, respectivamente). O temporizador deve ser implementado fazendo uso do fluxo de projeto de máquina de estados, FSM, visto nas aulas teóricas e de laboratórios. Para implementar a máquina de estados com um relógio de um segundo pode ser utilizado o conversor de frequência de 50MHz-1Hz disponibilizado no Moodle, com a entrada ligada ao relógio de 50MHz disponível na placa DE2.

A máquina de estados pode ser implementada de forma estrutural (controlando um somador de 5 bits) ou comportamental usando o comando *process*. No entanto, é fortemente aconselhável usar a abordagem comportamental estudada a partir da aula 7 de laboratório.

Na Figura 2 é fornecida uma **parte** do diagrama de estados a ser implementado, onde a Figura 2(a) pode ser usada como base para uma implementação comportamental, e a Figura 2(b) para uma implementação estrutural (as entradas da máquina de estados  $\{s_4, \dots, s_1, s_0\}$  são os valores de saída do somador síncrono de 5 bits). Em ambos diagramas já estão incluídas as entradas de RESET, INIT e FIRE. O seletor de nível é a parte encarregada de controlar o bloco temporizador. Usando os seletores  $\{SW_{16}, SW_{17}\}$  podemos escolher entre 4 faixas de tempo diferentes. Quanto maior a faixa de tempo, menor o nível de dificuldade do jogo.

*Dica: É importante destacar que o diagrama de estados mostrado não é único, e o aluno possui a liberdade de projetar outro diagrama de estados com a mesma funcionalidade.*

## COMPARADOR

Uma vez escolhidas as coordenadas pelo usuário que está jogando, o comparador deve selecionar o endereço da memória do adversário indicada pelos Switches 15 e 14.

Uma vez selecionada a linha de endereço da memória do adversário, o comparador deve comparar a linha de endereço com as saídas dos Switches 13 a 0. É importante ressaltar que nos  $\{SW_{13}, \dots, SW_1, SW_0\}$  só pode existir um único '1' lógico, o qual indica a posição do disparo.

Por exemplo, se o usuário seleciona  $\{SW_{15}, \dots, SW_1, SW_0\} = \overbrace{10}^2 \overbrace{01000000000000}^C$  (coordenada 2C mostrada nos displays HEX7 e HEX6), o comparador deve ir à terceira linha de endereço da memória do adversário, a qual vamos supor  $\{R_{13}, \dots, R_1, R_0\} = 01000000000001$ , e fazer uma comparação bit a bit de modo a encontrar  $SW_i = R_i = 1$ , para  $0 \leq i \leq 13$ . No exemplo,  $SW_{12} = R_{12} = 1$ , resultando em  $MATCH = 1$  na saída do comparador.

*Dica: O circuito comparador que seleciona a memória e realiza a comparação é puramente combinacional, e deve gerar um bit MATCH que determina se o usuário adivinhou uma posição do alvo, '1' lógico, ou se não adivinhou, '0' lógico.*

## CONTADOR DE PONTOS

Uma vez obtido o bit *MATCH*, o contador de alvos é um circuito que acumula (soma) de forma síncrona o número de vezes que o usuário acerta alvos em cada disparo. O contador de pontos deve ser zerado uma vez pressionado INIT (um novo jogo), ou quando for pressionado o *reset* assíncrono KEY3, ou ainda quando for pressionado o clear.

*Dica: Podem ser usados dois somadores de 3-bits, um para cada jogador, mas levando em consideração que devem ser síncronos com o sinal de relógio.*

### Orientações Gerais:

- Na apresentação, todos os membros do grupo deverão estar presentes.
- O trabalho escrito (relatório) deverá estar no formato [ABNT](#) e estruturado da seguinte forma: Introdução; Desenvolvimento (diagrama funcional em blocos, projetos, integração dos blocos, descrições do funcionamento, etc); Conclusão.
- A apresentação e a entrega do trabalho deverão ser feitas no horário da última aula de laboratório. Atrasos não serão tolerados, resultando em nota zero.
- A avaliação será feita levando em conta o projeto em funcionamento e o trabalho escrito, sendo ambos considerados com pesos iguais.
- Os testes do projeto no kit poderão ser feitos sempre nos horários de aula durante as semanas que antecedem o prazo final. Outros horários poderão ser eventualmente utilizados em função da disponibilidade do laboratório e do professor.
- Nesse projeto, o requisito mínimo é o desenvolvimento e a montagem do temporizador descendente. Assim, em caso de dificuldades com as outras etapas, priorize o projeto e montagem desse temporizador para evitar uma nota baixa. Para dar suporte ao projeto, podem ser usadas as interfaces para chaves, botões, LEDs e Displays disponíveis no site da disciplina, além dos circuitos desenvolvidos ao longo do semestre.
- Os alunos devem levar a planilha da ultima pagina do roteiro impressa para preencher a lápis na sala de aula durante a apresentação do trabalho.

# Batalha Naval

	d	C	b	A	9	8	7	6	5	4	3	2	1	0
0														
1														
2														
3														

	d	C	b	A	9	8	7	6	5	4	3	2	1	0
0														
1														
2														
3														

 Porta-aviões (x1)

 Cruzadores (x2)

 Submarino (x1)