

Universidade Federal de Santa Catarina
EEL5105: Circuitos e Técnicas Digitais Semestre:
2021/2 – Projeto

Jogo Mastermind

O projeto final consiste na implementação de um circuito para placa de desenvolvimento *DE2* fazendo uso das estruturas e conhecimentos obtidos durante o curso. O circuito implementa um jogo interativo para a obtenção de um código secreto de 4 caracteres BCD inserido através das chaves **SW(15..0)**. O comportamento do jogo é descrito a seguir:

- O usuário começa no estado *Init* resetando o jogo para passar diretamente ao estado *Setup*. Uma vez no estado *Setup* o usuário deve escolher um dos 4 níveis de dificuldade com os *Switches*, **SW(1..0)**. A letra *L* de *Level* aparecerá no HEX1 e o nível do jogo no HEX0, sendo 0 o nível mais fácil e 3 o nível mais difícil. Uma vez escolhido o nível do jogo, o usuário escolherá uma das 16 sequências possíveis com os *Switches*, **SW(5..2)**, as quais estão descritas em 4 blocos distintos de memória, um para cada nível de jogo. No HEX3 aparecerá a letra *C* de *code* e o caractere hexadecimal da sequência escolhida no HEX2. Cada bloco de memória deve possuir 16 linhas de informação de 16-bits. As memórias serão preenchidas da seguinte forma:

Para a memória do **nível 0**, cada linha deverá conter a uma sequência de **4 números de 0 até 3** em BCD.

Para a memória do **nível 1**, cada linha deverá conter a uma sequência de **4 números de 0 até 5** em BCD.

Para a memória do **nível 2**, cada linha deverá conter a uma sequência de **4 números de 0 até 7** em BCD.

Para a memória do **nível 3**, cada linha deverá conter a uma sequência de **4 números de 0 até 9** em BCD.

Corre a cargo do aluno preencher os blocos de memória que estão parcialmente preenchidos com sequências à sua escolha, lembrando que nas sequências **todos os dígitos tem de ser diferentes**. Nos LEDs vermelhos, **LEDR(15..0)**, estarão indicadas as rodadas em formato termométrico, na primeira rodada (rodada 0), os LEDRs estarão desligados.

- Uma vez pressionado o botão de pressão *enter* (KEY1) o jogo passa ao estado *Play* e se inicia o jogo. No estado *Play* o usuário tem 10 segundos para selecionar uma sequência usando os *Switches*, **SW(15..0)**, conforme ilustra a figura 1. Lembrando que no código todos os dígitos tem de ser diferentes. Neste estado, os displays HEX5 e HEX4 mostrarão a letra *t* de *time* e uma contagem de 0 a 9 com frequência de 1Hz, respectivamente e nos displays **HEX(3..0)** a sequência escolhida na rodada. Se o jogador não pressiona *enter* antes dos 10 segundos um sinal de status chamado *end_time* é ativado e o jogo vai para o estado *Result*, pelo contrário, se o jogador seleciona uma sequência e pressiona *enter* antes dos 10 segundos o jogo vai para o estado *Count_Round*.
- No estado *Count_Round* é feito a contagem de rodada, onde a rodada *x* ligará o **LEDR(x-1)**, e o jogo vai para o estado *Check*. O jogador possui 16 rodadas para adivinhar a sequência.
- No estado *Check* onde será avaliado se:
 - 1) O usuário acabou o número de rodadas máximo, onde caso tinha chegado a 16 rodadas, um sinal de *status* chamado *end_round* será ativado.
 - 2) O usuário adivinhou os quatro caracteres da sequência, onde um sinal de *status* chamado *end_game* será ativado. Caso um dos dois sinais de *status* esteja ativo, o jogo vai para o estado *Result*, em caso contrário vai para o estado *Wait*.
- No estado *Wait*, o display HEX3 mostrará a letra *P* de posição e o HEX2 mostrará o número de posições em que os caracteres coincidem com a sequência a ser adivinhada. O display HEX1 mostrará a letra *E* de existência e o HEX0 mostrará o número de caracteres que existem na sequência a ser adivinhada, mas estão fora da posição correta. Uma vez anotada a informação de *P* e *E* na folha das sequências disponível no Moodle e pensar na seguinte rodada, o jogador pressiona *enter* e passa de novo ao estado *Play* para a seguinte rodada.
- No estado *Result* será mostrado nos displays **HEX(3..0)** a sequência a ser adivinhada. No estado *Result* é também mostrado o resultado do número de pontos em Hexadecimal nos displays HEX7 e HEX6. O resultado final seguirá a seguinte fórmula $16 \times \text{end_game} + (10 - x)$, onde *x* é o número de rodadas (a fórmula está expressada em formato vetor no *datapath*). O usuário deverá pressionar *enter* para passar ao estado *Init* e iniciar outra rodada.
- Um usuário pode em qualquer momento parar o jogo usando o botão de pressão *reset* (KEY0) zerando o sistema, para assim reiniciar de novo.
- Visando evitar problemas de temporização em função do aperto de um KEY por um ser humano durar muitos ciclos de *clock*, o *Button Press Synchronizer* (ButtonSync) será fornecido em conjunto com o projeto deve ser utilizado. O *ButtonSync* converte apertos das KEYS em pulsos com período de um ciclo de clock.
- O projeto deverá ser implementado obrigatoriamente usando a abordagem *datapath-control* vista nas aulas de teoria. O modelo de *datapath* pode ser encontrado no Moodle da disciplina. Corre ao cargo do aluno projetar o *datapath* e juntar com o controlador.

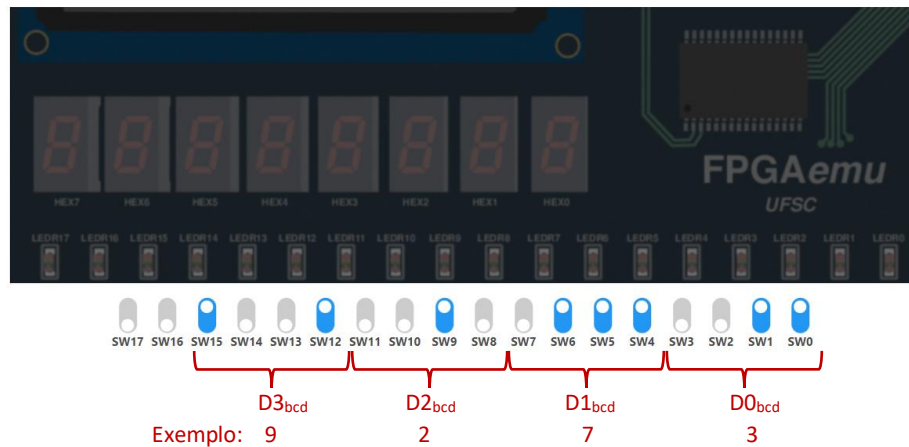


Figura 1 – Exemplo de como efetuar a entrada do código de 4 dígitos em uma jogada.

Orientações Gerais:

- Os testes do projeto poderão ser feitos durante as semanas que antecedem o prazo final usando as ferramentas on-line usadas no semestre. O professor estará disponível para solucionar dúvidas nos horários indicados como *Projeto* no cronograma.
- Os alunos devem mostrar no dia da apresentação remota uma emulação indicando o funcionamento do projeto, todos os membros do grupo deverão estar presentes.