

Universidade Federal de Santa Catarina  
EEL5105: Circuitos e Técnicas Digitais  
Semestre: 2025/2 – Projeto

Jogo IQ da memorização rápida

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 vai implementar a um jogo interativo para a memorização de códigos. O comportamento do jogo está definido 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 4 sequências possíveis com os *Switches*, *SW*(3..2), as quais estão descritas em 4 memórias (e 4 memórias auxiliares para fazer a comparação). No HEX3 aparecerá a letra *C* de *code* e o caractere hexadecimal da sequência escolhida no HEX2. As memórias principais possuem 16 palavras de 32-bits e as auxiliares tem 16 palavras de 15 bits. As memórias principais serão preenchidas com caracteres de 0 ate E em hexadecimal distribuídos em 8 posições. Para a memória do nível 0, 1, 2 ou 3 cada linha corresponde a uma sequência com 3, 4, 5, ou 6 caracteres possíveis, respectivamente.

O aluno deve preencher as memórias 1, 2 e 3 (um exemplo de memória foi dado na memória 0) com sequências à sua escolha lembrando que todos os dígitos tem de ser diferentes nas sequências e que as posições sem caracteres devem ser preenchidos com uns). Para o preenchimento das memórias auxiliares foi disponibilizado um exemplo no Moodle. Nos LEDs vermelhos, *LEDR*(15..0), serão mostradas as rodadas e os bônus em formato termométrico usando o Switch *SW*(17) (*SW*(17)=0 mostra as rodadas e *SW*(17)=1 mostra os bônus).

- Uma vez pressionado o botão de pressão *enter* (KEY1) o jogo passa ao estado *Play\_FPGA* e se inicia o jogo. No estado *Play\_FPGA* será mostrada nos Displays *HEX7*...*HEX0* a sequência de caracteres durante um tempo definido no nível de dificuldade. Para a dificuldade 0, 1, 2 ou 3 a sequência será mostrada durante 5, 4, 3 ou 2 segundos, respectivamente.

Uma vez que o tempo de amostragem da sequência termina, um status chamado *end\_FPGA* será ativado e passamos ao estado *Play\_user*.

- No estado *Play\_user*, os displays ficam desligados e o usuário tem 10 segundos para selecionar uma sequência usando os *Switches*, *SW*(14..0), lembrando de novo que no código todos os dígitos mostrados tem de ser diferentes. Neste estado, os displays *HEX5* e *HEX4* mostrarão a letra *t* de *time* e uma contagem regressiva de 10 a 0 com frequência de 1Hz, respectivamente. 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*, em caso contrario, 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 completar o jogo.
- No estado *Check* será avaliado se:

1) O numero de rodadas máximo foi atingido, onde caso tinha chegado a 16 rodadas, um sinal de *status* chamado *end\_round* será ativado.

2) O usuário errou na seleção dos switches, onde nesse caso será descontado o numero de posicoes que errou ao valor de bônus o qual será mostrado nos *LEDR*(7 downto 0) em formato termométrico decrescente. O numero de bônus inicial será 15. Um sinal de *status* chamado *end\_game* será ativado caso o jogador chegue a acabar com os bônus.

Caso algum dos sinais de *status* esteja ativo, o jogo vai para o estado *Result*, e em caso contrario vai para o estado *Wait*.

- No estado *Wait*, o jogador pressiona *enter* e passa de novo ao estado *Play\_FPGA* para a próxima rodada.
- No estado *Result*, é mostrado o resultado do numero de pontos em Hexadecimal nos *displays* *HEX7* e *HEX6*. O resultado final seguirá a seguinte formula  $32 \times L + 4 \times \lfloor \frac{B}{2} \rfloor + \lfloor \frac{X}{4} \rfloor$ , onde *B* é o número de bônus, *X* o numero de rodadas e *L* o nível do jogo seleccionado (o símbolo  $\lfloor \rfloor$  indica arredondamento para baixo). O usuário deverá pressionar *enter* para passar ao estado *Init* e iniciar outro jogo.
- Um usuário pode a qualquer momento, parar o jogo usando o botão de pressão *reset* (KEY0) zerando o sistema, para assim re-iniciar 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 para 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. O aluno deve projetar o *datapath* e juntar com o controlador.

Orientações Gerais:

- Os testes do projeto poderão ser feitos durante as semanas que antecedem o prazo final usando as ferramentas 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 o funcionamento do projeto, todos os membros do grupo deverão estar presentes.