

Universidade Federal de Santa Catarina  
EEL5105: Circuitos e Técnicas Digitais  
Semestre: 2022/1 – Projeto

Jogo das sequencias

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 de dois jogadores, o jogador *left* (L) e o jogador *right* (GR), para continuar sequencias mostradas no displays (sequencias em BCD) ou LEDs (sequencias binarias). O comportamento do jogo está definido a seguir:

- Os jogadores começam no estado *Init* resetando o jogo para passar diretamente ao estado *Setup*. Uma vez no estado *Setup* os jogadores deverão escolher um dos 4 níveis de dificuldade com os *Switches*, *SW*(1..0). A letra *L* de *Level* aparecera no HEX1 e o nível do jogo no HEX0, sendo 0 o nível com menor penalização por erro na detecção das sequencias e 3 o nível com maior penalização por erro na detecção das mesmas. Uma vez escolhido o nível do jogo, os jogadores escolherão um dos 4 jogos possíveis com os *Switches*, *SW*(3..2), as quais estão descritas em 4 memórias. No HEX3 aparecera a letra *J* de *Jogo* e o caracter decimal da sequencia escolhida no HEX2. As memórias são dadas aos alunos.
- Uma vez pressionado o botão de pressão *enter* (KEY1) o jogo passa ao estado *Sequence* e se inicia o jogo com a primeira rodada. No estado *Sequence*, aparecera em HEX7 e HEX6 o valor inicial de uma contagem regressiva em segundos para o jogador da esquerda (L) e em HEX3 HEX2 o mesmo valor para o jogador da direita (R). O valor inicial da contagem regressiva será de 99 segundos. Existem dois tipos de rodadas:

**Rodada impar:** Sao mostrados nos displays HEX5 HEX4 uma sequencia de 4 valores em BCD para o jogador da esquerda (L) e nos displays HEX1 HEX0 uma sequencia de 4 valores em BCD para o jogador da direita (R), ambas mostradas a uma frequência de 1Hz.

**Rodada par:** Sao mostrados nos LEDR(17.. 10) uma sequencia de 4 valores em binário para o jogador da esquerda (L) e nos LEDR(7.. 0) uma sequencia de 4 valores em binário para o jogador da direita (R), ambas mostradas a uma frequência de 1Hz.

Uma vez que foi mostrada a sequencia de 4 valores (seja par ou impar) será ativado um status *end\_sequence* e o jogo passa a o estado *Play*.

- Uma vez no estado *Play* começa a contagem regressiva e os usuários tem de colocar o valor do seguinte caracter da sequencia usando os *SW*(17...10) o jogador da esquerda (L) e os *SW*(7..0) o jogador da direita (R). Caso estejamos na rodada impar será introduzido em formato BCD e se fosse rodada par será introduzido usando código binário. Uma vez selecionado o valor nos *SW* o jogador da esquerda pressionara *enter\_left* no botão KEY(3) e a sua contagem parará, e de igual forma o jogador da direita pressionara *enter\_right* no botão KEY(2) e a sua contagem parará. Quando *enter\_left* e *enter\_right* tenham sido pressionado passamos ao estado *Check*.
- No estado *Check* será avaliado se foi adivinhado o valor das sequencias pelos jogadores. Caso algum usuário não adivinhe será subtraído 2, 4, 6 ou 8 segundos na contagem regressiva dependendo de se escolhermos nível de dificuldade 0, 1, 2 ou 3, respectivamente. Nesse estado será descontado uma rodada do contador de rodadas. Do estado *Check* passamos diretamente ao estado *Wait*.
- No estado *Wait*, e mostrado nos LEDRs o numero de rodadas em formato termometrico decrescente e o tempo restante para os jogadores nos displays HEX(7..6) para o jogador L e HEX(3..2) para para o jogador R. Um dos jogadores pressiona *enter* e passa de novo ao estado *Sequence* para a próxima rodada. Caso a contagem de algum jogador esteja zerada (ou negativa) um sinal de *status* chamado *end\_game* será ativado e o sistema vai para o estado *Result*. Se o numero de rodadas máximo 16 foi atingido, um sinal de *status* chamado *end\_round* será ativado e o jogo vai para o estado *Result*.
- No estado *Result*, será mostrado nos displays *HEX*(3..2) o tempo que sobrou para cada usuário e os LEDRs associados ao usuário ganhador piscando a uma frequência de 1HZ. Os usuários deverão pressionar *enter* para passar ao estado *Init* e iniciar outra rodada.
- 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.
- Para o sinal de relógio o aluno pode usar  $CLK = CLK\_500Hz$  para testes usando emulador online (ou usar o sinal de  $CLK\_1Hz$  ao invés do divisor de frequência dado). Para testes na placa DE2 os alunos devem usar o relógio interno de  $50Mhz$   $CLK = CLOCK\_50$ . 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 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 o projeto em funcionamento na placa DE2, todos os membros do grupo deverão estar presentes.