

Sistemas Digitais I	Projeto	Data	MS
PCS3115			

**PROJETO 3.** Deseja-se construir um circuito digital sequencial síncrono para emular o jogo Bass Hero, uma variante do Guitar Hero com 4 cordas apenas. No jogo, o jogador deve acertar as notas musicais (botões do jogo) apresentadas para ganhar pontos e tornar-se uma estrela do Rock n' Roll. \,,,/

O projeto é dividido em duas partes:

A1) MODO SOLO: A interface do circuito é a seguinte:

```
entity bass_hero_solo is
    port (clk, reset: in bit;
          target: in bit_vector (3 downto 0);
          played: in bit_vector (3 downto 0);
          score: out bit_vector (2 downto 0);
          cheers: out bit
    );
end entity;
```

Esse circuito síncrono, usado para o modo com 1 único jogador, calcula a saída `score` em função dos acertos desse jogador em cada borda do relógio `clk`: no início, após fazer `reset=1` (síncrono), o circuito dá como saída 0 pontos; toda vez que o jogador acerta uma nota (ou seja, `target = played` na borda de subida do relógio), o número de pontos dado pelo circuito é incrementado de 1, até o limite de 2 pontos; toda vez que o jogador erra (ou seja, `target ≠ played` na borda de subida do relógio), o número de pontos é decrementado de 1, até o limite de -2 pontos (usando notação em complemento de dois). Como incentivo para jogadores que estejam indo muito bem, toda vez que ele estiver no limite superior de pontos e continuar acertando as notas, é a saída `cheers` passa a dar '1', o que ativa um circuito de aplausos e gritos eufóricos. A qualquer momento, o jogador pode acionar o `reset` (indicando que desiste), o que reinicia a pontuação para o valor 0 e interrompe os aplausos.

Os exemplos abaixo ilustram o comportamento desejado (onde X representa "don't care"):

clk	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Reset	1	0	0	0	0	0	0	0	0	0	0	0
Target	X	0100	0110	0100	1100	0001	0110	0010	1100	0001	0110	1100
Played	X	0100	0110	0100	1100	1001	1111	1101	0011	1010	0110	1100
Score	000	001	010	010	010	001	000	111	110	110	111	000
Cheers	0	0	0	1	1	0	0	0	0	0	0	0
Obs:	0	1	2	2	2	1	0	-1	-2	-2	-1	0

clk	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Reset	1	0	0	0	0	0	0	1	0	0	1	0
Target	X	1100	1100	1100	1100	1100	1100	X	1100	1100	X	1100
Played	X	1100	0011	1100	1100	0000	1100	X	0011	0011	X	1100
Score	000	001	000	001	010	001	010	000	111	110	000	001
Cheers	0	0	0	0	0	0	0	0	0	0	0	0
Obs:	0	1	0	2	2	1	2	0	-1	-2	0	0

A2) MODO VERSUS: A interface do circuito é a seguinte:

```
entity bass_hero_versus is
  port (clk, reset: in bit;
        target: in bit_vector (3 downto 0);
        played: in bit_vector (3 downto 0);
        jam: in bit;
        score: out bit_vector (2 downto 0);
        cheers: out bit
  );
end entity;
```

Esse circuito, usado no modo com 2 jogadores, tem comportamento idêntico ao circuito do modo solo com uma exceção: a entrada jam, fornecida pelo oponente do jogador. Essa entrada, quando colocada em '1', faz com que o número de pontos do jogador seja decrementado de 2 independentemente de acertos ou erros da nota tocada, de modo análogo ao que aconteceria se, naquela borda de subida do clock, houvessem "dois erros simultâneos". Esse decremento é tal que o limite inferior de -2 pontos ainda é respeitado e também leva à interrupção de aplausos como ocorre em erros normais. A qualquer momento, o jogador ainda pode acionar o reset, o que reinicia a pontuação para o valor 0 e também interrompe os aplausos.

O exemplo abaixo ilustra o comportamento desejado (onde X representa "don't care"):

clk	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Reset	1	0	0	0	0	0	0	0	0	1	1	0
Target	X	1100	1100	1100	1100	1100	1100	1100	1100	X	1100	1100
Played	X	1100	1100	1100	X	1100	X	X	1100	X	1100	X
Jam	X	0	0	0	1	0	1	1	0	X	0	1
Score	000	001	010	010	000	001	111	110	111	000	001	111
Cheers	0	0	0	1	0	0	0	0	0	0	0	0
Obs:	0	1	2	2	0	1	-1	-2	-1	0	1	-1

### **Instruções para Entrega**

Você deve acessar o link específico para cada tarefa (T3A1, T3A2) dentro do tópico "Projetos" no e-Disciplinas, já logado com seu usuário e senha, o que o levará à página apropriada do juiz eletrônico. Em cada atividade, você pode enviar apenas um único arquivo codificado em UTF-8. O nome do arquivo não importa, mas sim a descrição VHDL que está dentro: as entidades nas suas soluções devem ser idênticas àquelas neste enunciado ou o juiz não irá processar seu arquivo.

Quando acessar o link no e-Disciplinas, o navegador abrirá uma janela para envio do arquivo. Selecione-o e envie para o juiz. Jamais recarregue a página de submissão pois seu navegador pode enviar o arquivo novamente, o que vai ser considerado pelo juiz como um novo envio e pode prejudicar sua nota final. Caso desista do envio, simplesmente feche a janela. Depois do envio, a página carregará automaticamente o resultado do juiz, quando você poderá fechar a janela. A nota dada pelo juiz é somente para a submissão que acabou de fazer.

O prazo para a submissão das soluções no Juiz é aquele definido no e-disciplinas. O Juiz aceitará até 5 submissões para cada atividade deste projeto. Sua submissão será corrigida

imediatamente e sua nota será apresentada. A maior nota dentre as submissões será considerada. Neste trabalho, os problemas valem no máximo 10 pontos no juiz, porém a nota final deste trabalho será calculada como a média simples das notas obtidas em cada atividade neste enunciado, totalizando 10 para o trabalho todo. Como boa prática de engenharia, faça seus testbenchs e utilize o GHDL ou o EDAPlayground para validar suas soluções antes de postá-las no juiz, ou há grandes chances de que você desperdice chances de submissão fazendo debug.

**Importante:** como trata-se de um circuito sequencial síncrono, você está autorizado (e, de fato, precisará!) utilizar comandos para construção de circuitos sequenciais, como **process**.