

Universidade Federal de Santa Catarina
EEL5105: Circuitos e Técnicas Digitais
Semestre: 2015/2 – Projeto

Slot Machine

O projeto final consiste na implementação de um circuito na placa de desenvolvimento *DE1* fazendo uso das estruturas e conhecimentos obtidos durante o curso. O circuito vai implementar uma máquina interativa similar às slot machines encontradas nos cassinos. O comportamento de tal máquina está definido a seguir:

- O usuário inicia o jogo com um crédito $23_{(10)}$ indicado em binário nos LEDs vermelhos. O objetivo consiste em ficar com um crédito maior que o inicial após 10 rodadas.
- O usuário escolhe um nível de dificuldade (velocidade de jogo), com o Switch $\{SW9\}$, $SW9 = 0$ nível 1 e $SW9 = 1$ nível 2.
- As três sequências em hexadecimal são mostradas nos displays *HEX2*, *HEX1* e *HEX0*. As sequências a serem implementadas pelo aluno serão fornecidas previamente pelo professor para cada grupo.
- A tabela de prêmios com as combinações que fazem aumentar ou subtrair o crédito será parcialmente fornecida pelo professor e parcialmente preenchida pelo aluno.
- Uma vez selecionado o nível de dificuldade, o usuário pulsa o botão *KEY3* e as sequências começam a correr nos displays.
- O usuário deve acertar as combinações que aumentam o crédito usando o botão de pressão *KEY3* (que tem também a funcionalidade *STOP*). Cada vez que *KEY3* é pressionada, um sequenciador será parado ($HEX2 \rightarrow HEX1 \rightarrow HEX0$) e, caso o aluno acerte uma combinação de prêmio, o crédito será aumentado ou subtraído.
- Caso o nível de jogo seja 2 os prêmios serão multiplicados $\times 2$.
- Caso o usuário fique sem crédito, o jogo deve ser finalizado e se reiniciará o sistema.
- O usuário pode em qualquer momento parar o jogo usando o botão de pressão *KEY0* zerando o crédito, para assim reiniciar novamente.

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

- *Sequenciadores*: Três sequenciadores (ou FSM) encarregados de gerar as três sequências da slot machine.
- *Seletor de nível*: FSM encarregada de gerar um clock de $1Hz$ ou $2Hz$ a partir do clock interno de $50MHz$ da placa *DE1*. O nível de jogo para cada rodada é definido pelo switch *SW9*, permitindo dois possíveis tempos de jogo.

- *Comparador*: Circuito encarregado de gerar o array, *CREDITO*, que determina qual será a soma/subtração de crédito. Este circuito possui uma memória ROM onde os alunos devem introduzir a tabela de prêmios. A disponibilidade do array *CREDITO* pode ser obtida mediante uma FSM ou mediante lógica combinatória.
- *Contador de pontos*: Circuito sequencial encarregado de somar os prêmios obtidos pelo usuário. Após 10 rodadas ou credito negativo o somador precisa ser zerado.
- *Controle*: Este bloco é responsável por gerar os sinais de controle para os sequenciadores, para o comparador e para o contador de crédito. Este é o bloco que deve implementar o diagrama de estados da slot machine. Portanto, para o controlador, é necessário projetar uma FSM adicional (FSM Control).

Um exemplo de três sequências e tabela de prêmios é mostrado na Figura 2. Após

HEX2: 0→8→A→5→6→4→2→1→3→7→9→b→E→F→C→d→0...
 HEX1: 0→8→d→2→1→3→7→9→b→E→F→C→A→5→6→4→0...
 HEX0: 0→1→3→7→9→b→E→F→C→d→8→A→5→6→4→2→0...

TABELA DE PREMIO

HEX2	HEX1	HEX0	Cred (L1)	Cred (L2)
7	7	7	50	100
A	A	A	5	10
1	2	3	1	2
1	2	1	-1	-2
1	2	7	-4	-8
A	A	8	-5	-10
A	A	5	-49	-99
7	7	3	-50	-100
7	7	9	-50	-100
7	7	b	remove all	remove all

Figura 2: Exemplo de sequências e tabela de prêmios.

10 rodadas, o valor máximo possível será 1023 tendo em consideração o crédito inicial de 23. **As sequências e tabelas serão parcialmente fornecidas numa planilha pelo professor para os alunos completarem. Os prêmios máximos e mínimos são fornecidos pelo professor na tabela podendo o aluno incluir até 32 linhas de prêmios na memória ROM. A última linha com o premio *remove all* é opcional no projeto.**

SEQUENCIADORES

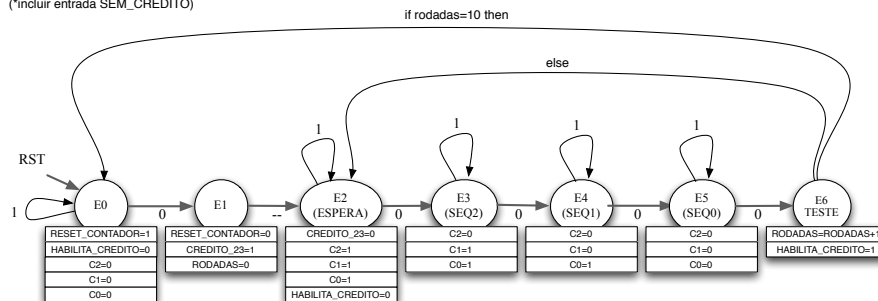
Os sequenciadores a ser projetado por cada um dos grupos deverá gerar as sequências de 16 valores. Por isso, cada dupla deverá enviar um e-mail ao seu professor de laboratório, identificando os nomes dos elementos da dupla que vão fazer o projeto, e posteriormente o professor responderá via e-mail com três sequências de 16 valores que o grupo deve implementar em VHDL fazendo uso dos conceitos sobre FSM aprendidos na sala de aula. O controlador dos sequenciadores deve ser implementado usando outra FSM usando *INIT/STOP* como entradas primárias e os bits de controle de cada sequenciador *C0*, *C1*, *C2*, como saídas primárias, além de outras saídas que controlaram os blocos comparador e somador. Um possível diagrama de estados do controlador é apresentado na Figura 3 (sem incluir a entrada *SEM_CREDITO*). É de responsabilidade do aluno interpretar o diagrama e, caso seja possível/necessário, melhorá-lo.

Dicas: Recomenda-se o uso de dois process para a implementação da FSM. A máquina de estados aqui apresentada não é única podendo ser refeita pelos alunos.

SELETOR DE NÍVEL

O seletor de nível a ser projetado deverá fornecer um clock de 1Hz e de 2Hz a partir do clock interno de 50MHz. O *Switch* 9 deverá controlar o nível de jogo: *SW9* = 0 acionará nível 1 e *SW9* = 1 acionará o nível 2.

Dicas: Recomenda-se aplicar os conceitos aprendidos na aula 7b sobre a FSM que gera o laço de atraso de 1 segundo. O MUX2 : 1 indicado na Figura 1 pode estar inserido dentro da própria FSM_clock.



COMPARADOR

Dica: A informação dos prêmios deve ser transferida para o contador de crédito só quando o bit HABILITA_CREDITO está disponível. Lembrar que é necessário incluir o bit SW9 no endereço da memória para poder incluir os prêmios $\times 2$.

A soma/subtração de créditos deverá ser implementada usando um somador sincronizado (somador e registrador). O somador sincronizado deverá estar habilitado ($HABILITA_CREDITO = 1$) para que aconteça a soma. O contador de crédito deve zerar em três situações: 1) uma vez pressionado o reset assíncrono $KEY0$, mas também 2) quando chegar à décima rodada ou 3) atingir um crédito negativo quando o bit de sinal (MSB) passa a 1 lógico (nas situações 2) e 3) o $RESET_CONTADOR = 1$). Uma vez reiniciado o jogo, o contador deve atualizar-se com o valor $23_{(10)}$ ("000010111₍₂₎").

O bit de sinal MSB deve ser entrada do controlador e corre a cargo do aluno incluir dita entrada no diagrama de estados como explicado a seguir na descrição do controlador.

Dica: Use CREDITO_23 da saída da FSM_control para atualizar a contagem a 23₍₁₀₎ usando MUX2:1. O reset a 0 deve ser seleccionado usando três sinais, use um circuito combinatorio para detectar quando reset deve estar activo tendo em consideração as 8 combinações possíveis.

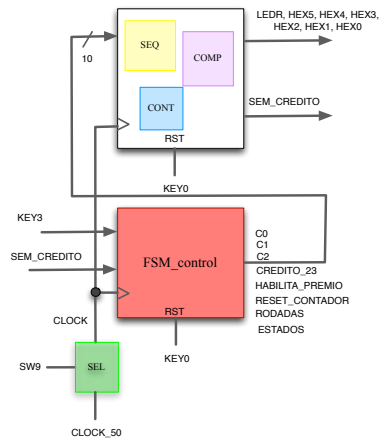


Figura 4: Controlador e bloco operacional.

CONTROLADOR

A Figura 4 apresenta o controlador com as suas saídas controlando os demais blocos operacionais. Na Figura 2 podemos observar que o estado inicial da FSM é aguardando a escolha do nível de dificuldade do jogo. Neste estado inicial o contador de crédito está em estado de reset (*RESET_CONTADOR*) e a soma de prêmios desabilitada (*HABILITA_PREMIO*), pois o jogo ainda não iniciou.

Uma vez que o usuário escolhe o nível (a partir da chave *SW9*) e pressiona o botão *KEY3* para confirmar, há uma mudança de estado e o contador sai do estado de reset. O número de rodadas (*RODADAS*) recebe seu valor inicial e o crédito inicial é atribuído para o usuário (*CREDITO_23*).

No próximo estado a FSM aguarda o usuário parar os sequenciadores. Para o controle dos sequenciadores, o controlador deve fornecer as saídas *C0*, *C1*, *C2*. Portanto, existirá um estado para cada sequenciador, onde a FSM irá aguardar que o botão *KEY3* seja pressionado.

Em um próximo estado, a saída *RODADAS* (quatro bits) é incrementada e verificada. Caso dez rodadas tenham sido realizadas, o jogo deve reiniciar. Caso contrário, os sequenciadores devem voltar a correr e uma nova rodada se inicia.

Os *displays* *HEX5* e *HEX4* mostraram a letra E de *Estado* e o número de estado, respectivamente. Por exemplo, no estado "espera" deveria aparecer nos Displays 5 e 4 os caracteres *E2*.

Dica: A partir das informações acima desenhe um diagrama de estados, identificando os valores das saídas para cada estado. Neste diagrama anote também o nível lógico da entrada primária KEY3. Atenção, lembre-se que a FSM deve voltar ao estado inicial em dois casos: depois de dez rodadas ou quando a soma de crédito se tornar negativa. Neste último caso, o contador de crédito deve informar a FSM de controle por meio de um sinal SEM_CREDITO. Este sinal pode ser verificado no mesmo estado em que RODADAS é verificado.

Orientações Gerais:

- Na apresentação, todos os membros do grupo deverão estar presentes.
- O aluno deverá usar o modelo de trabalho escrito disponível no moodle da turma de laboratório (Trabalho.doc), e deve ser mantida a estrutura apresentada neste modelo.
- Os códigos VHDL do trabalho devem ser entregues via moodle e eles não devem ser incluídos, mas sim comentados, no trabalho escrito.
- 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 zero. Para dar suporte ao projeto, pode usar as interfaces para chaves, botões, LEDs e displays disponíveis no site da disciplina, além dos circuitos obtidos ao longo do semestre.
- Os alunos devem levar a planilha com a tabela de prêmios e sequências impressa para preencher a lápis na sala de aula na apresentação do trabalho.