

Universidade Federal de Santa Catarina  
EEL5105: Circuitos e Técnicas Digitais  
Semestre: 2016/1 – Projeto

## Telefone Celular

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 com funcionamento similar a um telefone celular. O comportamento do telefone está definido a seguir:

- O usuário inicia com o telefone desligado (estado *DESL*). Mediante o uso do botão *KEY3* (botão *enter*) liga o celular e passa para o estado (*intro\_PASS*).
- O usuário deve introduzir o *password* de 4 caracteres em decimal usando os *Switches*  $\{SW9, \dots, SW0\}$  e botão *enter*. Uma vez seleccionado o valor em decimal  $X$  no switch  $X$  correspondente é pressionado o botão *KEY3*, e o caracter em hexadecimal deve aparecer nos *displays*. O primeiro caracter aparecerá no *HEX3* no estado *REG3*, o segundo no *HEX2* no estado *REG2*, o terceiro no *HEX1* no estado *estado REG1* e o ultimo no *HEX0* no estado *REG0*.
- Uma vez introduzidos os 4 caracteres do *password* o sistema passará ao estado *TESTE*. Uma vez pressionado *enter* se avaliará se o *password* está correto e o numero de tentativas que o usuário usou. Se o usuário introduz erradamente o *password*, o sistema volta ao estado *intro\_PASS*. Caso esteja o *password* correto o sistema passará ao estado *ON*.
- O numero de tentativas para introduzir o *password* é quatro. Caso o usuário erre 4 vezes consecutivas o *password*, o sistema voltara ao estado desligado.
- No estado *ON* o usuário deverá escolher um nome da agenda usando os *Switches*  $\{SW3, \dots, SW0\}$ , e o nome é mostrado nos displays *HEX(3...0)*. Uma vez seleccionada a pessoa a ligar, pressiona-se *enter* e passamos ao estado *LIGA*.
- No estado *LIGA* o sistema deve mostrar dois contadores, o contador ascendente nos displays *HEX(3..0)* que indica o tempo da ligação em segundos, e o contador descendente que indica o saldo no LEDs. O formato da contagem ascendente deve ser *MM : SS* (minutos e segundos) e nos LEDs em formato binário. Uma vez pressionado *enter* o sistema deve voltar ao estado *ON*, caso acabe o saldo o sistema deve ser desligado.
- A palavra passe corresponderá aos últimos 4 dígitos da matrícula do aluno (caso o trabalho seja feito em dupla, corresponderá aos 4 últimos dígitos do aluno mais novo).
- O usuário possui um credito inicial igual a  $1000 + M$  onde  $M$  é o mês de nascimento do aluno (caso o trabalho seja feito em dupla, corresponderá a 1000 mais meses de nascimento do aluno mais velho).
- O período do relógio associado ao sistema (excepto a contagem descendente) será de 1 segundo. A contagem descendente terá uma frequência  $f_{Conta\_DES} =$

$((C + 2) \times f_{Conta\_ASC})$ , onde  $C$  é o ultimo digito da matricula do aluno (caso o trabalho seja feito em dupla, corresponderá ao último dígito do aluno mais novo). Tendo em consideração a formula anterior, a frequência do CLK2 deverá estar entre 2 e 11Hz.

- O usuário pode, em qualquer momento, parar o sistema usando o botão de pressão *KEY0*, zerando o crédito para assim reiniciar novamente.
- 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*) fornecido em conjunto com o projeto deve ser utilizado. O *ButtonSync* converte apertos das KEYS em pulsos com período de um ciclo de clock. Assim, em seu projeto, as KEYS devem ser ligadas nas estradas do *ButtonSync*, e as saídas *BTN0* a *BTN3* do *ButtonSync* deverão ser utilizadas para controlar o projeto.

O esquema geral do projeto é mostrado na Figura 1 e inclui seis blocos diferenciados que serão explicados com mais detalhe a seguir.

## REGISTRADORES

Quatro registradores encarregados de gerar os quatro dígitos do *password*. As entradas de enable  $C0$ ,  $C1$ ,  $C2$  e  $C3$  estarão definidas na maquina de controle definida mais a frente.

*Dicas: Recomenda-se o uso das dicas fornecidas na aula 6 para a implementação de registradores.*

## AGENDA

A memória  $2^4 \times 20$  ROM é onde o aluno deve preencher com a informação da agenda. No moodle da disciplina é fornecido parte do código de uma memoria ROM (ROM.vhd). O aluno deve escrever o resto do código da memória usando as dicas fornecidas na aula 9 de laboratório sobre memórias. O *array* de saída da ROM deve ser de 20 bits onde cada caracter a ser mostrado nos *displays* *HEX3*, *HEX2*, *HEX1* e *HEX0* possui 5 bits. A lista de caracteres a ser fornecida pelo decodificador de 7 segmentos está parcialmente mostrada na Figura 2(a). Por exemplo para visualizar LISA quando seleccionamos nos *Switiches*  $SW3..0 = "0000"$ , a ROM devera gerar uma

palavra de 20 bits na primeira linha da memória igual a "  $\overbrace{10101}^L \overbrace{10011}^I \overbrace{00101}^S \overbrace{01010}^A$ ". Um exemplo de agenda parcialmente preenchida é mostrada na Figura 2(c). O aluno devera preencher com nomes as 16 linhas da memória com nomes aleatórios.

*Dicas: O aluno de incluir mais um bit de entrada no decodificador de 7 segmentos desenvolvido na aula 5 e incluir até 16 caracteres adicionais mostrados na Figura 2(a) usando como exemplo os caracteres para sete segmentos da Figura 2(b). Por outro lado recomenda-se aplicar os conceitos mostrados na aula 9 sobre a memória ROM. O aluno deve preencher a tabela de verdade do decodificador de 7 segmentos com 32 caracteres diferentes (incluindo o caracter espaço para  $C4C3C2C1C0 = "11111"$ ). Por outro lado, o aluno também deve preencher a agenda com 16 nomes diferentes a partir da tabela dada na Figura 2(b).*

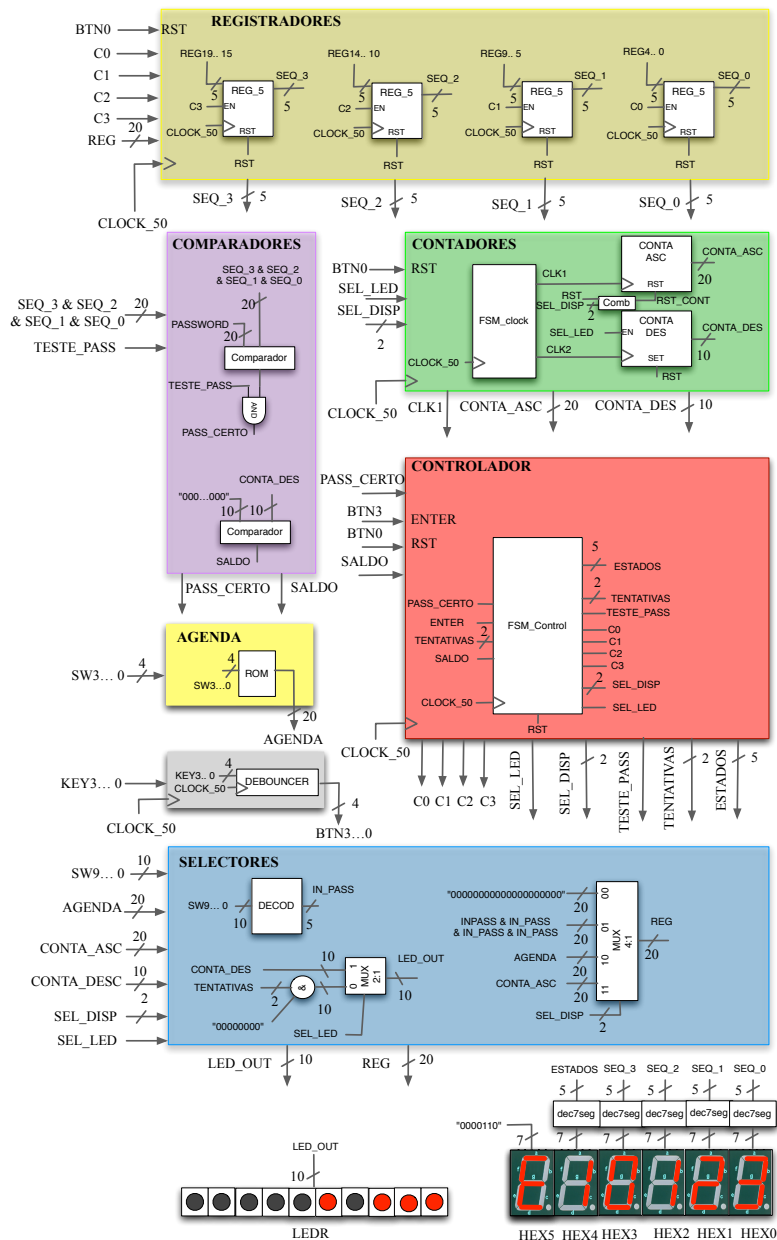


Figura 1: Diagrama de blocos do telefone celular.

**TABELA DEC7SEG**

[illegible]

(a)



(b)

## AGENDA

[illegible]

(c)

Figura 2: a) Tabela de verdade do Decodificador de 7 Segmentos a ser implementado, b) exemplo de caracteres em hexadecimal, e b) exemplo de agenda.

## COMPARADORES

Circuito combinatório encarregado de gerar duas entradas do controlador as quais indicam se determinados alvos foram atingidos (saldo zero ou *password* certo).

O primeiro comparador devera gerar um '1' lógico quando a introduzimos o *password* correto e o bit *TESTE PASS* dado pelo controlador é '1' lógico.

O segundo comparador deverá gerar um '1' lógico quando a conta descendente que indica o saldo chegou a zero.

*Dica: Podem usar uma abordagem estrutural baseada em portas ANDs.*

## CONTADORES

FSM encarregada de gerar dois clocks, *CLK1* de 1Hz e outro *CLK2* compreendido entre 2Hz até 11Hz a partir do *clock* interno de 50MHz da placa DE1. Além disso, este bloco deve gerar duas contagens crescentes e decrescente usando os *clocks* *CLK1* e *CLK2*, respectivamente.

A contagem crescente deverá ser mostrada nos *displays* em formato *MM : SS* (minutos e segundos). O aluno deverá criar uma FSM de um contador crescente usando o relógio CLK1 no qual ao chegar a 60 periodos some mais um na saída

correspondente a minutos. A contagem crescente deverá estar activa unicamente no estado *LIGA* como indicado na Figura 3 e zerada quando não se encontra em dito estado. O aluno deverá explicar qual será a lógica a ser introduzida na entrada de *reset* do contador crescente.

A contagem decrescente deverá contar desde um valor previamente definido na introdução do projeto até zero. A frequência usada nesta FSM deverá estar entre 2 e 11Hz tal e dependerá do mês de nascimento do aluno tal e como explicado previamente. A saída deverá estar em formato binário, assim não é preciso colocar mais saídas adicionais na FSM. A contagem decrescente deverá ser activa no estado *LIGA* e mantida fora dele. O aluno deverá procurar uma solução para "colocar" o crédito inicial cada vez que o celular é desligado.

### SELECTORES

Esta unidade consiste em um decodificador e dois multiplexadores. O decodificador deve transformar o sinal de 10 bits da selecção dos dígitos do *password* para os 5 bits que precisa a entrada do decodificador de 7 segmentos. Por exemplo quando seleccionamos o valor 7 nos *Switches SW*(9...0) = "0010000000" o decodificador deverá fornecer uma saída de 5 bits "00111". O aluno deverá obter a tabela de verdade para as 10 combinações permitidas.

Os multiplexadores deveram atuar para seleccionar o que é esperado nos displays e LEDs para cada estado.

*Dicas: Recomenda-se aplicar os conceitos aprendidos na aula 4 sobre o Multiplexador.*

### CONTROLADOR

Este bloco é responsável por gerar os sinais de controle para os outros blocos. Este é o bloco que deve implementar o diagrama de estados do telefone celular. A Figura 4 apresenta o controlador com as suas saídas controlando os demais blocos operacionais.

Na Figura 3 podemos observar que o estado inicial (*DESL*) da FSM consiste no aguardo do sistema ligar. Neste estado inicial o contador está em estado de *reset* pois o celular ainda não foi ligado. As saídas *SEL\_DISP* e *SEL\_LED* estão a "00" e "0" respectivamente indicando que vamos zerar os displays e mostrar o número de tentativas nos LEDs. O número de tentativas para introduzir o *password* é zerado.

O usuário deve pressionar o botão *KEY3* (*enter*) para ligar o celular e passamos ao estado *Intro\_PASS*, a saída *TESTE\_PASS* passa a zero indicando que ainda não vai acontecer a comparação com o *password*, a saída *SEL\_DISP* passa a "01" indicando que vai ser mostrado o *password* nos displays.

Nos seguintes 4 estados a FSM aguarda o usuário seleccionar os dígitos do *password*. Para a habilitação dos registradores que memorizam os dígitos do *password*, o controlador deve fornecer as saídas *C0*, *C1*, *C2*, *C3*. Portanto, existirá um estado para cada registrador, onde a FSM irá aguardar que o botão *KEY3* seja pressionado.

Em um próximo estado, a saída *TENTATIVAS* (quatro bits) é incrementada e verificada. Caso três tentativas tenham sido realizadas, o celular deve reiniciar. Caso o *password* tenha sido introduzido erroneamente, mas não usamos as 4 tentativas, o

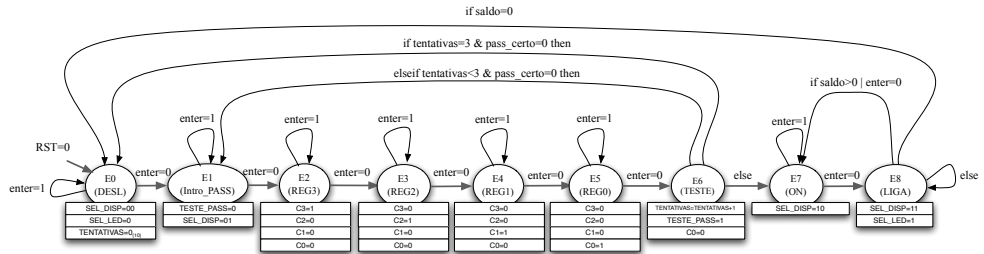


Figura 3: Diagrama de estados de alto nível da *FSM\_control*.

sistema volta ao estado *Intro\_PASS*. Caso o *password* esteja correcto passaremos ao estado *ON*.

No estado *ON* teremos que introduzir o nome desejado da agenda disponível usando os *SW(4...0)*, os *displays* devem mostrar o nome da agenda seleccionada. Uma vez seleccionada a pessoa a ser ligada é pressionado *enter* e passamos ao estado *LIGA*.

No estado *LIGA* deve ser mostrado no display a contagem crescente referente à duração da chamada (em formato MM:SS) e nos LEDs a contagem decrescente referente ao credito disponível (em binário).

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*.

*Dicas: Recomenda-se o uso de dois process para a implementação da FSM. O diagrama de estados apresentado na Figura 3 não é único podendo ser feito pelos alunos.*

*Dicas: Recomenda-se aplicar os conceitos aprendidos na aula 7b sobre a FSM que gera o laço de atraso de 1 segundo.*

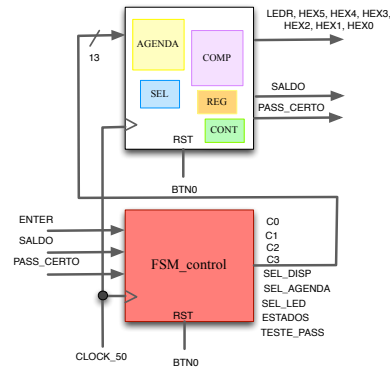


Figura 4: Controlador e bloco operacional.

#### 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 podem ser parcialmente 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.
- 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.
- Os alunos devem levar a planilha com a agenda na apresentação do trabalho.