



Universidade Federal de Santa Catarina
Centro Tecnológico – CTC
Departamento de Engenharia Elétrica



“EEL7020 – Sistemas Digitais”

Prof. Eduardo Augusto Bezerra

Eduardo.Bezerra@eel.ufsc.br

Florianópolis, agosto de 2011.

Sistemas Digitais

*Uso de FSMs no controle do fluxo
de execução de sistemas digitais.
Estudo de caso: Projeto de Calculadora.*

Definição do problema: calculadora modificada

- Para realizar uma **operação de soma**, a calculadora do Lab. 8 utiliza:
 - as chaves **SW(7..0)** para leitura do **operando A**, e
 - as chaves **SW(15..8)** para leitura do **operando B**.
- Visando reduzir a quantidade de chaves no projeto de uma placa, solicita-se utilizar **apenas um conjunto** de 8 chaves, **SW(7..0)**, para **leitura dos dois operandos**.

Entrada de dados para realizar uma soma

Entrada de dados na calculadora original (Lab. 8):



Operação
SW(17..16)

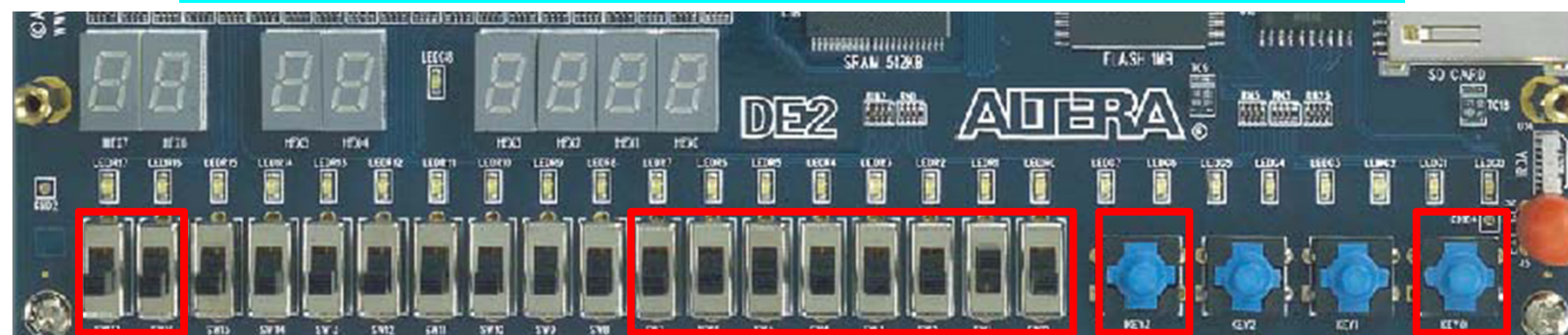
Operando B
SW(15..8)

Operando A
SW(7..0)

Reset
Key(3)

Enter
Key(0)

Entrada de dados na calculadora modificada:



Operação
SW(17..16)

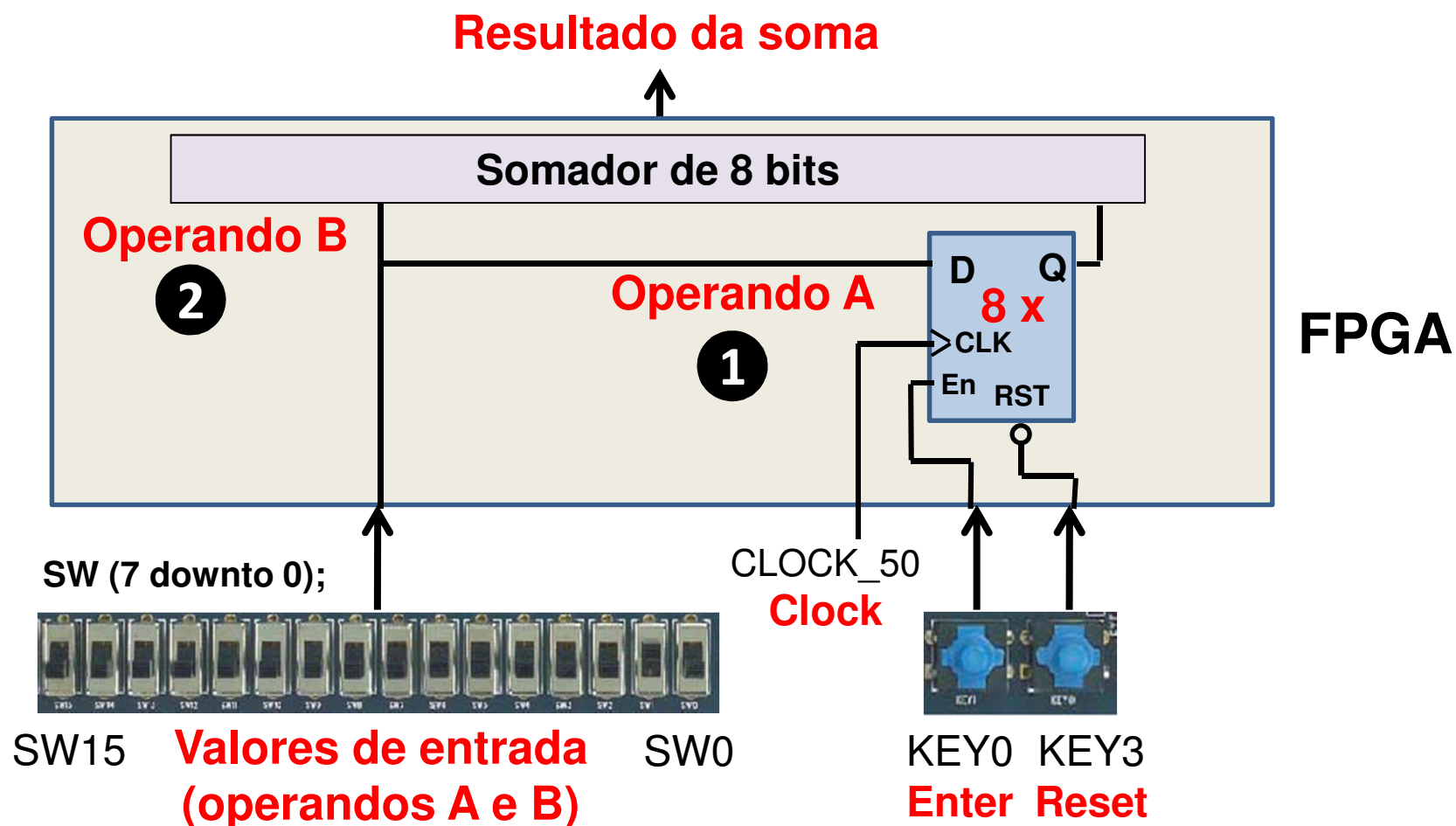
Operando A
Operando B
SW(7..0)

Reset
Key(3)

Enter
Key(0)

Estudo de caso: Calculadora do Lab. 8

Para a entrada dos dois operandos (A e B), com apenas 8 chaves, um novo registrador armazena o operando A, e o mesmo conjunto de chaves pode ser usado para leitura do operando B.



Tarefa a ser realizada na aula prática

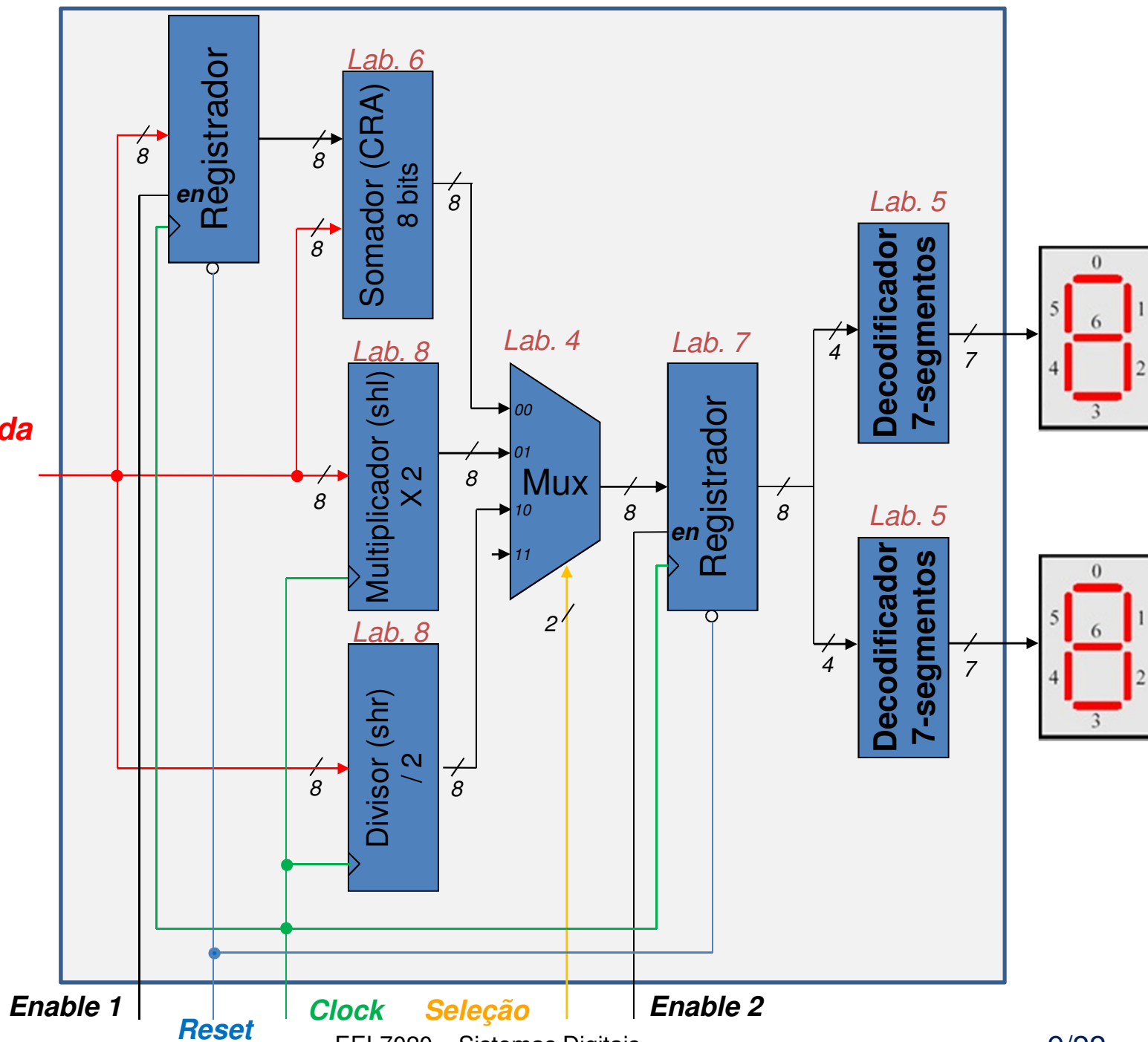
Controle do fluxo de operações da calculadora

- Utilizar como base a calculadora do Lab.8, e alterar o arquivo top (Calculadora.vhd), incluindo um registrador na entrada, de forma a utilizar apenas 8 chaves para leitura de ambos operandos para operações de soma.
- Criar uma FSM de controle, a ser incluída no top, conforme apresentado nos diagramas de blocos dos slides a seguir.

Controle do fluxo de operações da calculadora

- No slide a seguir é apresentada uma **sugestão de modificação** na calculadora, de forma a incluir o registrador adicional.
- Algumas observações:
 - **Não é necessário alterar o VHDL de nenhum componente.**
 - No arquivo “Calculadora.vhd”, deve ser criada uma cópia do componente Registrador (novo registrador).
 - Devem ser realizadas **alterações em alguns *port maps***, visando a conexão do novo registrador aos demais componentes existentes, e também às entradas.
- Notar também que os **sinais de habilitação** de registradores, seleção de multiplexador, entre outros, foram desconectados, visando a **inclusão de um controle automático do fluxo de operações** por intermédio de uma **FSM**.

**Dados de entrada
(operandos)**
SW(7 downto 0)

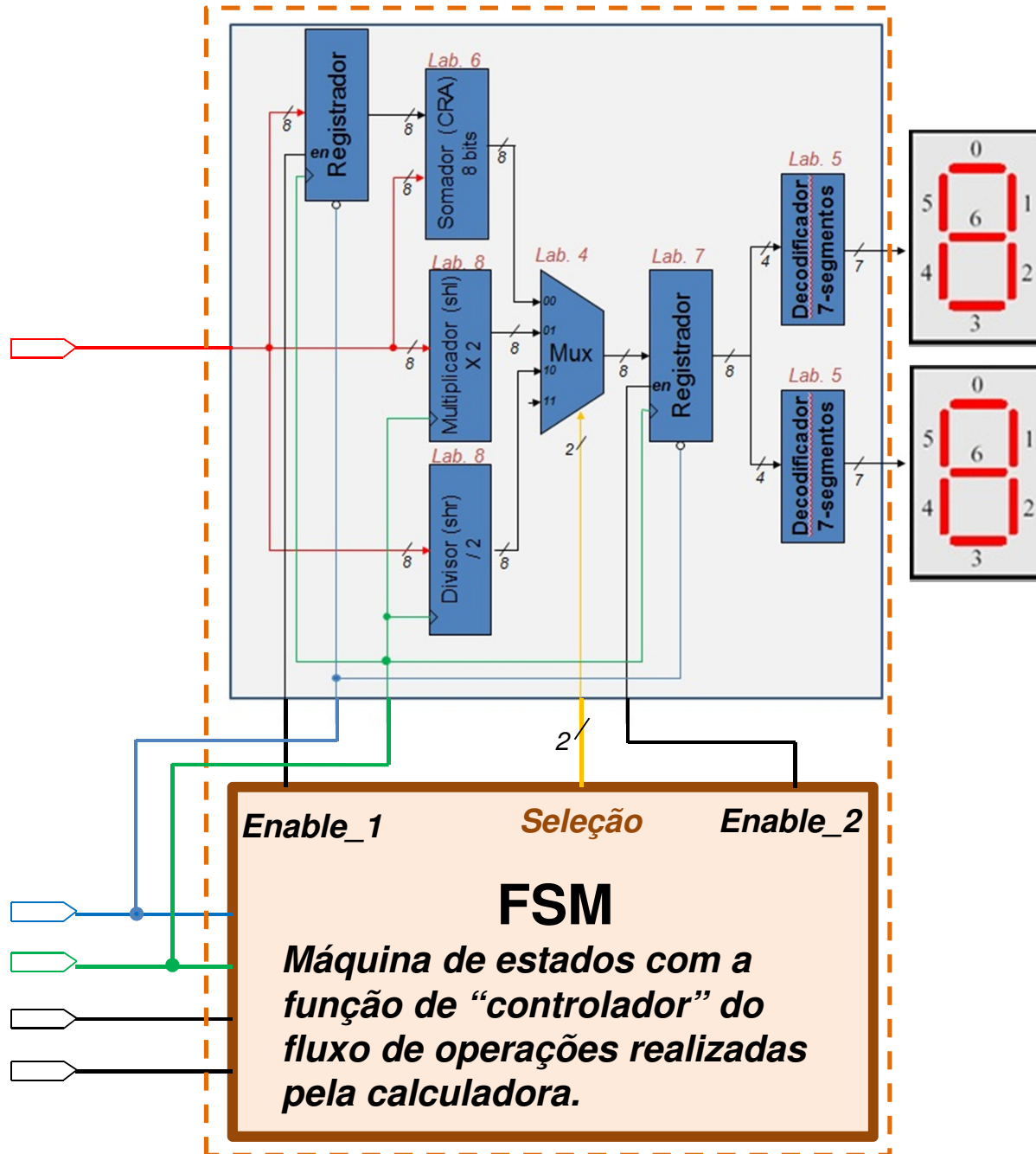


Controle do fluxo de operações da calculadora

- No slide a seguir é apresentado o diagrama de blocos do circuito, incluindo o controlador (FSM) para gerenciar o fluxo de operações.
- Sugestão de interface para o novo componente (FSM):

```
component FSMctrl  
  port ( Enter   : IN STD_LOGIC;  
         Clk     : IN STD_LOGIC;  
         Rst     : IN STD_LOGIC;  
         Operacao : IN STD_LOGIC_VECTOR(1 downto 0);  
         Enable_1 : OUT STD_LOGIC;  
         Enable_2 : OUT STD_LOGIC;  
         Selecao  : OUT STD_LOGIC_VECTOR(1 downto 0)  
    );  
end component;
```

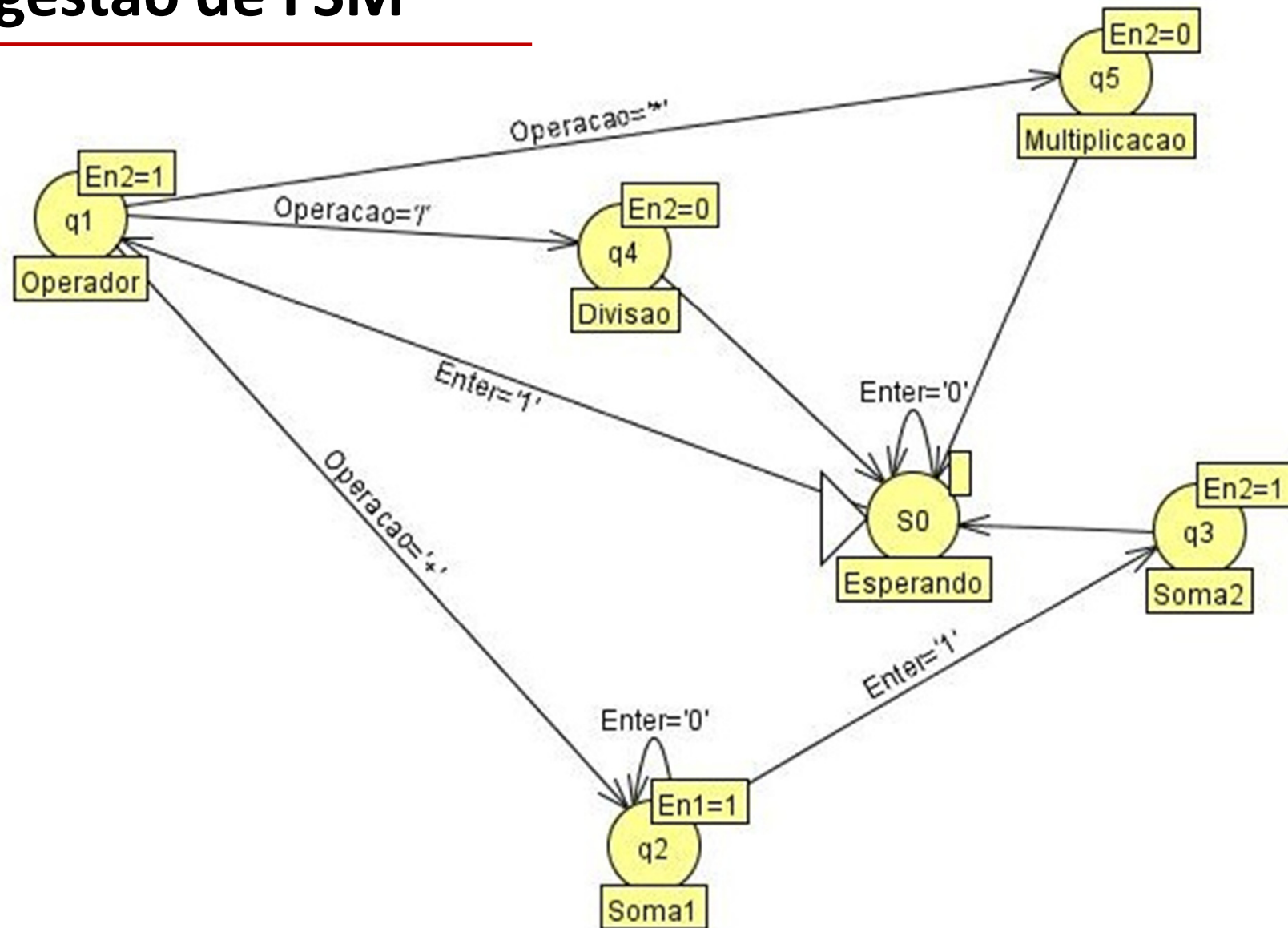
**Dados de entrada
(operandos)**
SW(7 downto 0)



Funcionamento da FSM (controlador)

- No estado inicial, “Esperando”, as três saídas da FSM estão desabilitadas (Enable_1='0', Enable_2='0', Selecao="11").
- Isso garante que nenhuma atividade ocorrerá na calculadora, enquanto o usuário não fornecer um operando e um operador.
- Quando Enter for '1', a FSM avança para o estado “Operador”.
- Nesse estado, a “Operação” selecionada é identificada, e a saída “Seleção” é ativada (Mux) com a função desejada.
- O próximo estado depende da operação selecionada. Se for uma multiplicação ou divisão, a operação é realizada, o resultado é armazenado no registrador de saída (Enable_2='1') e a seguir a FSM retorna para “Esperando”.
- Se for uma soma, será necessário um estado adicional para leitura do segundo operando (Enable_1='1', seguido por Enable_2='1').
- A seguir, a FSM retorna para “Esperando”.

Sugestão de FSM



*Sugestão para
implementação da
FSM descrita.*

```
library ieee;
use ieee.std_logic_1164.all;
entity FSMctr is
port (
  Clk, Rst, Enter, Operacao: in std_logic;
  Enable_1, Enable_2: out std_logic;
  Sel: out std_logic_vector(1 downto 0)
);
```

```
end FSMctr;
architecture FSM_beh of FSMctr is
  type states is (S0, S1, S2, S3);
  signal EA, PE: states;
  signal clock: std_logic;
  signal reset: std_logic;
begin
  clock <= Clk;
  reset <= Rst;
```

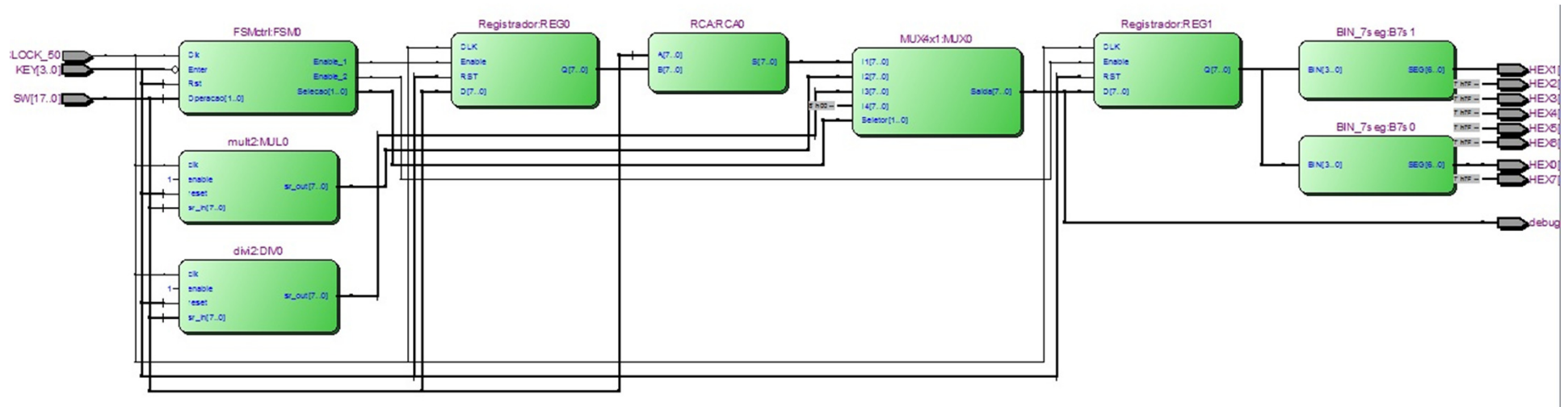
```
process (clock, reset)
begin
  if reset = '0' then
    EA <= S0;
  elsif clock'event and
        clock = '1' then
    EA <= PE;
  end if;
end process;
```

```
process (EA, Enter)
begin
  case EA is
    when S0 => if Enter = '0' then
      PE <= S0; else PE <= S1;
    end if;
    Enable_2='0';
    when S1 => -- Operador
      Enable_1 <= '0';
      if Operacao = "00" then
        Selecao = "00";
        PE <= S2;
      elsif Operacao = "01" then
        ...
      end case;
    end process;
  end FSM_beh;
```

Controle do fluxo de operações da calculadora

- Para utilizar a calculadora é necessário:
 1. Selecionar a operação desejada nas chaves SW(17..16).
 2. Fornecer um valor nas chaves SW(7..0) - operando A.
 3. Pressionar *Enter* - o botão KEY(0).
 4. Se for operação de multiplicação ou divisão, o resultado será apresentado nos displays de 7-segmentos.
 5. Se for operação de soma, fornecer o segundo operando nas chaves SW(7..0), e pressionar *Enter*.
 6. Após apresentado o resultado, essa sequência é reiniciada, voltando ao passo 1.

Diagrama de blocos da calculadora modificada



Simulação com *ModelSim*

Simulação do projeto com ModelSim

1. Criar uma nova pasta dentro da pasta do projeto.
2. Copiar os **scripts de simulação** disponíveis na página da disciplina para dentro da nova pasta.
3. Entrar na nova pasta, editar o arquivo "*compila.do*", e alterar **top.vhd** (e demais nomes de arquivos VHDL do projeto) para o nome do seu arquivo VHDL a ser simulado.
4. Copiar APENAS os arquivos VHDL a serem simulados para a nova pasta, que já deve possuir os *scripts* de simulação copiados da página da disciplina.
5. Executar o *ModelSim-Altera*, que se encontra no menu *Iniciar* do Windows, pasta "*Altera*".
6. No menu "*File*" do *ModelSim*, definir a pasta do projeto (opção "*Change Directory*"), selecionando a nova pasta.

Simulação do projeto com ModelSim (cont.)

7. Execução da simulação (arquivo *compila.do*)
 - a) No menu "Tools" do ModelSim, selecionar "*Tcl*" -> "*Execute Macro*".
 - b) Selecionar o arquivo "*compila.do*", e "*Open*".
8. O *ModelSim* irá compilar os arquivos VHDL e iniciar a simulação.
9. A janela com as formas de onda irá abrir, apresentando o resultado da simulação.

Obs. Se desejar, editar o arquivo *tb.vhd* para alterar a simulação a ser realizada, e repetir o passo 7.

Simulação do projeto com ModelSim (cont.)

Obs. Se o resultado da simulação não estiver de acordo com o esperado, alterar o seu VHDL, salvar, e executar novamente a simulação (arquivo *compila.do*).

Obs. A simulação só irá funcionar se o seu projeto possuir **EXATAMENTE** a do próximo slide.

Simulação do projeto com ModelSim (cont.)

```
entity top is port (  
    CLOCK_50: in std_logic;  
    SW : IN STD_LOGIC_VECTOR(17 downto 0);  
    KEY : IN STD_LOGIC_VECTOR(3 downto 0);  
    HEX0 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    HEX1 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    HEX2 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    HEX3 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    HEX4 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    HEX5 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    HEX6 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    HEX7 : OUT STD_LOGIC_VECTOR(6 downto 0);  
    debug: OUT STD_LOGIC_VECTOR(7 downto 0)  
);  
end top;
```

Simulação do projeto com ModelSim (cont.)

Obs. O arquivo "*compila.do*" contém os comandos do *ModelSim* necessários para realizar a simulação, incluindo:

- a) Criação da biblioteca de trabalho - comando *vlib*.
- b) Compilação dos arquivos VHDL para a biblioteca de trabalho - comando *vcom*.
- c) Inicialização do simulador com o arquivo *testbench* - comando *vsim*.
- d) Execução da janela de formas de onda (*waveform*) - comando *wave*.
- e) Adição dos sinais na janela de formas de onda - comando *wave*.
- f) Execução da simulação - comando *run*.