

Projeto

- Implementar em VHDL os componentes que seguem:
 - Um registrador;
 - Um contador nódulo 10;
 - Uma máquina de vender bombom;
- Os projetos devem ser apresentados até dia 7/2/08, sem falta.
- Os projetos do contador e da máquina de vende bombom deverão ser também implementadas na plataforma física da Altera, até o dia 27/08.
- Todas das entradas feitas através de botões deverão ser protegidas por circuito anti-ruído(debounce). Este circuito será fornecido.



Registrador de 4 bits

■ Funcionalidades

- Reset
- load

-- Registrador

entity reg_4bits is

```
port ( A : in bit_vector (3 downto 0);  
      clk, reset, load:in bit;  
      C : out bit_vector (3 downto 0)
```

```
);
```

```
end reg_4bits;
```

architecture arc_reg of reg_4bits is

```
begin
```

```
process (clk)
```

```
begin
```

```
if (reset = '1') then
```

```
  C <= "0000";
```

```
elsif (clk = '1' and clk'event) then
```

```
  if (load = '1') then
```

```
    C <= A;
```

```
  end if;
```

```
end if;
```

```
end process;
```

```
end arc_reg;
```



Contador

- Contador de 4 bits controlado pelos sinais clk, clear, ld, d, enable;

- Entidade

Entity counter is

port (

d : in integer range 0 to 9;

clk : in bit;

clear : in bit;

ld : in bit;

q : out integer range 0 to 9

);

end counter;

- Arquitetura do contador

Architecture a of counter is

begin

-- clear, load síncronos

Process(clk)

variable cnt : integer range 0 to 9;

begin

if(clk'event and clk = '1') then

if clear = '0' then

cnt := '0';

else

if ld = '0' then

cnt := d;

else

cnt := cnt+1;

end if;

end if;

end if;

q <= cnt;

end process;

end a;



Máquina de vender bombom

- Implementar a unidade de controle de uma máquina de vender bombom, em VHDL (código descrito abaixo), com as características abaixo:
 - A máquina libera um pacote de bomboms se o cliente depositar pelo menos 15 centavos.
 - A máquina não dá trôco.
 - Moedas aceitas na máquina:
 - 5 centavos (C)
 - 10 centavos (D)
 - Implemente a mesma máquina permitindo que o cliente possa reutilizar os 5 centavos que ficam presos caso duas moedas de 10 sejam inseridas.
 - Os valores das moedas, valor acumulado (inserido ou restante), deverá ser apresentado nos displays de 7-segmentos.

Máquina de vender bombom

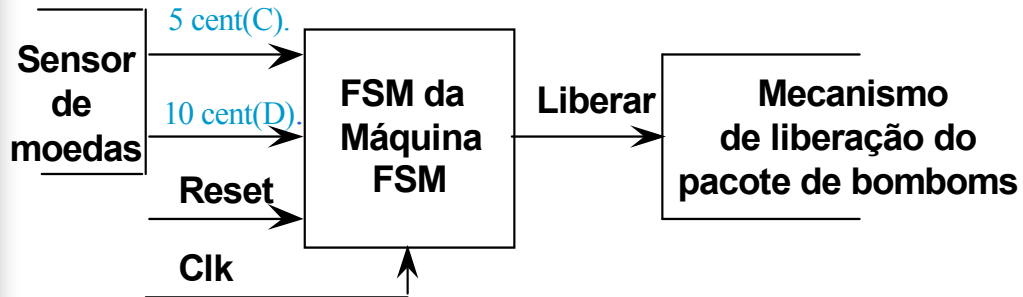
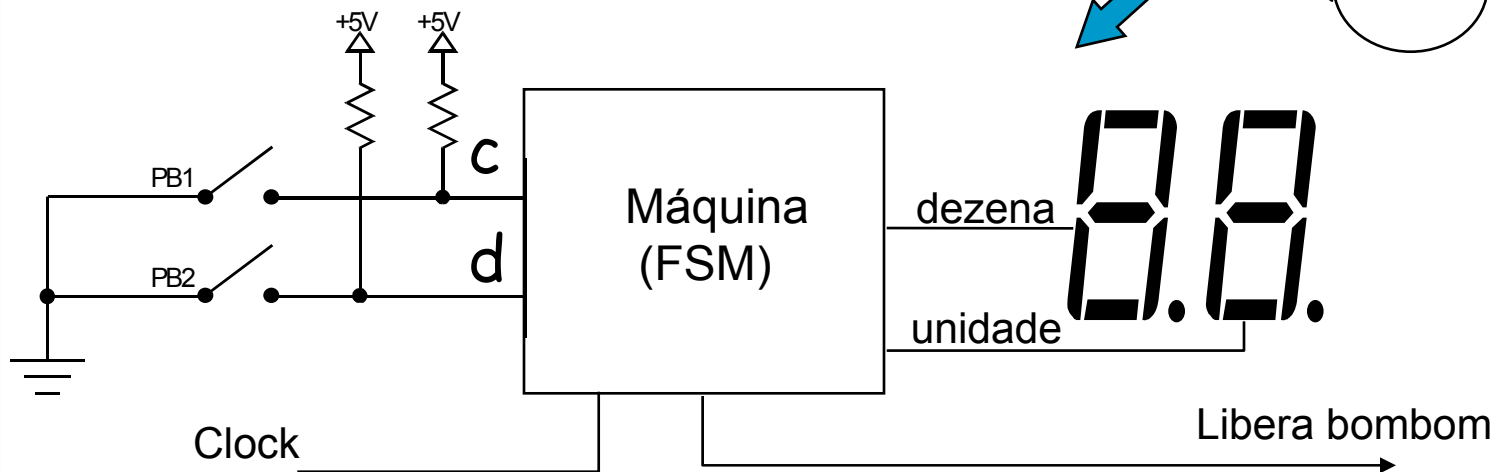
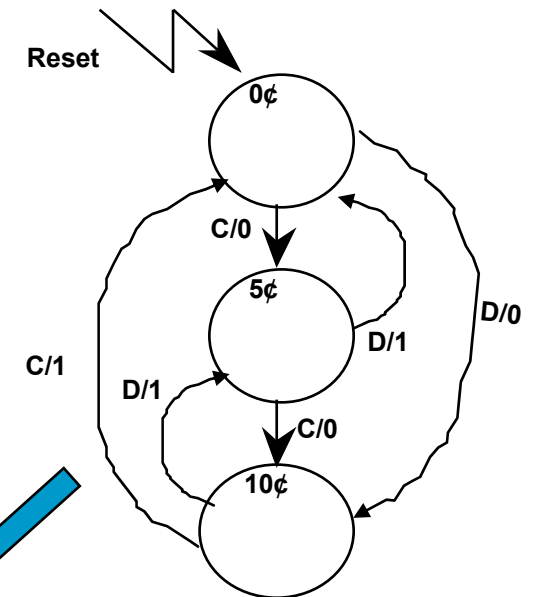


Diagrama de estados



Máquina de vender bombom

```
-- Máquina de vender bombom

entity bombom01 is
port( reset, clk, c, d : in bit;
      libera_bombom: out bit );
end bombom01;
architecture arc_bombons of bombom01 is
-- Definição dos estado da máquina
type estados is (tem_zero, tem_cinco, tem_dez);
signal estado_atual : estados;
begin
    process (clk)
begin
    if (reset = '1') then
        estado_atual <= tem_zero;
    elsif (clk = '1' and clk'event) then
        case estado_atual is
            when tem_zero =>
                libera_bombom <= '0';
                if (c = '1' ) then
                    estado_atual <= tem_cinco;
                elsif (d = '1') then
                    estado_atual <= tem_dez;
                end if;
            end case;
        end if;
    end process;
end arc_bombons;
```

```
-- Máquina de vender bombom

when tem_cinco =>
    libera_bombom <= '0';
    if (c = '1') then
        estado_atual <= tem_dez;
    elsif (d = '1') then
        estado_atual <= tem_zero;
        libera_bombom <= '1';
    end if;
when tem_dez =>
    if (c = '1') then
        estado_atual <= tem_zero;
        libera_bombom <= '1';
    elsif(d = '1') then
        estado_atual <= tem_cinco;
        libera_bombom <= '1';
    end if;
    end case;
end if;
end process;
end arc_bombons;
```



Circuito "debounce"

```
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.all;
USE ieee.STD_LOGIC_UNSIGNED.all;
-- Title "Key debounce circuit";
-- Prepared by: D. N. Warren-Smith
-- Updated: 7 February 2001

ENTITY DEBOUNCE IS
PORT (
  Clk    : IN STD_LOGIC;
  Key    : IN STD_LOGIC; -- active low input
  pulse  : OUT STD_LOGIC);
END DEBOUNCE;
ARCHITECTURE clean_pulse OF DEBOUNCE IS
  SIGNAL cnt    : STD_LOGIC_VECTOR (17 DOWNTO 0);
BEGIN
  PROCESS (Clk)
  BEGIN
    IF Key = '1' THEN
      cnt <= "000000000000000000";
    ELSIF (clk'EVENT AND Clk = '1') THEN
      IF (cnt /= "1111111111111111") THEN cnt <= cnt + 1; END IF;
    END IF;
    IF (cnt = "11111111111111110") AND (Key = '0') THEN pulse <= '1'; ELSE pulse <= '0'; END IF;
  END PROCESS;
END clean_pulse;
```