MC613

Laboratório de Circuitos Lógicos

IC/UNICAMP

Profs.:

Guido Araújo (guido @ ic ….)

Mário Lúcio Cortes (cortes @ ic…..)

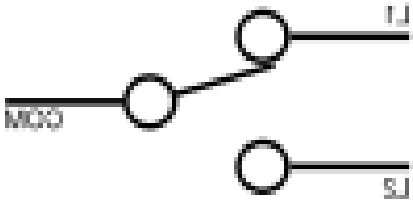
MC613

Debouncer

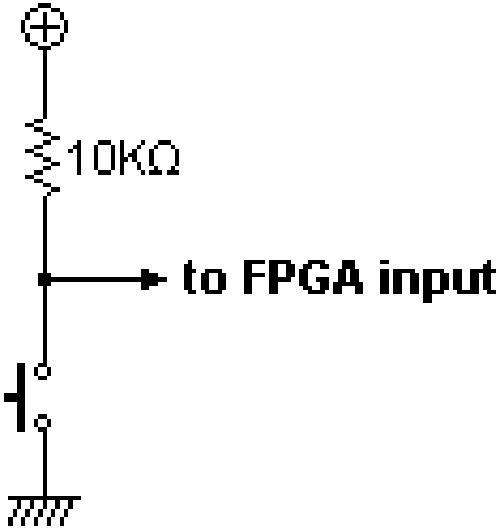
Conteúdo

- O Problema
  - SPDT – Single Pole, Double Throw
  - SPST – Single Pole, Single Throw
- SPDT – Solução
- SPST – Solução
  - Circuito RC
  - Digital - Contador (freqüência de operação)
  - Digital - Shift-Register (baixa freqüência)

O Problema



SPDT



SPST

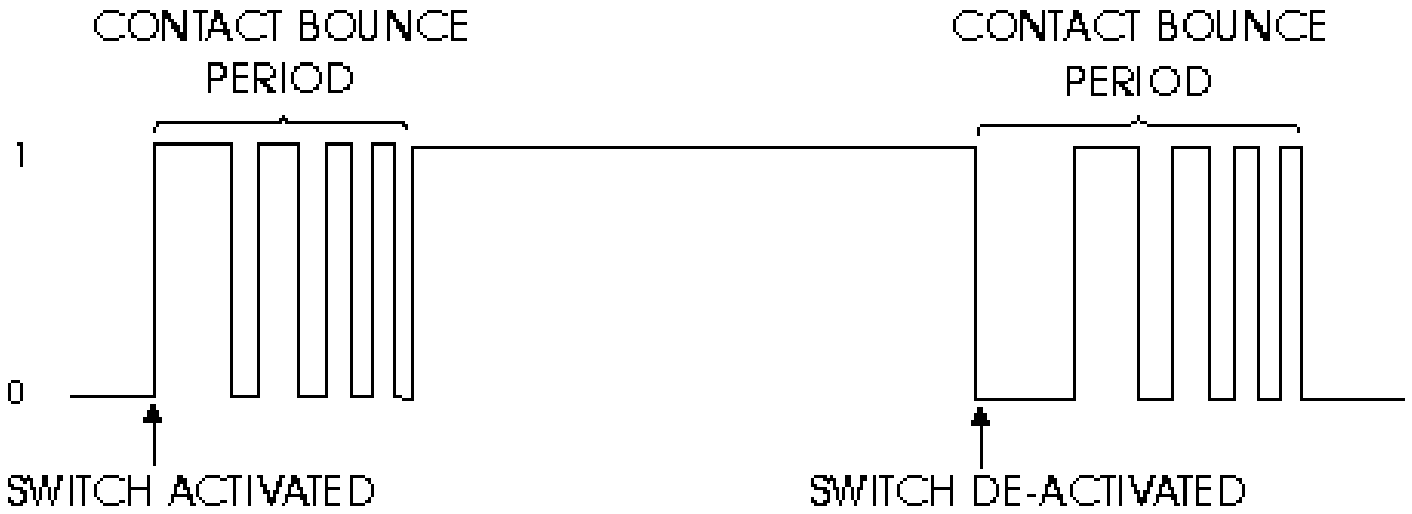


Figure 1

SPDT - Solução

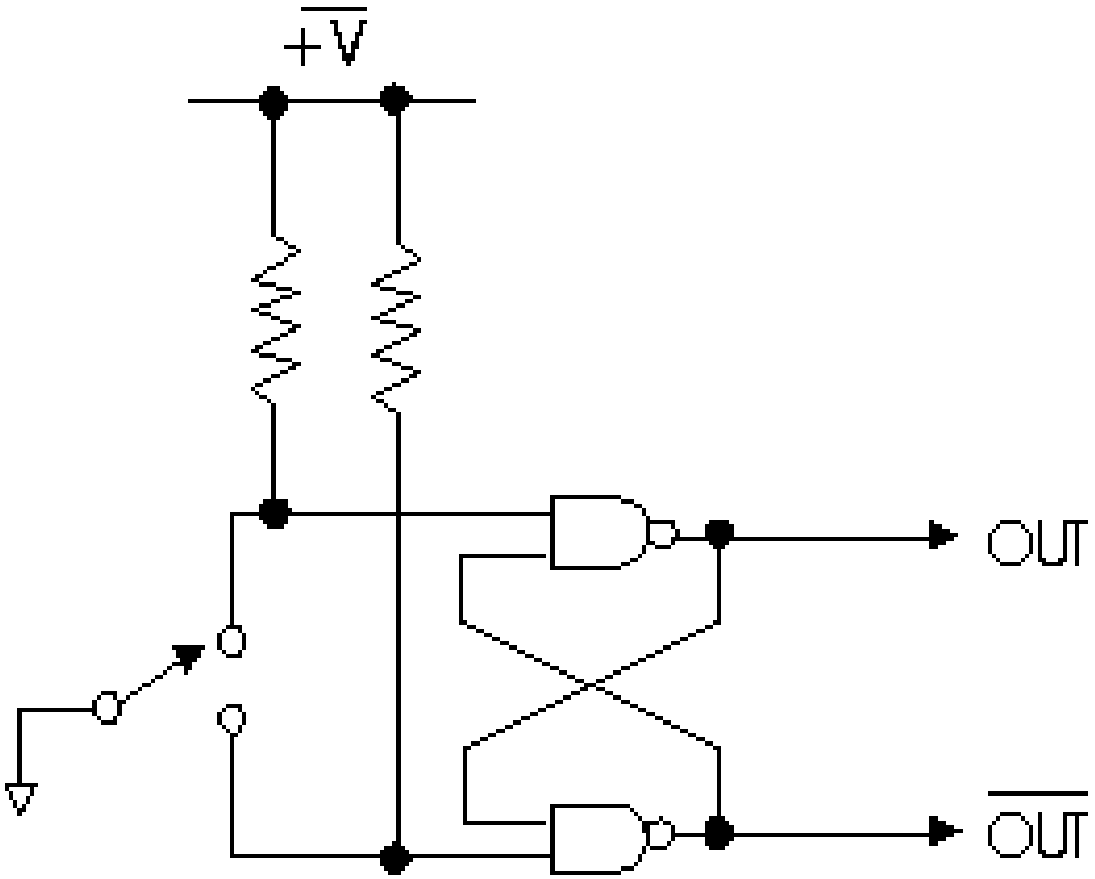


Figure 3

SPST – Solução
Circuito RC

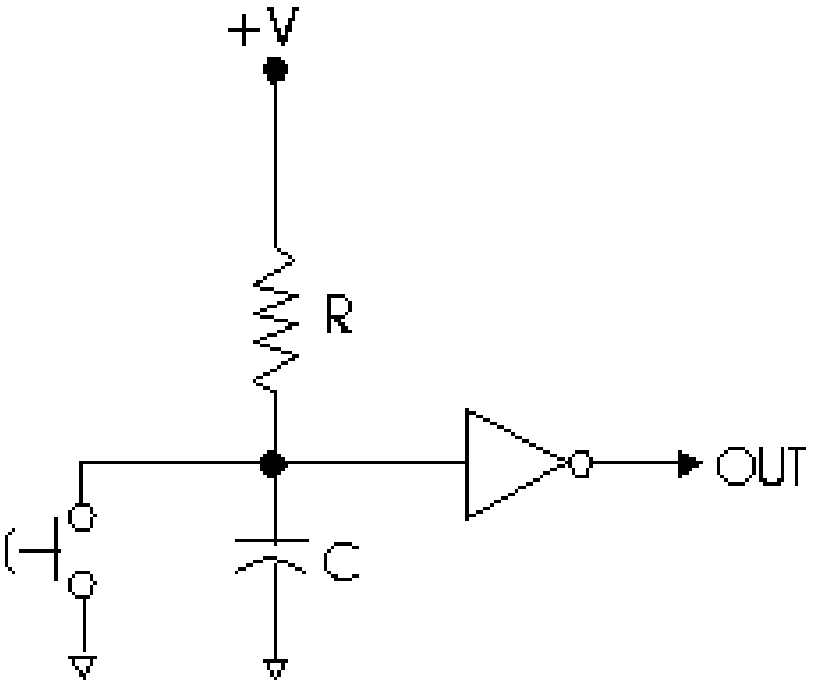
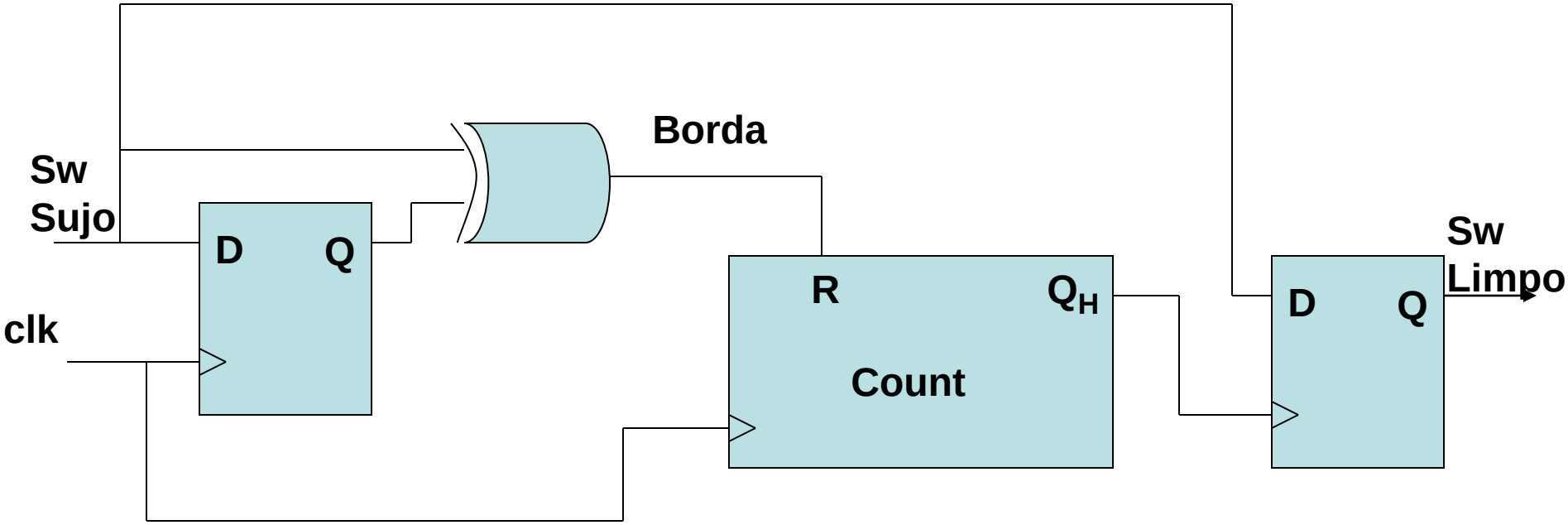


Figure 2

SPST – Solução
Digital – freqüência de operação



SPST – Solução
VHDL - baixa freqüência

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_ARITH.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;

-- Debounce Pushbutton: Filters out mechanical switch
-- bounce for around 40Ms.

ENTITY debounce IS
    PORT(pb, clock_100Hz : IN STD_LOGIC;
         pb_debounced    : OUT STD_LOGIC);
END debounce;
```

SPST – Solução
VHDL (cont.)

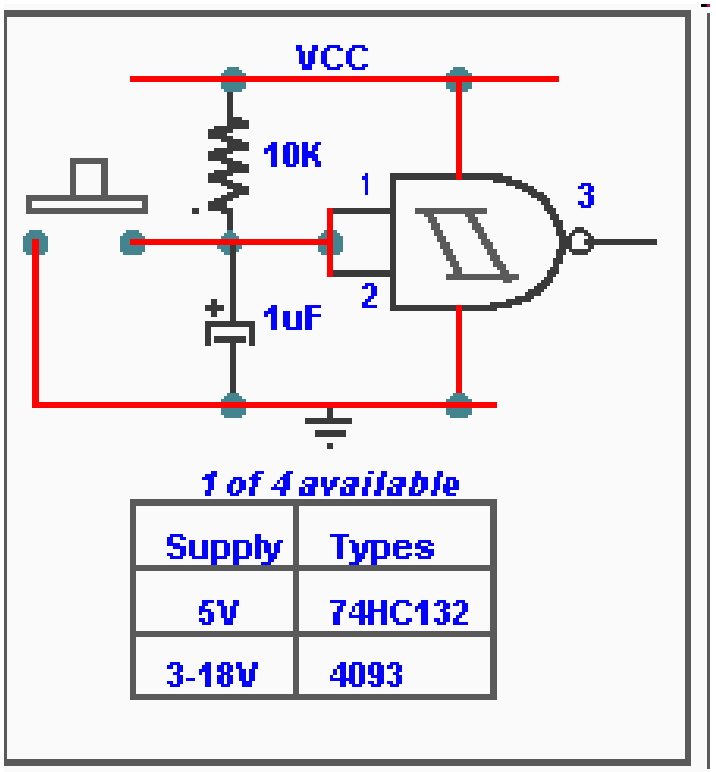
```
ARCHITECTURE a OF debounce IS
   SIGNAL SHIFT_PB : STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN -- Debounce clock should be approximately 10ms or 100Hz

   PROCESS
   BEGIN
      WAIT UNTIL (clock_100Hz'EVENT) AND (clock_100Hz = '1');
      -- Use a shift register to filter switch contact bounce
      SHIFT_PB(2 DOWNTO 0) <= SHIFT_PB(3 DOWNTO 1);
      SHIFT_PB(3) <= NOT PB;
      IF SHIFT_PB(3 DOWNTO 0) = "0000"
        THEN PB_DEBOUNCED <= '0';
        ELSE PB_DEBOUNCED <= '1';
     END IF;
   END PROCESS;
END a;
```

Debouncer na placa DE1

- Push buttons
  - sinais já limpos com Schmitt Trigger

concepção →



- Toggle switches
  - sinal sujo → necessário passar por debouncer se for usado como clock