



Circuitos Digitais II - 6882

Paulo Roberto de Oliveira

Universidade Estadual de Maringá
Departamento de Informática

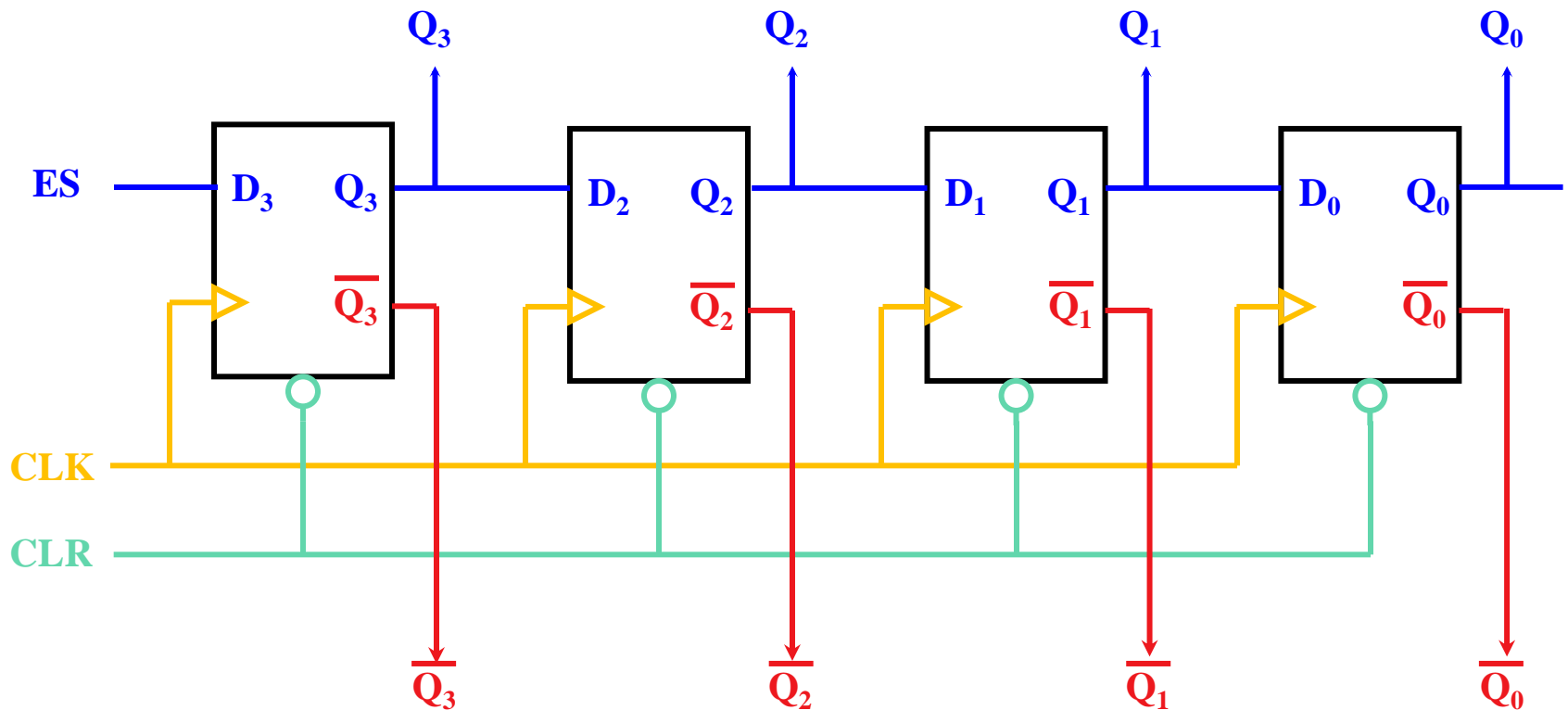
Bacharelado em Ciência da Computação

Aula de Hoje

- Registrador de Deslocamento
- Contador Assíncrono
- Contador Síncrono

Registrador de Deslocamento

Registrador de 4 Bits - Arquitetura Comportamental



VHDL - Código

○ Solução: Arquitetura Comportamental

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY reg_desl IS
    PORT(clr, clk, shiftin : IN STD_LOGIC;
          shiftout : OUT STD_LOGIC_VECTOR(3 DOWNT0 0));
END reg_desl;
--shiftin corresponde a ES e shiftout a Qi na figura
ARCHITECTURE comportamental OF reg_desl IS
    SIGNAL shift_bit : STD_LOGIC_VECTOR(3 DOWNT0 0);
BEGIN
    PROCESS(clr, clk)
    BEGIN
        IF (clr = '0') THEN
            shift_bit <= "0000";
```

VHDL - Código

○ Solução: Arquitetura Comportamental (continuação)

```
ELSIF (clk'EVENT AND clk = '1') THEN
```

```
    shift_bit(3) <= shiftin;
```

```
    shift_bit(2) <= shift_bit(3);
```

```
    shift_bit(1) <= shift_bit(2);
```

```
    shift_bit(0) <= shift_bit(1);
```

```
END IF;
```

```
END PROCESS;
```

```
shiftout(3) <= shift_bit(3);
```

```
shiftout(2) <= shift_bit(2);
```

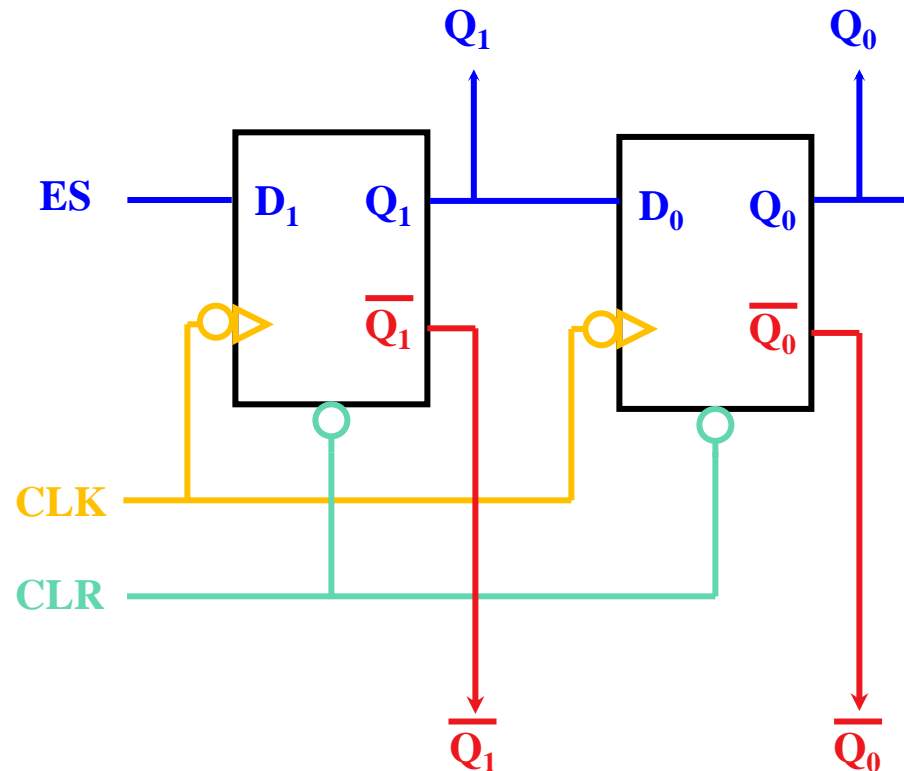
```
shiftout(1) <= shift_bit(1);
```

```
shiftout(0) <= shift_bit(0);
```

```
END comportamental;
```

Registrador de Deslocamento

Registrador de 2 Bits - Arquitetura Estrutural



VHDL - Código

○ Solução: Arquitetura Estrutural

Criação do Componente ffd_c (Flip-Flop D com Entrada Clear)

➤ Declaração da Entidade

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY ffd_c is  
    PORT(clr, clk, d : IN STD_LOGIC;  
          q, qbar : BUFFER STD_LOGIC);  
END ffd_c;
```

VHDL - Código

- Solução: Arquitetura Estrutural (continuação)
Criação do Componente ffd_c (Flip-Flop D com Entrada Clear)
 - Arquitetura da Entidade

```
ARCHITECTURE comportamental OF ffd_c IS
BEGIN
    PROCESS(clr, clk, d)
    VARIABLE qv,qbarv : STD_LOGIC;
    BEGIN
        IF (clr = '0') THEN
            qv := '0';
            qbarv := NOT qv;
        ELSIF (falling_edge(clk)) THEN
            qv := d;
            qbarv := NOT qv;
        END IF;
        q <= qv;
        qbar <= NOT q;
    END PROCESS;
END comportamental;
```


VHDL - Código

○ Solução: Arquitetura Estrutural (continuação)

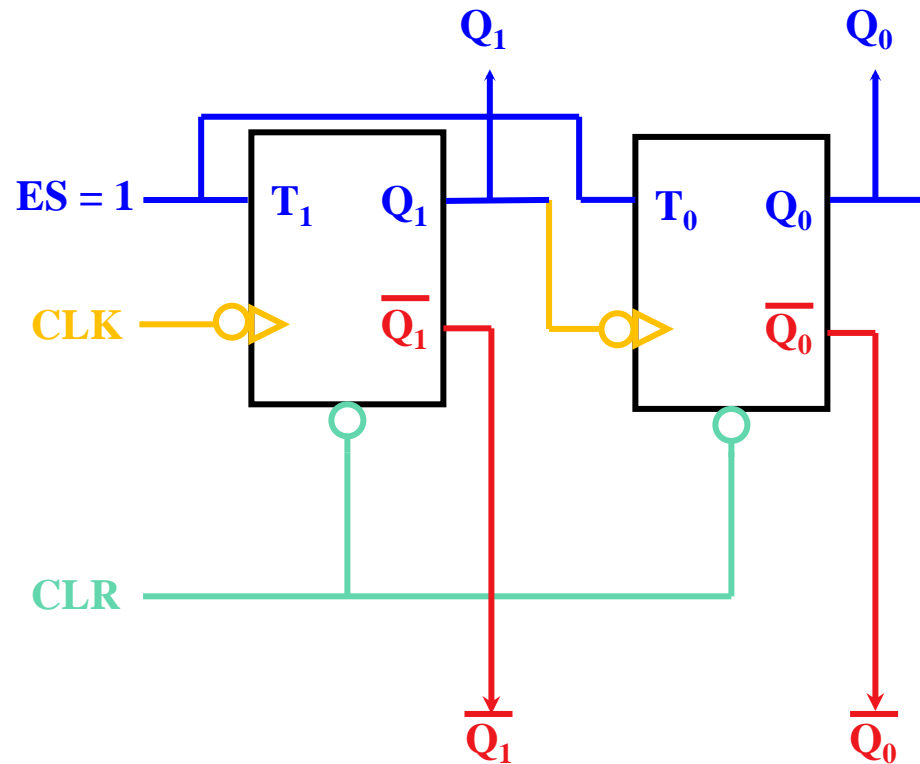
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY reg_desl_est IS
    PORT(clr, clk, es : IN STD_LOGIC;
         q_out, qb_out : BUFFER STD_LOGIC_VECTOR(1 DOWNTO 0));
END reg_desl_est;

ARCHITECTURE estrutural OF reg_desl_est IS
    COMPONENT ffd_c IS
        PORT(clr, clk, d : IN STD_LOGIC;
             q, qbar : BUFFER STD_LOGIC);
    END COMPONENT;
BEGIN
    ffd_c1 : ffd_c PORT MAP(clr, clk, es, q_out(1), qb_out(1));
    ffd_c0 : ffd_c PORT MAP(clr, clk, q_out(1), q_out(0), qb_out(0));
END estrutural;
```

Contador Assíncrono

Contador Assíncrono de Módulo 4 - Arquitetura Estrutural



VHDL - Código

○ Solução: Arquitetura Estrutural

Criação do Componente fft_c (Flip-Flop T com Entrada Clear)

➤ Declaração da Entidade

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY fft_c IS  
    PORT(clr, clk, t : IN STD_LOGIC;  
         q, qbar : BUFFER STD_LOGIC);  
END fft_c;
```

VHDL - Código

○ Solução: Arquitetura Estrutural (continuação)

Criação do Componente fft_c
(Flip-Flop T com Entrada Clear)

➤ Arquitetura da Entidade

```
ARCHITECTURE comportamental OF fft_c IS
BEGIN
    PROCESS(clr, clk, t)
    VARIABLE qv,qbarv : STD_LOGIC;
    BEGIN
        IF (clr = '0') THEN
            qv := '0';
            qbarv := NOT qv;
        ELSIF (falling_edge(clk)) THEN
            IF (t = '1') THEN
                qv := NOT qv;
                qbarv := NOT qv;
            ELSE
                qv := qv;
                qbarv := NOT qv;
            END IF;
        END IF;
        q <= qv;
        qbar <= qbarv;
    END PROCESS;
END comportamental;
```

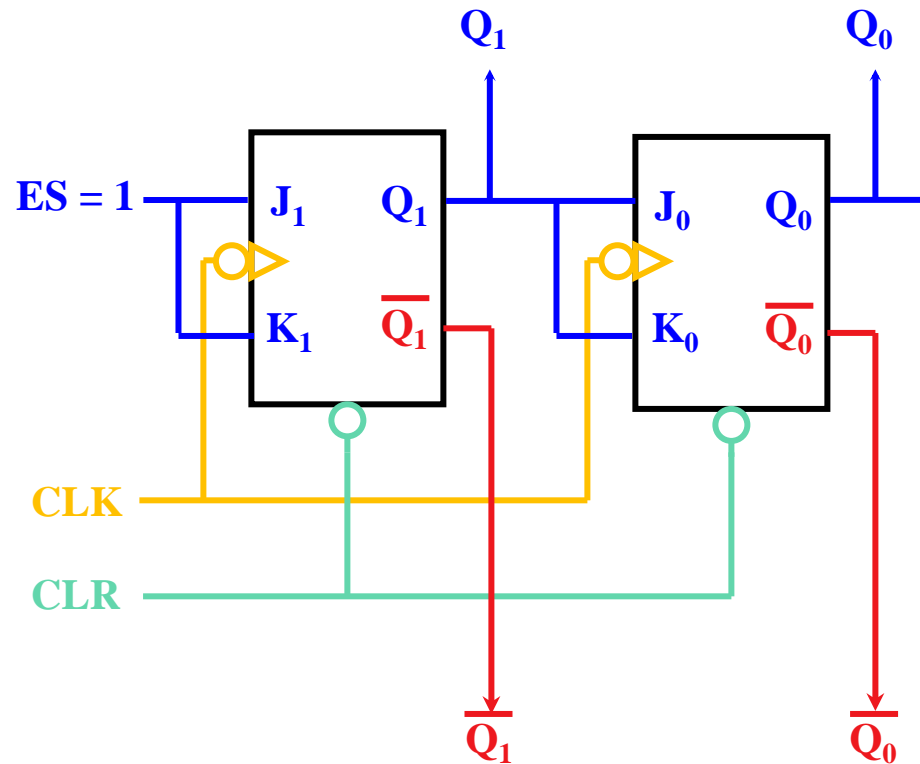
VHDL - Código

○ Solução: Arquitetura Estrutural (continuação)

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY cont_assinc_mod4 IS  
    PORT(clr, clk, es : IN STD_LOGIC;  
          q_out, qb_out : BUFFER STD_LOGIC_VECTOR(1 DOWNT0 0));  
END cont_assinc_mod4;  
  
ARCHITECTURE estrutural OF cont_assinc_mod4 IS  
    COMPONENT fft_c IS  
        PORT(clr, clk, t : IN STD_LOGIC;  
              q, qbar : BUFFER STD_LOGIC);  
    END COMPONENT;  
    BEGIN  
        fft_c1 : fft_c PORT MAP(clr, clk, es, q_out(1), qb_out(1));  
        fft_c0 : fft_c PORT MAP(clr, q_out(1), es, q_out(0), qb_out(0));  
    END estrutural;
```

Contador Síncrono

Contador Síncrono de Módulo 4 - Arquitetura Estrutural



VHDL - Código

○ Solução 01: Arquitetura Estrutural

Criação do Componente ffjk_c (Flip-Flop JK com Entrada Clear)

➤ Declaração da Entidade

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY ffjk_c IS  
    PORT(clr, clk, j, k : IN STD_LOGIC;  
         q, qbar : BUFFER STD_LOGIC);  
END ffjk_c;
```

VHDL - Código

○ Solução 01: Arquitetura Estrutural (continuação)

Criação do Componente ffjk_c (Flip-Flop JK com Entrada Clear)

➤ Arquitetura da Entidade

```
ARCHITECTURE comportamental OF ffjk_c IS
BEGIN
    PROCESS(clr, clk, j, k)
    VARIABLE qv,qbarv : STD_LOGIC;
    BEGIN
        IF (clr = '0') THEN
            qv := '0';
            qbarv := NOT qv;
        ELSIF (falling_edge(clk)) THEN
            IF (j = '1' AND k = '0') THEN
                qv := '1';
                qbarv := NOT qv;
            ELSIF (j = '0' and k = '1') THEN
                qv := '0';
                qbarv := NOT qv;
            
```


VHDL - Código

○ Solução 01: Arquitetura Estrutural (continuação)

Criação do Componente ffjk_c (Flip-Flop JK com Entrada Clear)

➤ Arquitetura da Entidade

```
ELSIF (j = '1' and k = '1') THEN
    qv := NOT qv;
    qbarv := NOT qv;
ELSE
    qv := qv;
    qbarv := NOT qv;
END IF;
END IF;
q <= qv;
qbar <= qbarv;
END PROCESS;
END comportamental;
```

VHDL - Código

○ Solução 01: Arquitetura Estrutural (continuação)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY cont_sinc_mod4 IS
    PORT(clr, clk, es : IN STD_LOGIC;
         q_out, qb_out : BUFFER STD_LOGIC_VECTOR(1 DOWNTO 0));
END cont_sinc_mod4;

ARCHITECTURE estrutural OF cont_sinc_mod4 IS
    COMPONENT ffjk_c IS
        PORT(clr, clk, j, k : IN STD_LOGIC;
             q, qbar : BUFFER STD_LOGIC);
    END COMPONENT;
BEGIN
    ffjk_c1 : ffjk_c PORT MAP(clr, clk, es, es, q_out(1), qb_out(1));
    ffjk_c0 : ffjk_c PORT MAP(clr, clk, q_out(1), q_out(1), q_out(0), qb_out(0));
END estrutural;
```

VHDL - Código

○ Solução 02: Arquitetura Comportamental

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;                                -- *
ENTITY cont_mod4_ex IS
    PORT(clr, clk : IN STD_LOGIC;
          q, qbar : OUT STD_LOGIC_VECTOR(1 DOWNT0 0));
END cont_mod4_ex;
ARCHITECTURE comportamental OF cont_mod4_ex IS
BEGIN
    PROCESS(clr, clk)
        VARIABLE cont : STD_LOGIC_VECTOR(1 DOWNT0 0);
    BEGIN
        IF (clr = '0') THEN
            cont := "00";
        ELSIF (clk'EVENT AND clk = '0') THEN
            cont := cont + 1;                                -- *
        END IF;
        q <= cont;
        qbar <= NOT cont;
    END PROCESS;
END comportamental;
```

VHDL - Código

○ Solução 03: Arquitetura Comportamental

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY cont_mod4_ex1 IS
    PORT(clr, clk : IN BIT;
          q : OUT INTEGER RANGE 0 TO 3);
END cont_mod4_ex1;
ARCHITECTURE comportamental OF cont_mod4_ex1 IS
BEGIN
    PROCESS(clr, clk)
        VARIABLE cont : INTEGER RANGE 0 TO 3;
        BEGIN
            IF (clr = '0') THEN
                cont := 0;
            ELSIF (clk'EVENT AND clk = '0') THEN
                IF (cont < 3) THEN
                    cont := cont + 1;
                ELSE
                    cont := 0;
                END IF;
            END IF;
            q <= cont;
        END PROCESS;
    END comportamental;
```

Resumo da Aula de Hoje

Tópicos mais importantes:

- Registrador de Deslocamento
- Contador Assíncrono
- Contador Síncrono

Próxima da Aula

- Trabalho
 - Relógio Digital

