



IC-UNICAMP

MC 613

IC/Unicamp

Prof Guido Araújo

Prof Mario Côrtes

Circuitos Aritméticos

Tópicos

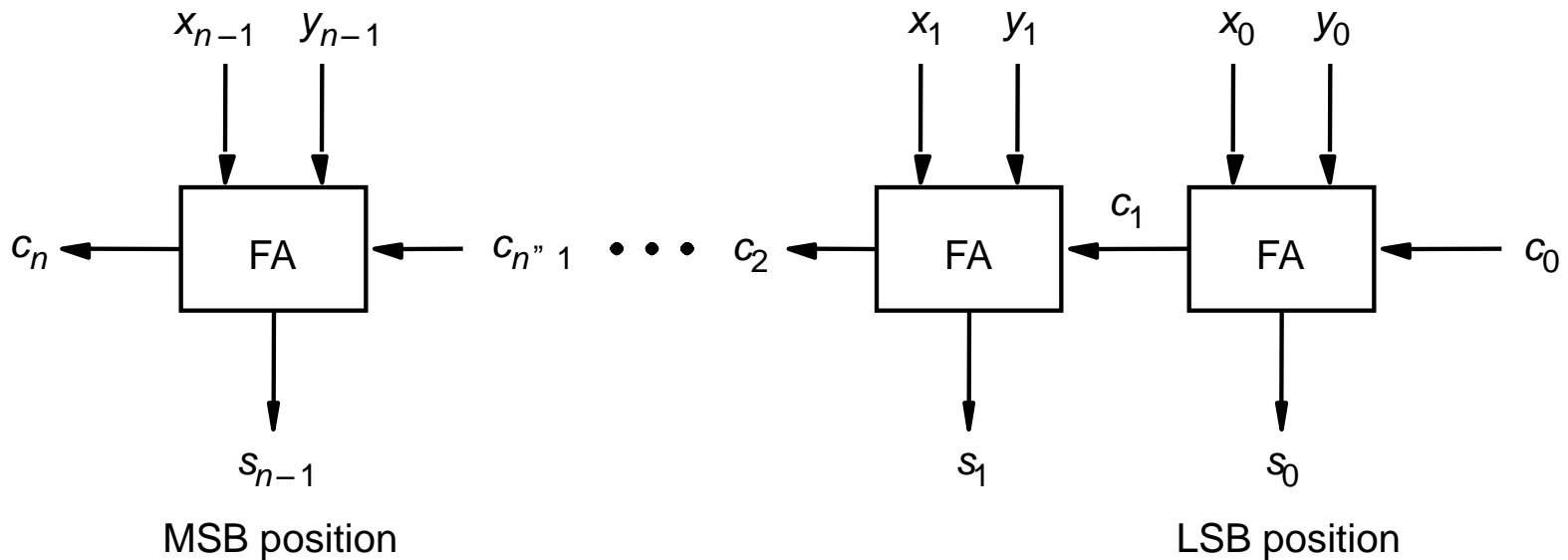
- Antecipação de carry (Carry Look Ahead)
- Soma de números em representação BCD

Somador Ripple Carry

- Atraso para um somador de n bits:

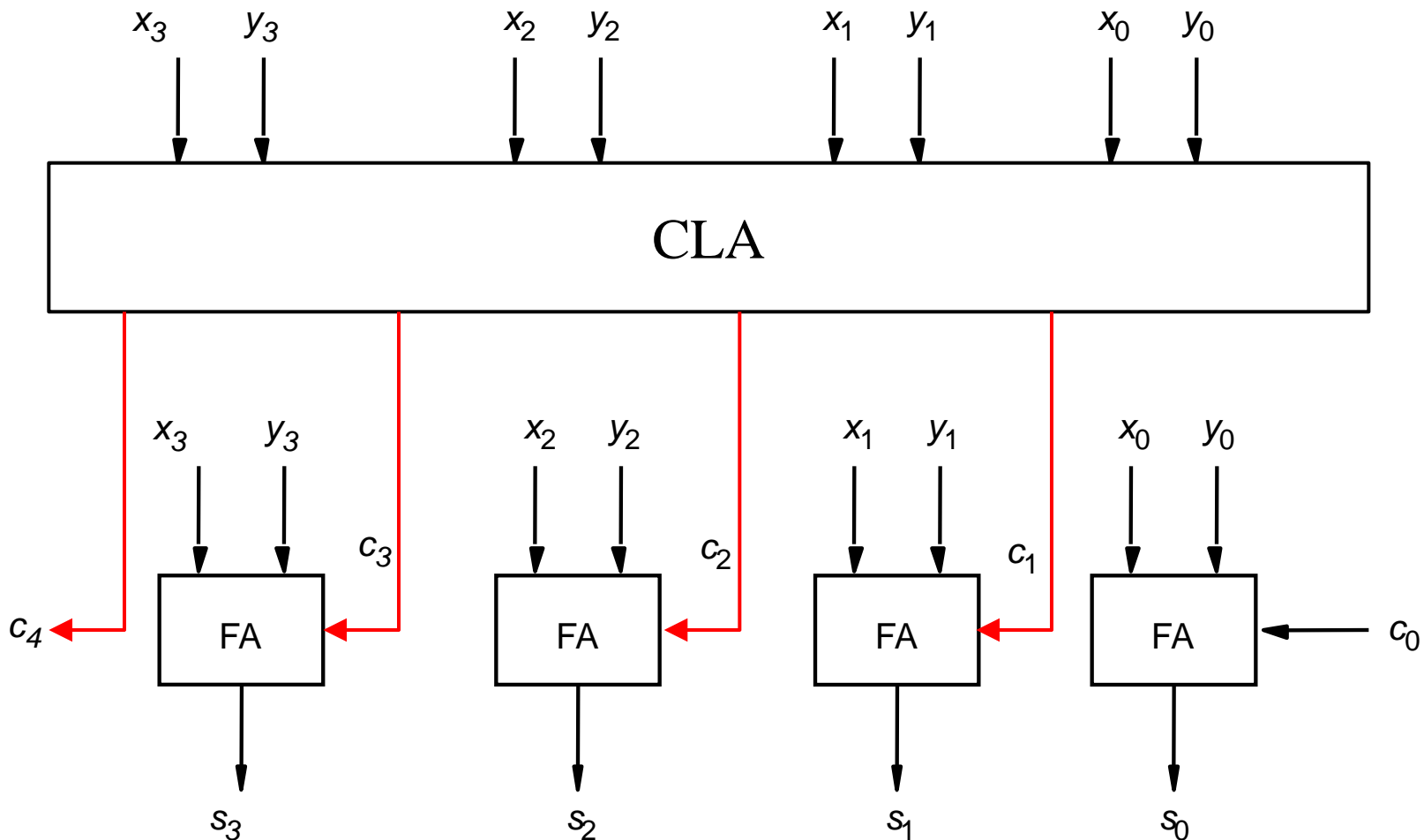
$$t_{\text{ripple}} = Nt_{FA}$$

Onde t_{FA} é o atraso de um full adder



Antecipação de Carry: Carry Look Ahead (CLA)

- Aplicado para módulo de 4 bits



CLA: Generate e Propagate

- Para gerar carries com atraso menor e fixo
- Observar para o bit i
 - Carry é gerado sempre independente das entradas e dos carries de nível anterior:
 - $g_i = x_i y_i$
 - Carry é propagado sempre independente das entradas e dos carries de nível anterior:
 - $p_i = x_i + y_i$
 - observar que um carry de entrada é morto/killed se:
 - $\sim x_i \cdot \sim y_i$
 - Que é exatamente $\sim p_i$

CLA: Como gerar os carries a partir de g e p

$$C_1 = g_0 + p_0 C_0$$

$$C_2 = g_1 + p_1 C_1 \quad C_2 = g_1 + p_1 g_0 + p_1 p_0 C_0$$

$$C_3 = g_2 + p_2 C_2 \quad C_3 =$$

$$C_4 = g_3 + p_3 C_3 \quad C_4 =$$

- Atraso:
 - entradas $\Rightarrow g_i p_i$ (1G)
 - $g_i p_i \Rightarrow$ carry (2G) : 1 AND seguido de 1 OR
 - carry \Rightarrow saídas (2G)
- Total: 5G, independente de n



Codificação em BCD

“No mundo há 10 tipos de pessoas: as que sabem contar em binário e as que não sabem”



BCD

Decimal digit	BCD code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Table 5.3 Binary-coded decimal digits

Adição Usando BCD

$$\begin{array}{r}
 X \quad \quad 0111 \quad \quad 7 \\
 + Y \quad + 0101 \quad + 5 \\
 \hline
 Z \quad \quad 1100 \quad \quad 12 \\
 \quad \quad + 0110 \\
 \hline
 \text{carry} \rightarrow 10010 \\
 \quad \quad \underbrace{\quad\quad\quad} \\
 \quad \quad S = 2
 \end{array}$$

$$\begin{array}{r}
 X \quad \quad 1000 \quad \quad 8 \\
 + Y \quad + 1001 \quad + 9 \\
 \hline
 Z \quad \quad 10001 \quad \quad 17 \\
 \quad \quad + 0110 \\
 \hline
 \text{carry} \rightarrow 10111 \\
 \quad \quad \underbrace{\quad\quad\quad} \\
 \quad \quad S = 7
 \end{array}$$

Passou de 10? Remove 10:

$$\begin{aligned}
 S - 10 &= S - 9 - 1 \\
 &= S + K_2(9_{10}) - 1 \\
 &= S + K_1(9_{10}) + 1 - 1 \\
 &= S + \text{not}(1001_2) \\
 &= S + 0110_2 \\
 &= S + 6_{10}
 \end{aligned}$$

Raciocínio Alternativo

Passou de 10?

Remove 10 (carry=1)

$$\begin{aligned}
 S - 10 &= S - (16 - 6) \\
 &= S + 6 - 16 \\
 &= (S + 6) - 16
 \end{aligned}$$

soma

carry

Somador em BCD

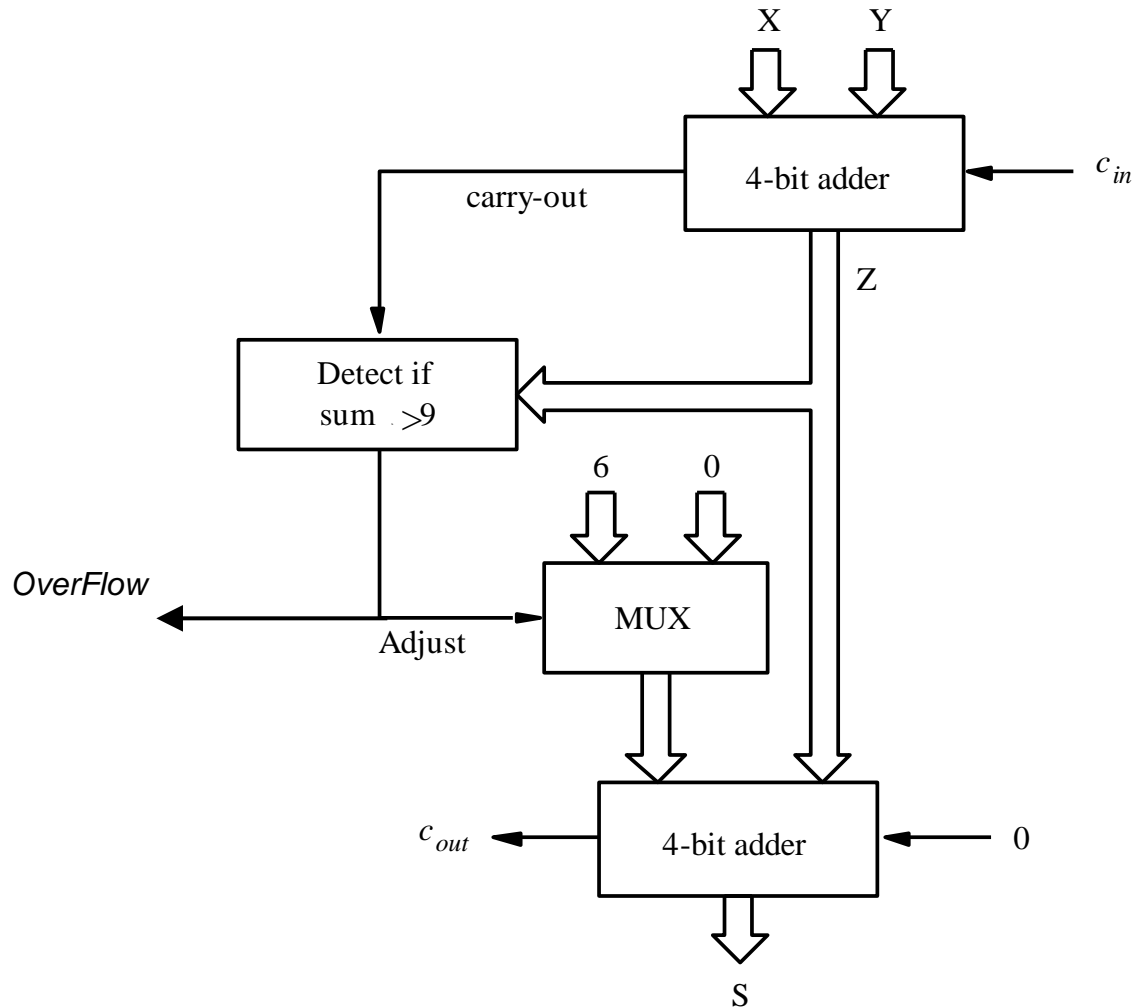


Figure 5.37 Block diagram for a one-digit BCD adder

Somador de um Dígito BCD

$$\begin{array}{r}
 z_3 \ z_2 \ z_1 \ z_0 \\
 + 0 \ 1 \ 1 \ 0 \\
 \hline
 \end{array}$$

$C_{out} = 1 ?$

$$C_{out} = d_{out} + z_2 z_3 + z_1 z_3$$

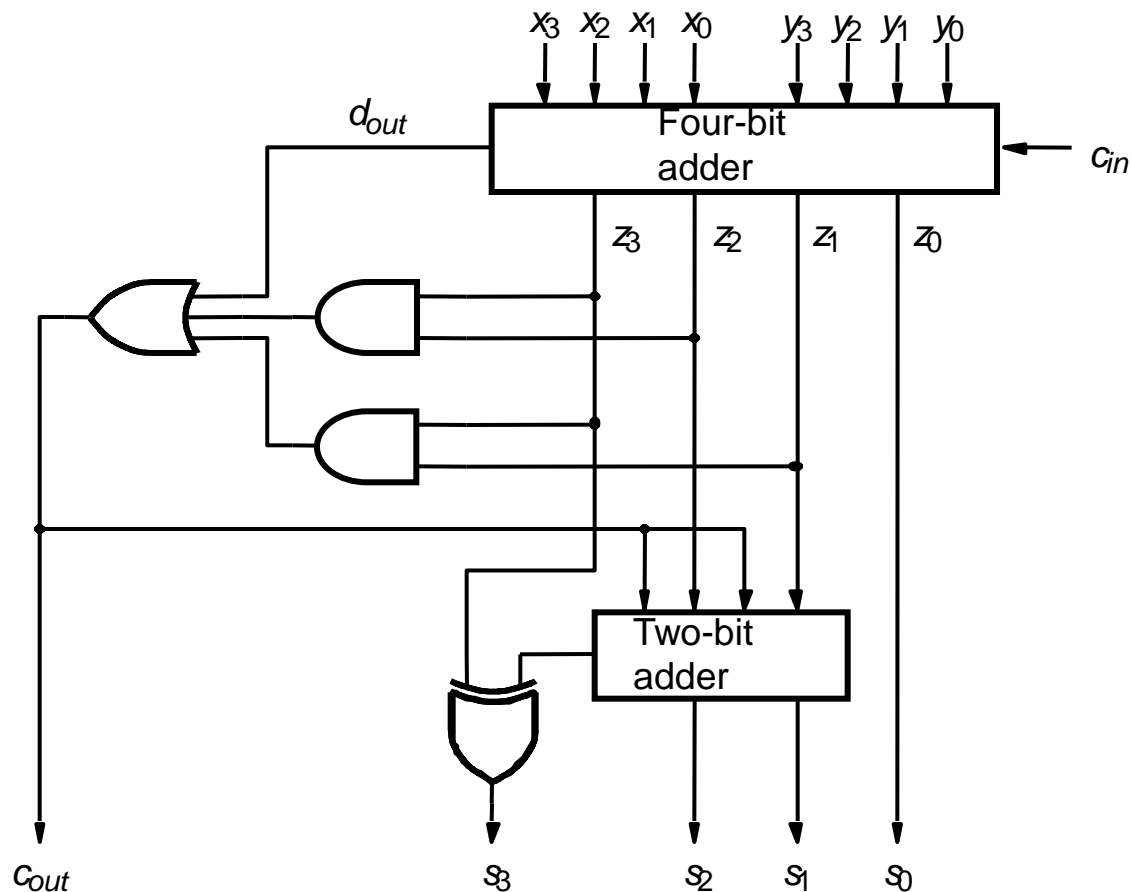


Figure 5.40 Circuit for a one-digit BCD adder

Somador BCD



IC-UNICAMP

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

ENTITY BCD IS
    PORT ( X, Y: IN STD_LOGIC_VECTOR(3 DOWNT0 0) ;
          S: OUT STD_LOGIC_VECTOR(4 DOWNT0 0) ) ;
END BCD ;

ARCHITECTURE Behavior OF BCD IS
    SIGNAL Z : STD_LOGIC_VECTOR(4 DOWNT0 0) ;
    SIGNAL Adjust : STD_LOGIC ;
BEGIN
    Z <= ('0' & X) + Y ;
    Adjust <= '1' WHEN Z > 9 ELSE '0' ;
    S <= Z WHEN (Adjust = '0') ELSE Z + 6 ;
END Behavior ;
```

Figure 5.38 VHDL code for a one-digit BCD adder