

CIRCUITOS ARITMÉTICOS E CÓDIGOS ESPECIAIS

OBJETIVOS

- Projetar circuitos somadores binários;
- Projetar circuitos subtratores binários;
- Projetar circuitos multiplicadores binários;
- Projetar circuitos somadores e subtratores em BCD;
- Trabalhar com Unidades Lógicas e Aritméticas – ALU;
- Trabalhar com o Código de GRAY;
- Trabalhar com Códigos Detectores e Corretores de Erro.

RESUMO

Introdução

Adição

Subtração

Unidade Lógica e
Aritmética

Adição em BCD

Código de Gray

Código Detector de Erro

Código Corretor de Erro

ADIÇÃO

Entradas			Saídas		
"Bit" 1	+	"Bit" 2	→	Resultado	Transporta
0		0		0	0
0		1		1	0
1		0		1	0
1		1		0	1

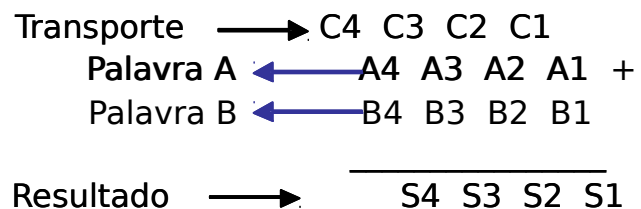
Transporte 1 1 0 0
 1 1 0 0 + 12 +
 0 1 1 0 6

 1 0 0 1 0 18

1 0 1 1 1 0
 1 0 1 1 1 1 + 47 +
 1 0 0 0 1 0 34

 1 0 1 0 0 0 1 81

Projeto do Somador para quatro “bits”



Somador Incompleto (“Half Adder”)

Entradas		Saídas	
		Soma	“Carry ”
A1	B1	S1	C1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{aligned}
 S1 &= \bar{A}1\bar{B}1 + A1\bar{B}1 \\
 &= A1 + B1 \\
 C1 &= A1 B1
 \end{aligned}$$

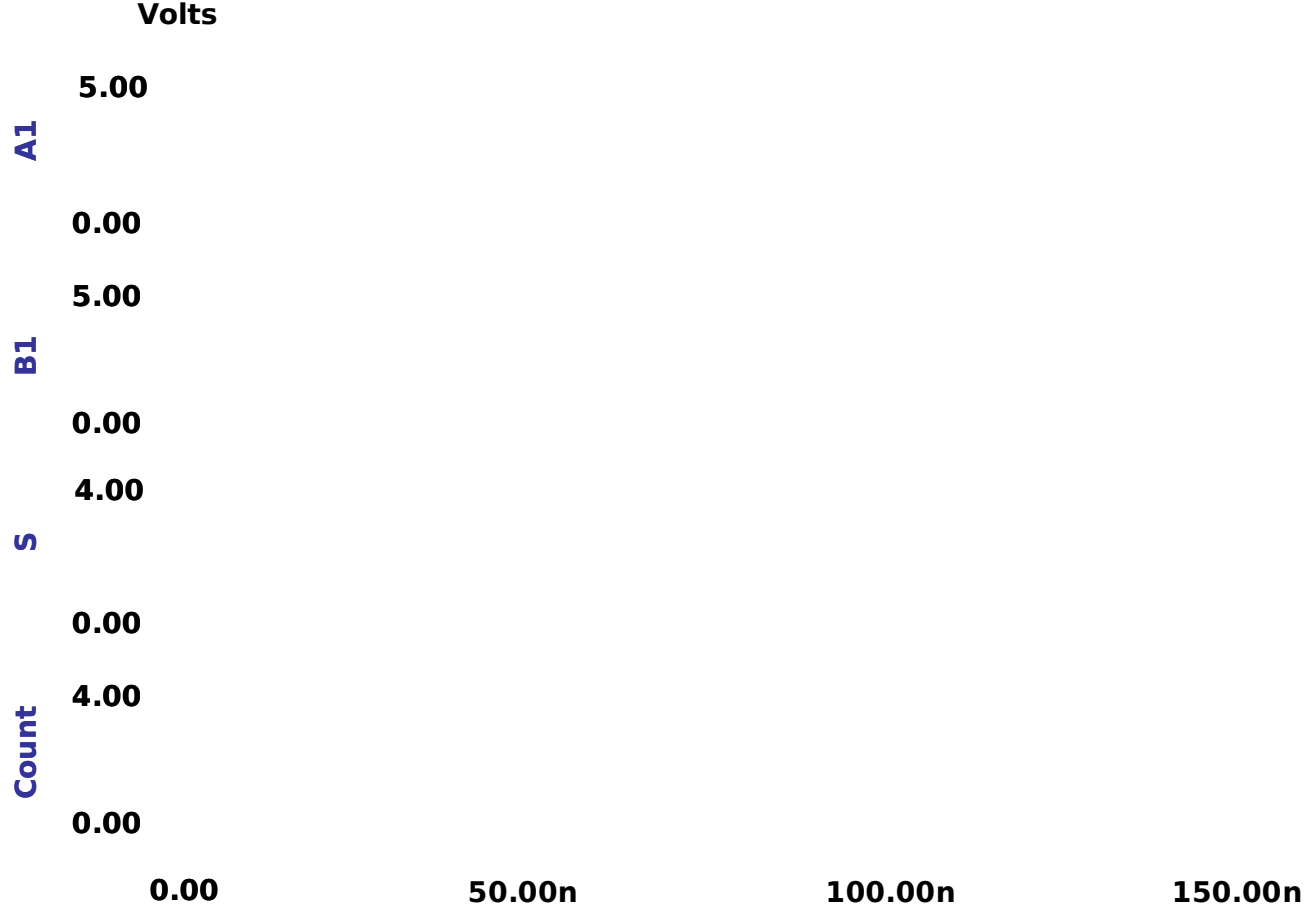
A1

B1

S1

C1

Entradas		Saídas	
		Soma	“Carry ”
A1	B1	S1	C1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Somador Completo (“Full-Adder”)

Entradas			Saídas	
Ci-1	Ai	Bi	Si	Ci
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Si = (Ci-1) \cdot (Bi)' \cdot Ai + (Ci-1) \cdot Bi \cdot (Ai)' + (Ci-1) \cdot Bi \cdot Ai$$

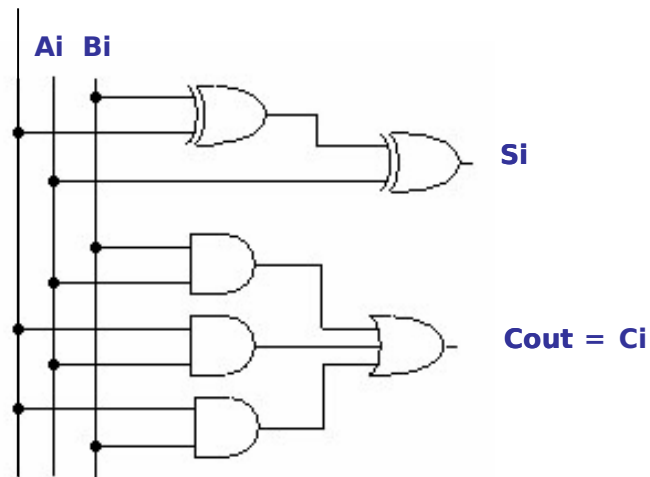
$$= (Ci-1) \cdot (Ai \oplus Bi) + (Ci-1) \cdot (Ai \oplus Bi)$$

$$= (Ci-1) \oplus Ai \oplus Bi$$

$$Ci = (Ci-1) \cdot Ai + (Ci-1) \cdot Bi + Ai \cdot Bi$$

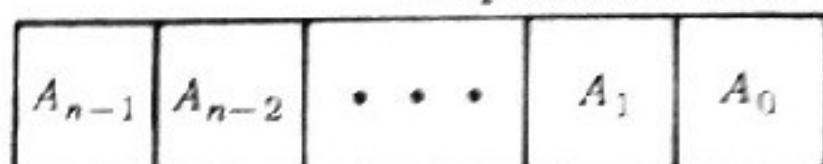
$$= Ai \cdot Bi + (Ci-1) \cdot (Ai \oplus Bi)$$

Ci-1



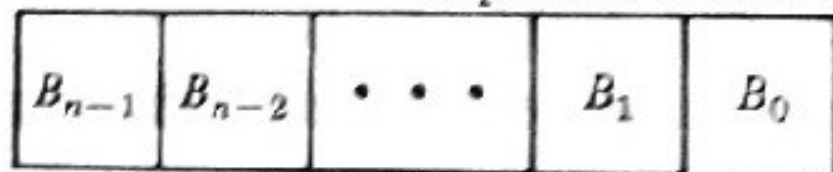
Relógio

Registrador de deslocamento
de parcela com n bits

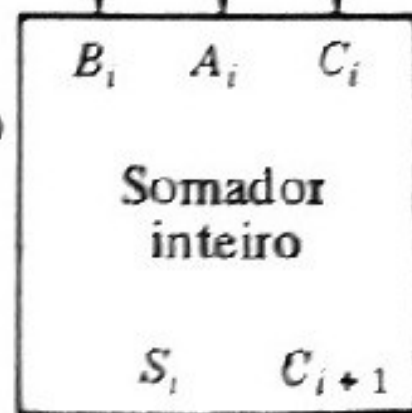
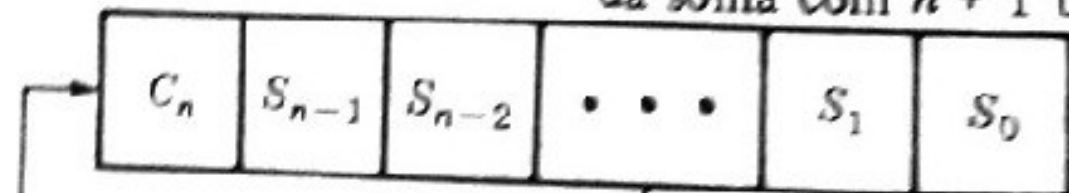


Direção de
deslocamento
→

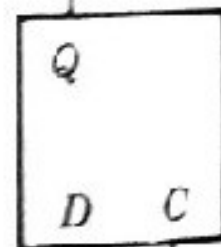
Registrador de deslocamento
de parcela com n bits



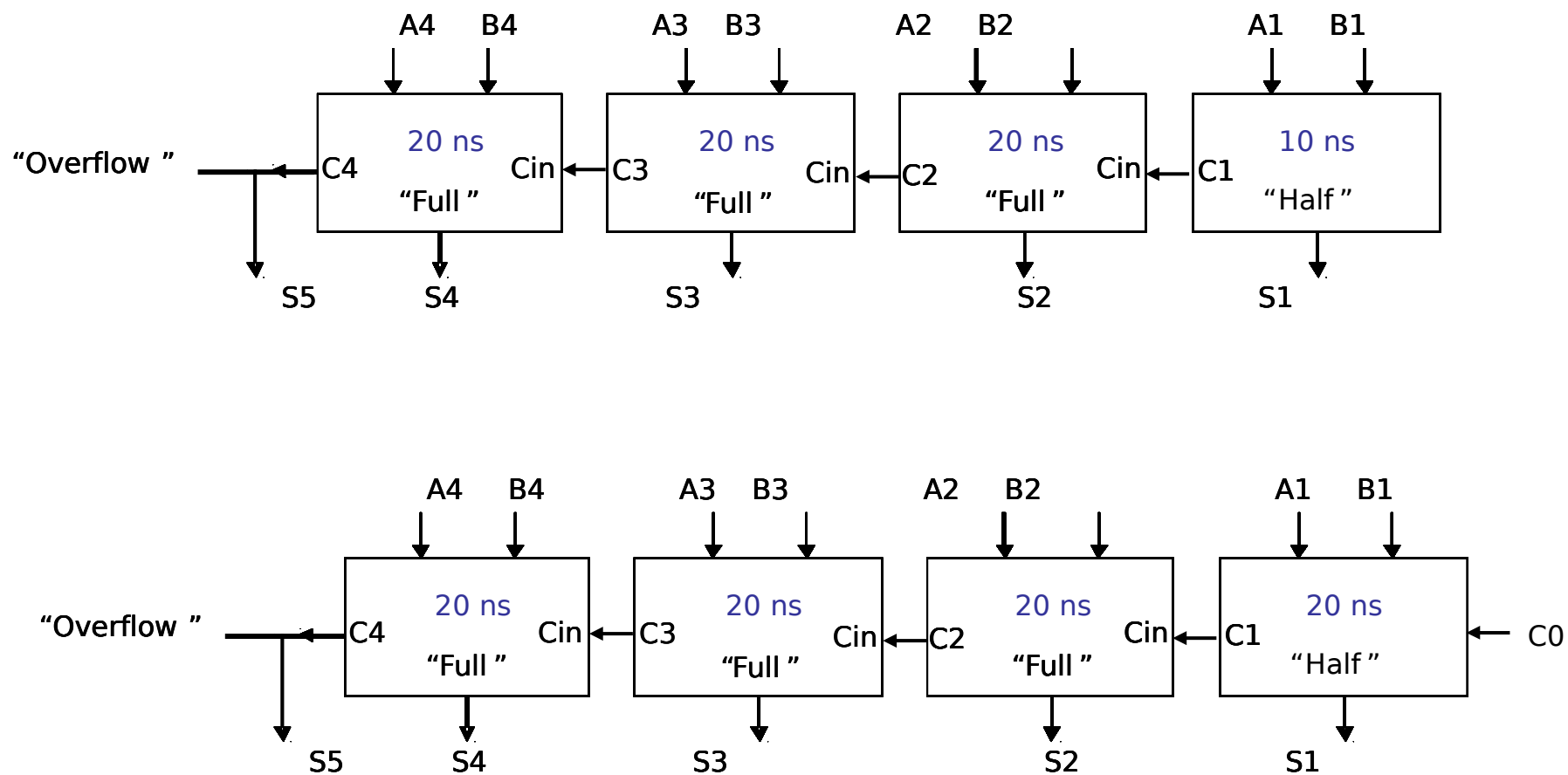
Registrador de deslocamento
da soma com $n + 1$ bits



Flip-flop
 D



Somador Paralelo

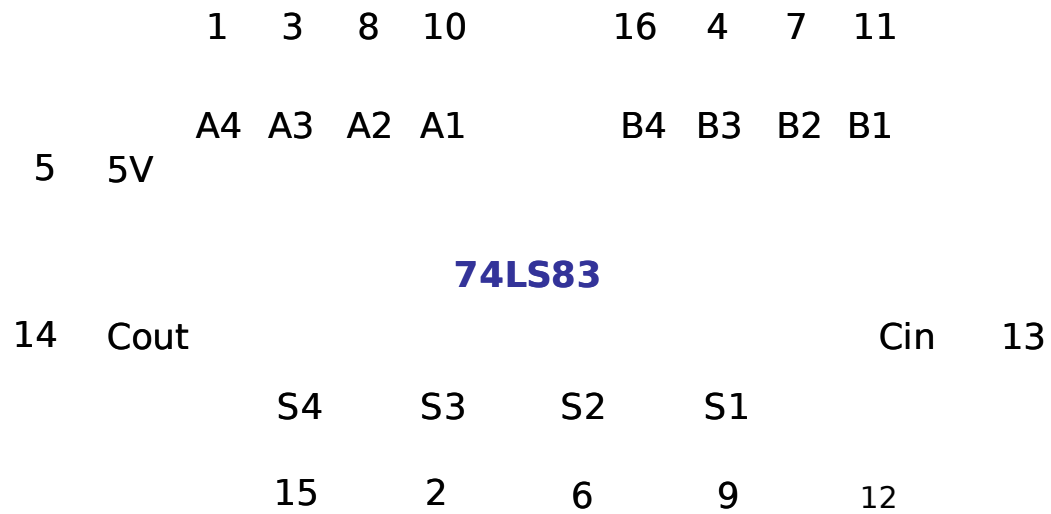


$$C1 = A1B1 + A1C0 + B1C0$$

$$\begin{aligned} C2 &= A2B2 + A2C1 + B2C1 \\ &= A2B2 + A2(A1B1 + A1C0 + B1C0) + B2(A1B1 + A1C0 + B1C0) \end{aligned}$$

$$\begin{aligned} C3 &= A3B3 + A3C2 + B3C2 \\ &= A3B3 + A3[A2B2 + A2(A1B1 + A1C0 + B1C0) + B2(A1B1 + A1C0 + B1C0)] + \\ &\quad B3[A2B2 + A2(A1B1 + A1C0 + B1C0) + B2(A1B1 + A1C0 + B1C0)] \end{aligned}$$

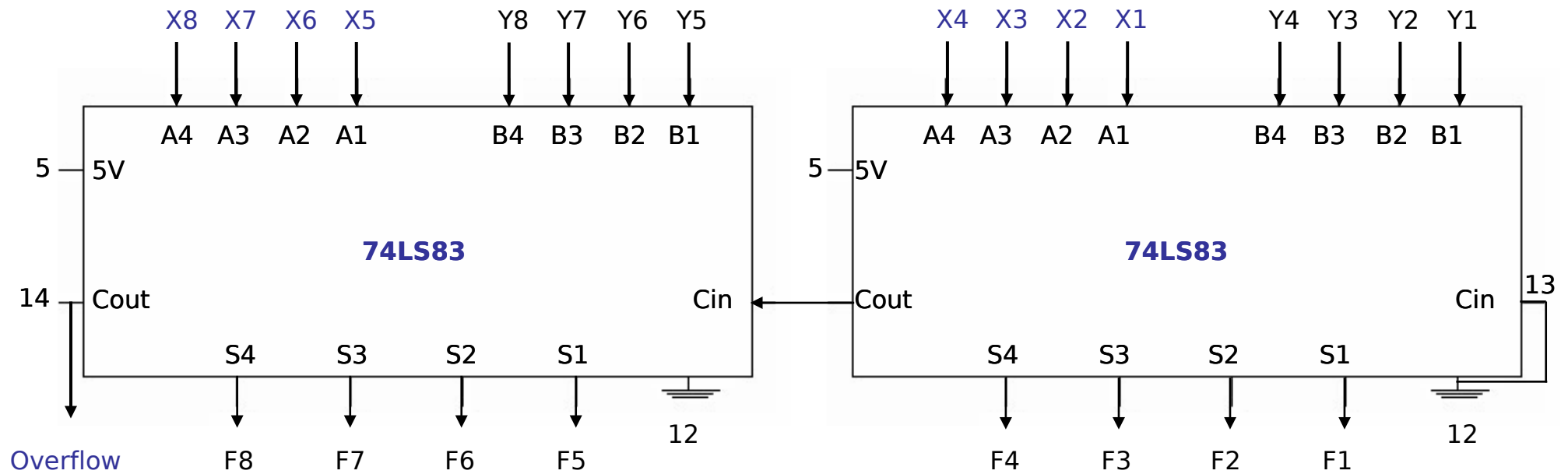
Circuito Integrado – 74LS83



Exemplo

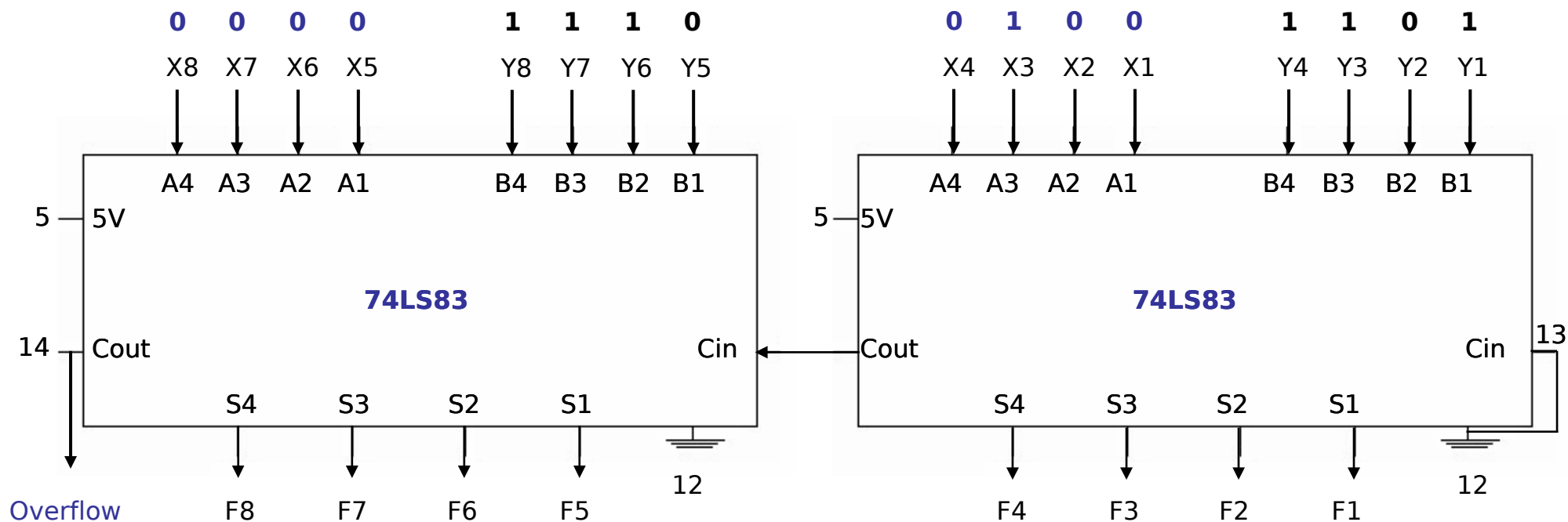
Projeto de um somador para números binários positivos com oito “bits”.

.



Projetar um circuito que realize a adição dos decimais (+4) e (-19) representados em complemento de 2 (assunto tratado no capítulo 1).

Sem adição complementar



SUBTRAÇÃO

Entradas

Saídas

"Bit" 1	-	"Bit" 2	→	Resultado	Empresta
0		0		0	0
0		1		1	1
1		0		1	0
1		1		0	0

Exemplo

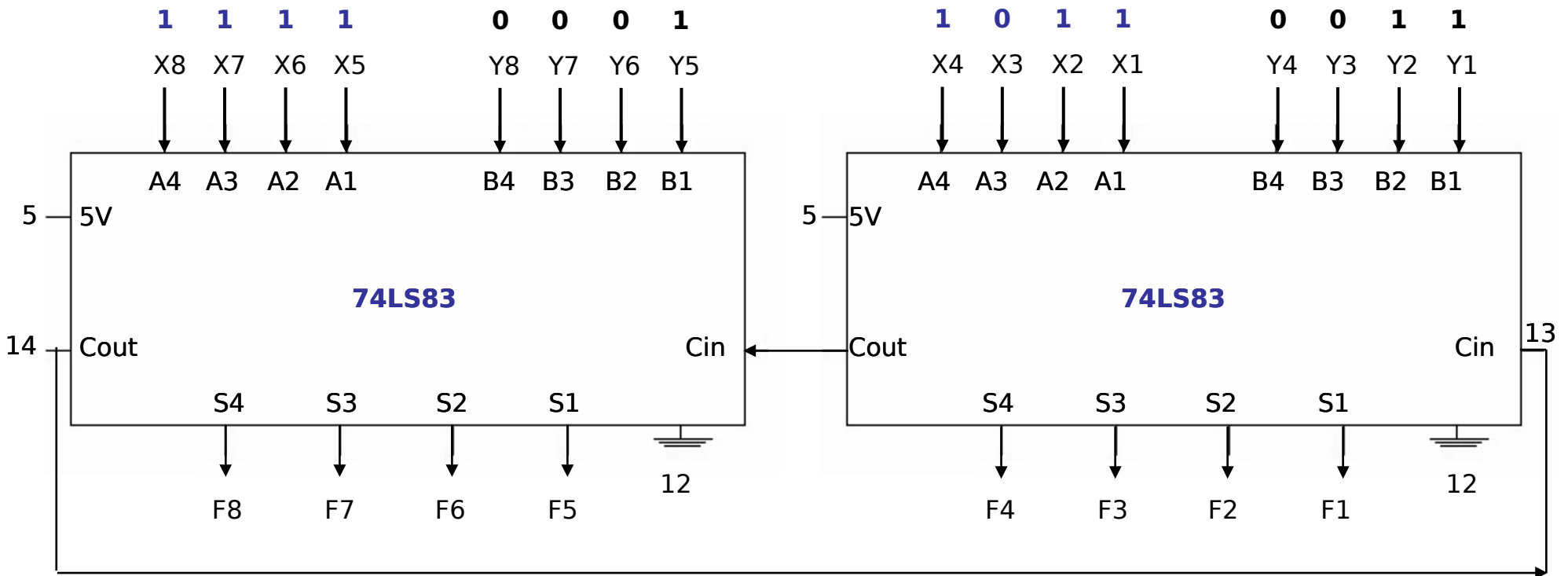
Subtração dos números binários 1100 e 0110.

Empréstimo	-1 -1	
	1 1 0 0 -	12 -
	0 1 1 0	6
Diferença	-----	-----
	0 1 1 0	6

Exemplo

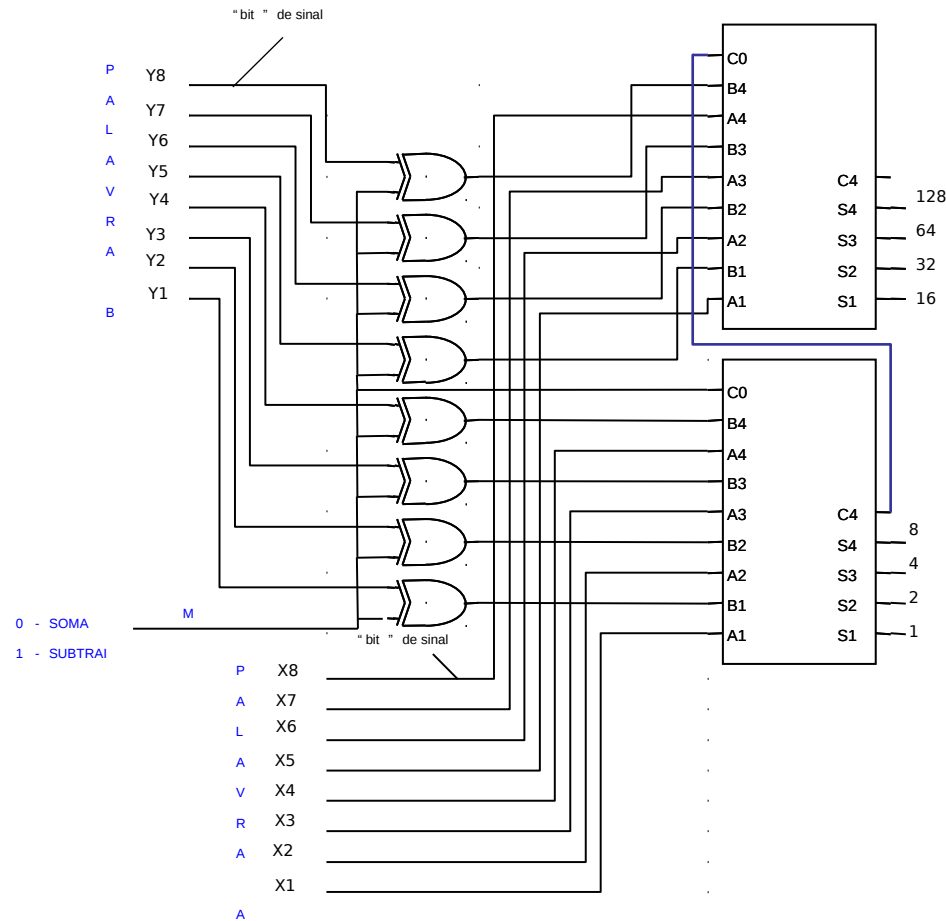
Projetar um circuito que realize a subtração, em complemento de 1 (assunto tratado no capítulo 1), dos decimais (-4) e (-19).

$$\begin{array}{rcl}
 \begin{array}{r}
 -4 \\
 -19 \\
 \hline
 +15
 \end{array} & \begin{array}{l} \rightarrow \\ \rightarrow \\ \rightarrow \end{array} & \begin{array}{r}
 1.1111011 \\
 1.1101100 \\
 \hline
 0.0001111
 \end{array}
 \end{array}
 \begin{array}{l}
 \rightarrow \text{Minuendo} \\
 \rightarrow \text{Subtraendo} \\
 \rightarrow
 \end{array}
 \begin{array}{r}
 -4 + \\
 +19 \\
 \hline
 +15
 \end{array}
 \begin{array}{r}
 \begin{array}{r}
 1111 \quad 11 \\
 1.1111011 + \\
 0.0010011 \\
 \hline
 0.0001110 + \\
 \quad 1 \\
 \hline
 0.0001111
 \end{array} \\
 \text{Adição Suplementar} \\
 \text{Resultado}
 \end{array}
 \end{array}$$

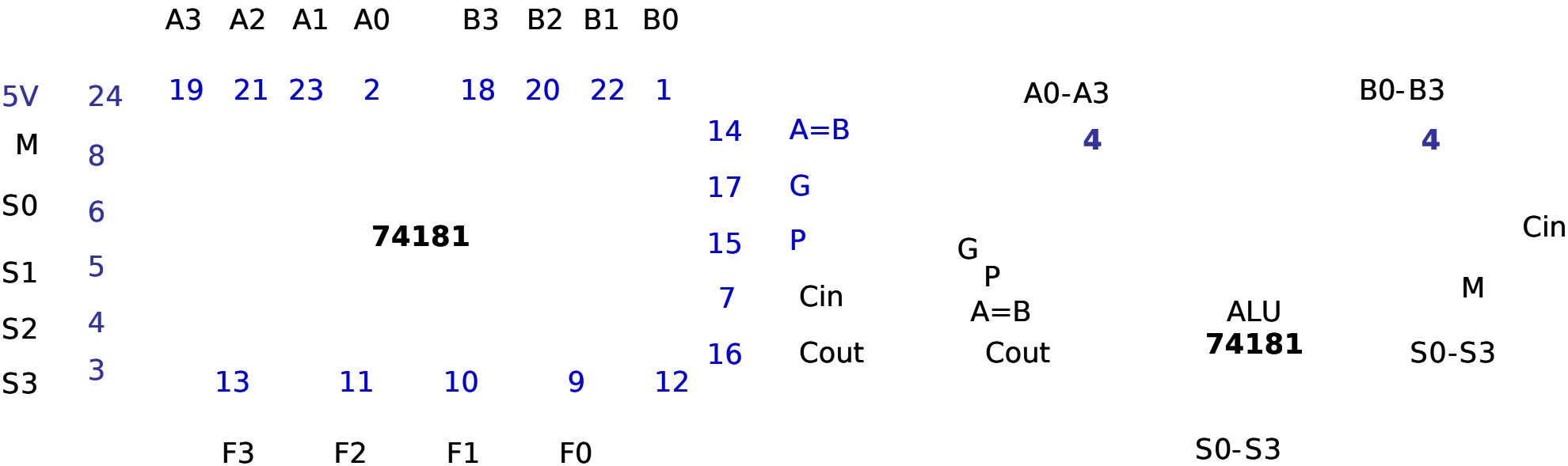


Exemplo

Projetar um circuito que apresente o resultado da adição ou a subtração de duas palavras, X e Y, cada uma com oito “bits” (incluindo o “bit” de sinal) e representadas na notação complemento de 2.



UNIDADE LÓGICA E ARITMÉTICA – ALU



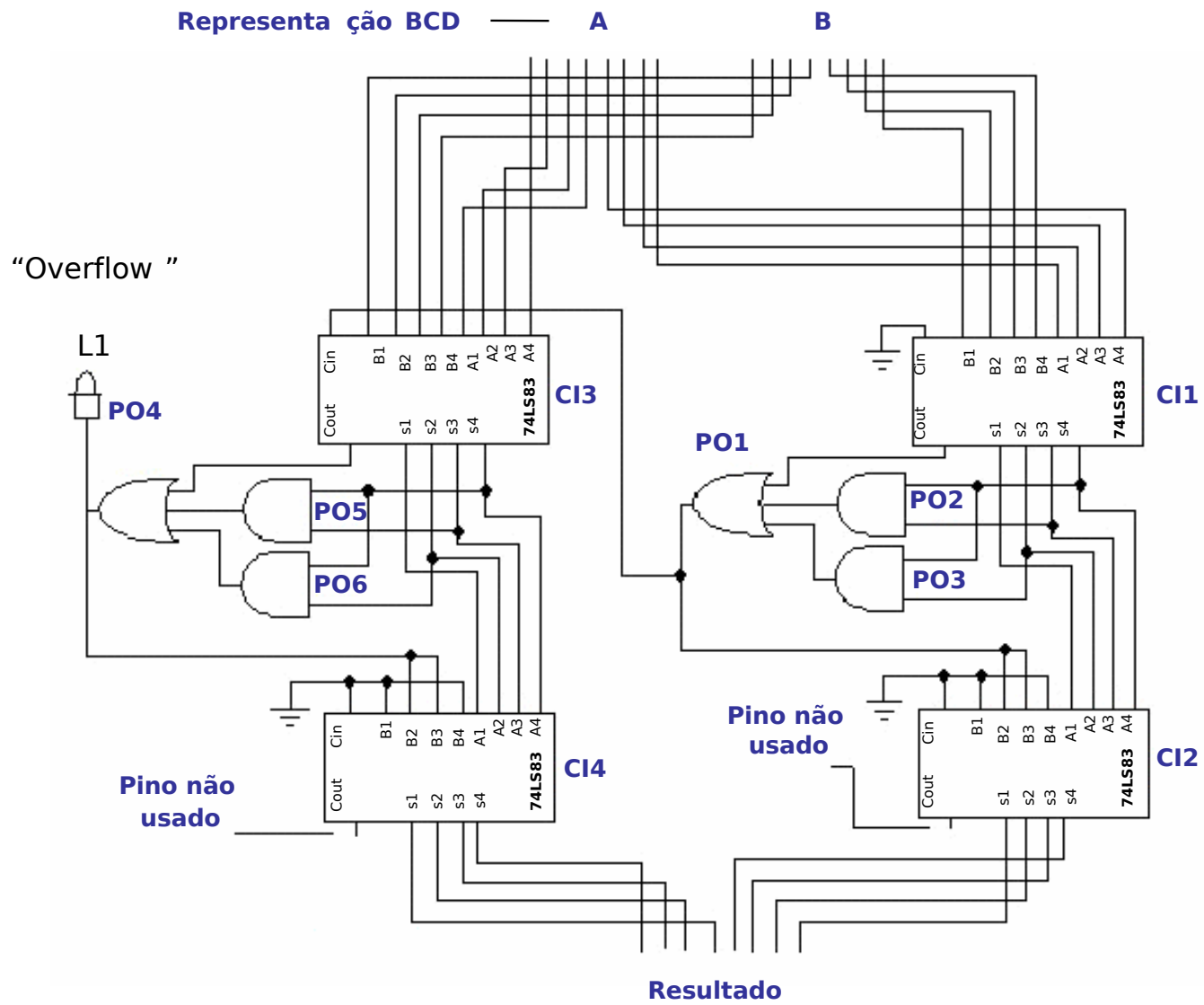
0	0	0	0	$F = A$	$F = A$	$F = A \text{ mais } 1$
0	0	0	1	$F = A + B$	$F = A + B$	$F = (A+B) \text{ mais } 1$
0	0	1	0	$F = A.B$	$F = A + B$	$F = (A+B) \text{ mais } 1$
0	0	1	1	$F = 0$	$F = \text{Menos } 1 \text{ (comp de } 2)$	$F = 0$
0	1	0	0	$F = A.B$	$F = A \text{ mais } A.B$	$F = A \text{ mais } A.B \text{ mais } 1$
0	1	0	1	$F = B$	$F = (A+B) \text{ mais } A.B$	$F = (A+B) \text{ mais } A.B \text{ mais } 1$
0	1	1	0	$F = A + B$	$F = A \text{ menos } B \text{ menos } 1$	$F = A \text{ menos } B$
0	1	1	1	$F = A.B$	$F = A.B \text{ menos } 1$	$F = A.B$
1	0	0	0	$F = A + B$	$F = A \text{ mais } A.B$	$F = A \text{ mais } A.B \text{ mais } 1$
1	0	0	1	$F = A + B$	$F = A \text{ mais } B$	$F = A \text{ mais } B \text{ mais } 1$
1	0	1	0	$F = B$	$F = (A+B) \text{ mais } A.B$	$F = (A+B) \text{ mais } A.B \text{ mais } 1$
1	0	1	1	$F = A.B$	$F = A.B \text{ menos } 1$	$F = A.B$
1	1	0	0	$F = 1$	$F = A \text{ mais } A.B$	$F = A \text{ mais } A \text{ mais } 1$
1	1	0	1	$F = A + B$	$F = (A+B) \text{ mais } A.B$	$F = (A+B) \text{ mais } A \text{ mais } 1$
1	1	1	0	$F = A + B$	$F = (A+B) \text{ mais } A$	$F = (A+B) \text{ mais } A \text{ mais } 1$
1	1	1	1	$F = A$	$F = A \text{ menos } 1$	$F = A$

ADIÇÃO EM BCD

	Transporte	1	1110	100	
	254 =	0010	0101	0100	+
	175 =	0001	0111	0101	
			1		
Soma Complementar		0100	1100	1001	+
			0110		
	429 =	0100	0010	1001	

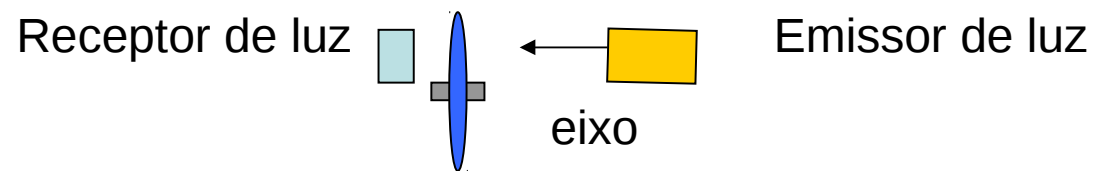
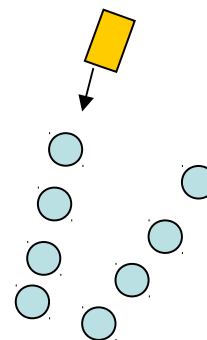
Transporte

Decimal 6



CÓDIGO DE GRAY

	A	B	X	Y	Z	Q	W	S	R	P	M	H	J
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	1	0	0	0	1
2	1	1	0	1	1	0	0	1	1	0	0	1	0
3	1	0	0	1	0	0	0	1	0	0	0	1	1
4			1	1	0	0	1	1	0	0	1	0	0
5			1	1	1	0	1	1	1	0	1	0	1
6			1	0	1	0	1	0	1	0	1	1	0
7			1	0	0	0	1	0	0	0	1	1	1
8						1	1	0	0	1	0	0	0
9						1	1	0	1	1	0	0	1
10						1	1	1	1	1	0	1	0
11						1	1	1	0	1	0	1	1
12						1	0	1	0	1	1	0	0
13						1	0	1	1	1	1	0	1
14						1	0	0	1	1	1	1	0
15						1	0	0	0	1	1	1	1



Raio de luz

Fotosensor

A = 1

B = 0

C = 1

D = 0

Eixo

Roda de Gray

CÓDIGO DETECTOR DE ERRO

**Código Binário
Original**

**“Bit” de
paridade**

A

A

B

Gerador de
Paridade

X

B

Gerador de
Paridade

**Código Binário
Com “bit” de
paridade**

C

C

D

D

**Código Binário
Original**

ABCD

0000	1
0001	2
0010	1
0011	3
0100	1
0101	2
0110	1
0111	4
1000	1
1001	2
1010	1
1011	3
1100	1
1101	2
1110	1
1111	4

ABCD**X**

0000	0	2
0001	1	2
0010	1	2
0011	0	4
0100	1	2
0101	0	2
0110	0	2
0111	1	3
1000	1	2
1001	0	2
1010	0	2
1011	1	4
1100	0	2
1101	1	2
1110	1	2
1111	0	4

$$X = A + B + C + D$$

A

B

C

D

X

CÓDIGO CORRETOR DE ERRO

