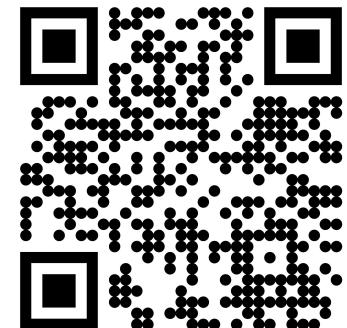


# Controladores Lógicos Programáveis

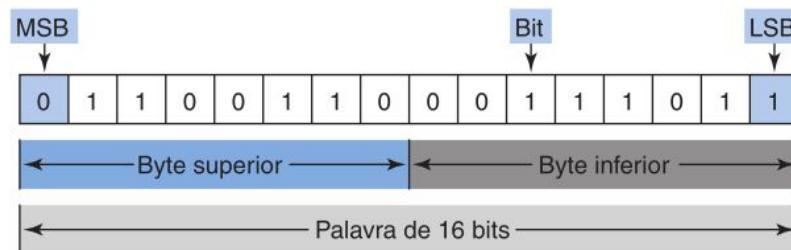
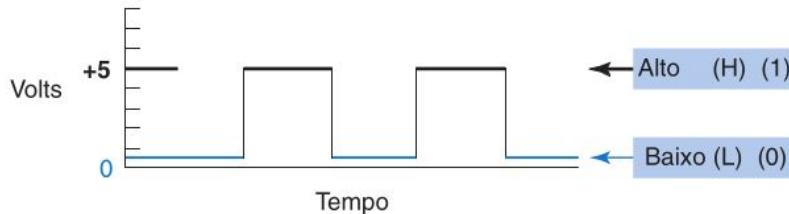
## Números e Lógicas

Profº José W. R. Pereira  
[jose.pereira@ifsp.edu.br](mailto:jose.pereira@ifsp.edu.br)  
[josewrpereira.github.io/docs](http://josewrpereira.github.io/docs)



# Sistemas de Numeração

Os CLPs operam internamente através de **números binários** para processar e armazenar informações, por isso há um importância em conhecer como são **representados** os dados em algumas de suas diferentes formas.



# Sistemas de Numeração

Bits

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0000																
0001																
0002																
0003																
0004	0	1	1	0	0	1	1	0	0	0	1	1	1	0	1	1
0005																

# Sistemas de Numeração

Decimal	Octal	Hexadecimal	Binário
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	10	8	1000
9	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111
16	20	10	10000
17	21	11	10001
18	22	12	10010
19	23	13	10011
20	24	14	10100

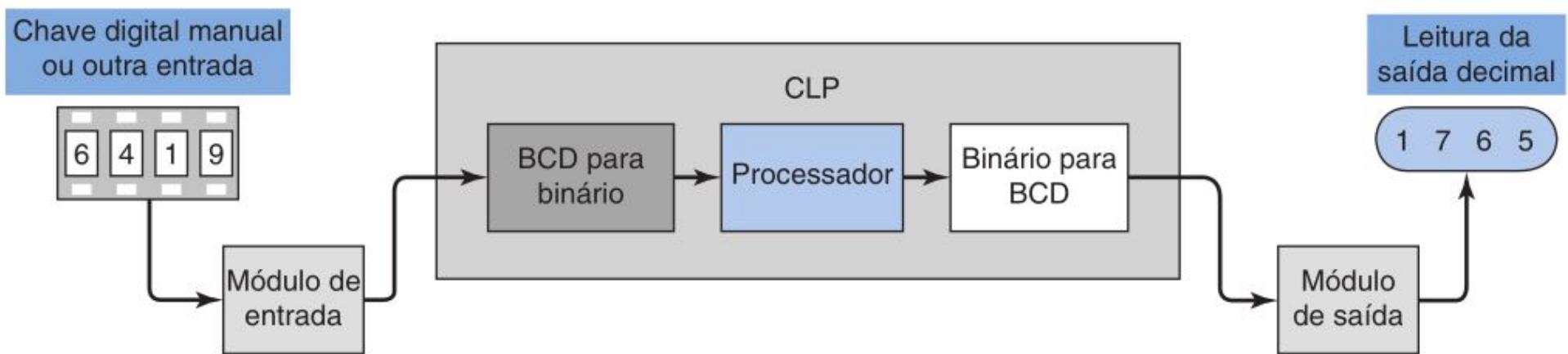
# Números Negativos

Magnitude Sinal	Valor decimal
0111	+7
0110	+6
0101	+5
0100	+4
0011	+3
0010	+2
0001	+1
0000	0
1001	-1
1010	-2
1011	-3
1100	-4
1101	-5
1110	-6
1111	-7

O mesmo  
que números  
binários

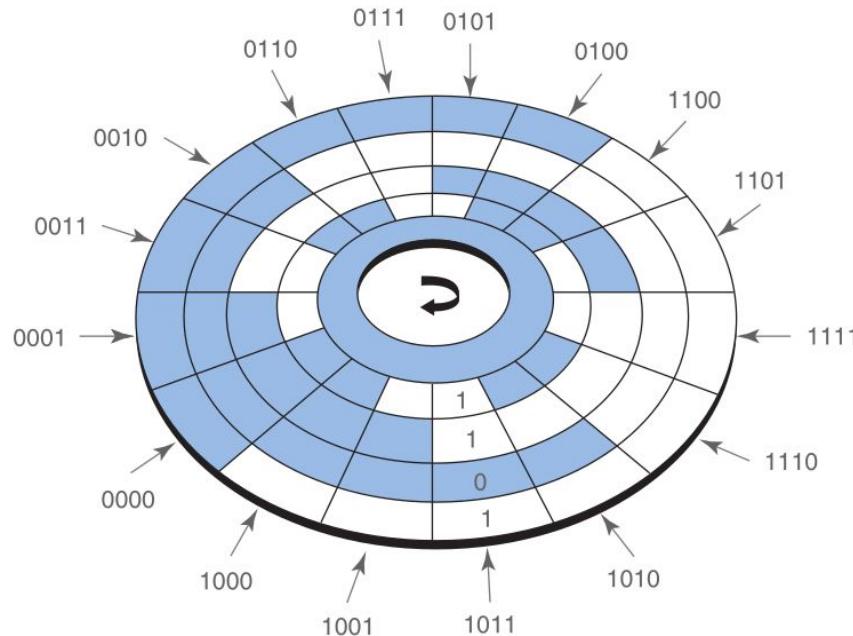
# Códigos Digitais

**BCD - Binary Coded Decimal (Decimal Codificado em Binário):** Representa cada dígito decimal individualmente através de um **grupo de 4 bits**. É o formato utilizado por dispositivos externos como **chaves de tambor (thumbwheels)** e displays de LED.



# Códigos Digitais

**Código Gray:** Caracteriza-se por mudar apenas **um bit** na transição entre números consecutivos, sendo ideal para **codificadores (encoders) ópticos**, pois minimiza erros de leitura em movimentos rápidos.



# Códigos Digitais

**Código ASCII:** Código alfanuméricico que utiliza 7 ou 8 bits para representar letras, números e caracteres especiais, permitindo a interface do CLP com teclados e impressoras.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

# Integridade de Dados

**Bit de Paridade:** É um dígito binário adicional anexado a uma palavra para **detectar erros** durante a transmissão de dados entre CLPs ou computadores. Pode ser configurado como paridade **par** ou **ímpar**, dependendo da lógica do sistema.

Caractere	Bit de paridade par	Bit de paridade ímpar
0000	0	1
0001	1	0
0010	1	0
0011	0	1
0100	1	0
0101	0	1
0110	0	1
0111	1	0
1000	1	0
1001	0	1

# Operações Lógicas

O CLP opera com base no **princípio binário**, onde estados como ligado/desligado ou verdadeiro/falso são representados pelos dígitos **1** e **0**. Uma **porta lógica** é um circuito que possui várias entradas e uma única saída, ativada por combinações específicas dessas entradas. A lógica permite que o controlador tome decisões baseadas em um ou mais fatores antes de executar uma ação.

# Lógica E (AND)

**Função AND:** A saída é 1 apenas se **todas as entradas forem 1**.

Em termos elétricos, é análoga a dispositivos conectados em **série**.

Círcuito montado com fios

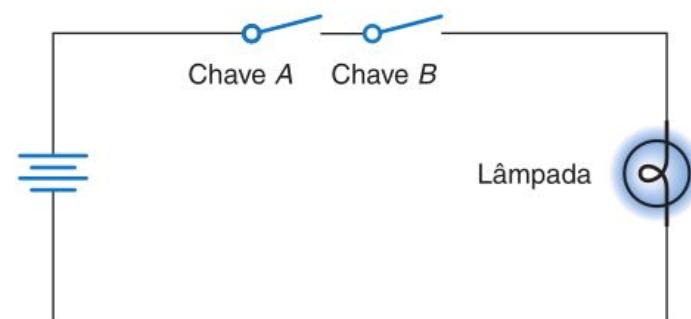
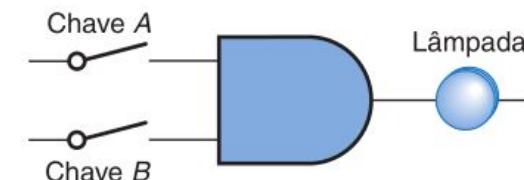


Tabela-verdade

Chave A	Chave B	Lâmpada
Aberta (0)	Aberta (0)	Desligada (0)
Aberta (0)	Fechada (1)	Desligada (0)
Fechada (1)	Aberta (0)	Desligada (0)
Fechada (1)	Fechada (1)	Ligada (1)

Representação lógica



# Lógica OU (OR)

- Função OR:** A saída é 1 se **uma ou mais entradas forem 1**. É similar a dispositivos conectados em **paralelo**.

Círcuito montado com fios

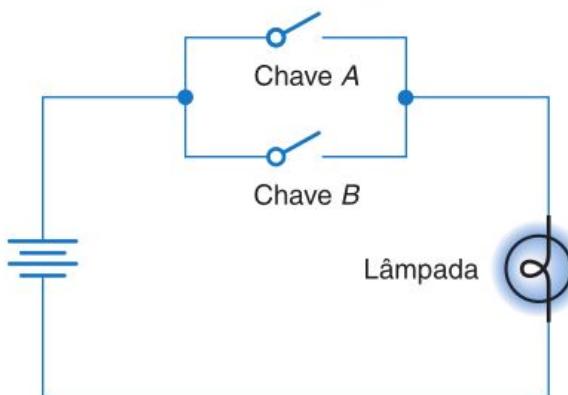
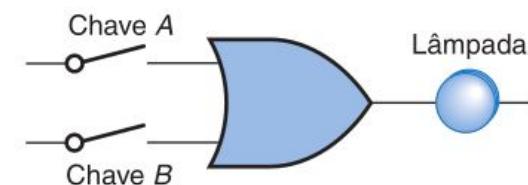


Tabela-verdade

Chave A	Chave B	Lâmpada
Aberta (0)	Aberta (0)	Desligada (0)
Aberta (0)	Fechada (1)	Ligada (1)
Fechada (1)	Aberta (0)	Ligada (1)
Fechada (1)	Fechada (1)	Ligada (1)

Representação lógica



# Lógica OU (OR)

- Função NOT:** Possui apenas uma entrada e sua saída é sempre o **inverso** desta, sendo por isso chamada de inversor.

## Círcuito montado com fios

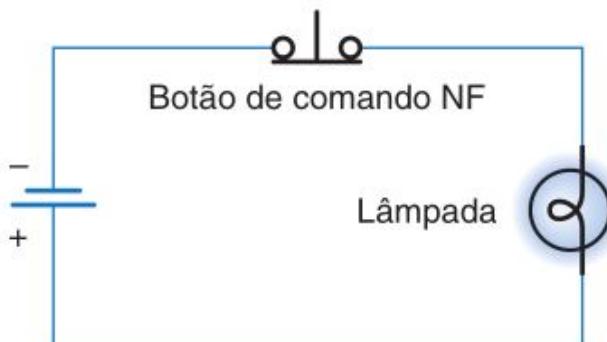
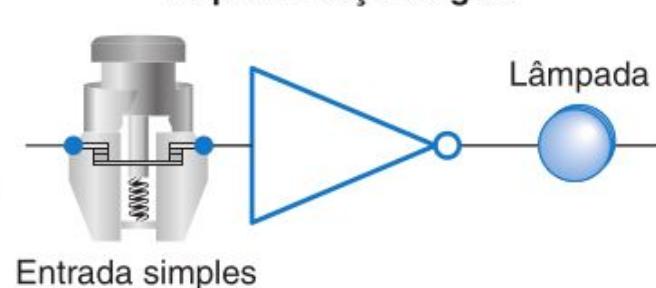


Tabela-verdade

Botão de comando	Lâmpada
Não pressionado (0)	Ligada (1)
Pressionado (1)	Desligada (0)

Representação lógica



# Álgebra de Boole

A **álgebra booleana** é o estudo matemático da lógica e fornece um método simples para escrever combinações complexas através de **equações**. É possível desenvolver circuitos lógicos a partir de expressões booleanas e vice-versa.

- Identidade:
  - $A + 0 = A$
  - $A \cdot 1 = A$
- Nulo (Absorvente):
  - $A + 1 = 1$
  - $A \cdot 0 = 0$
- Idempotência:
  - $A + A = A$
  - $A \cdot A = A$
- Complementaridade:
  - $A + \bar{A} = 1$
  - $A \cdot \bar{A} = 0$
- Involução (Dupla Negação):
  - $\bar{\bar{A}} = A$

- Comutativa:
  - $A + B = B + A$
  - $A \cdot B = B \cdot A$
- Associativa:
  - $A + (B + C) = (A + B) + C$
  - $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- Distributiva:
  - $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$
  - $(A \cdot B) + C = (A + C) \cdot (B + C)$
- Absorção:
  - $A + A \cdot B = A$
  - $A \cdot (A + B) = A$

- **Primeira Lei:**
  - $\overline{A \cdot B} = \overline{A} + \overline{B}$
- **Segunda Lei:**
  - $\overline{A + B} = \overline{A} \cdot \overline{B}$

# Referências

1. PETRUZELLA, Frank D. **Controladores Lógicos Programáveis**. Tradução de Romeu Abdo; revisão técnica de Antonio Pertence Júnior. 4. ed. Porto Alegre: AMGH, 2014.
2. GEORGINI, Marcelo. **Automação Aplicada: Descrição e Implementação de Sistemas Sequenciais com PLCs**. 9. ed. São Paulo: Érica, 2007.
3. SILVA FILHO, Bernardo Severo da. **Curso de Controladores Lógicos Programáveis**. Rio de Janeiro: UERJ, Faculdade de Engenharia, Laboratório de Engenharia Elétrica, [s.d.].
4. MITSUBISHI ELECTRIC BRASIL. **CLP: O que é, para que serve e como funciona na indústria**. [s.l.]: YouTube, [s.d.]. 1 vídeo.
5. ALTUS. **O que é CLP e quando utilizá-lo?**. São Leopoldo: Altus, [s.d.]. Disponível em: Blog da Altus.

# Controladores Lógicos Programáveis

Profº José W. R. Pereira  
[jose.pereira@ifsp.edu.br](mailto:jose.pereira@ifsp.edu.br)  
[josewrpereira.github.io/docs](https://josewrpereira.github.io/docs)

