Introdução à Informática

Alexandre Meslin
(meslin@nce.ufrj.br)

Objetivo do Curso

- Apresentar os conceitos básicos de informática, de software e de hardware.
- Introduzir os conceitos relativos à representação da informação e o sistema de numeração em base binária.
- Fornecer uma noção geral das partes constituintes de um computador e de sua funcionalidade.

Programação do Curso

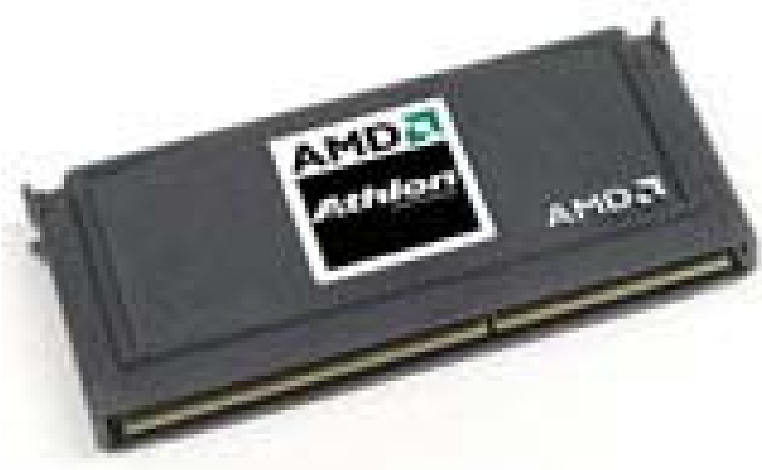
- Introdução ao Computador
- Números binários e hexadecimais
- Representação Interna
- Conceitos básicos do hardware do computador
- Organização lógica e funcional do modelo Von-Neumann
- Estudo dos diversos componentes de um processador
- Unidade de entrada e saída
- Memória

Aula 1

- Introdução ao Computador
- Hardware
  - ❖ Unidade de Entrada
  - ❖ Unidade de Saída
  - ❖ Memória Principal
  - ❖ CPU
- Software
  - ❖ Algoritmos
  - ❖ Programas
  - ❖ Linguagem de Programação
  - ❖ Compilador
  - ❖ Sistemas Operacionais

Vocabulário

- UCP ou CPU
  - ❖ Unidade Central de Processamento (Central Processing Unit)
  - ❖ Responsável pelo processamento de informações
  - ❖ Controla o fluxo de informações (dados)



Vocabulário

- Hardware
  - ❖ A parte física (palpável) da máquina
  - ❖ Composta por gabinetes, teclados, monitor, etc.

- Software
  - ❖ Programas, aplicativos, sistemas operacionais

Vocabulário

- Dados Analógicos
  - Os sinais que nós enviamos para nos comunicarmos são dados. Nosso dia a dia tem muitas formas de dados: sons, letras, números e outros símbolos (escritos ou impressos), fotografias, gráficos, filmes, etc.
  - Todos estes dados são na sua natureza analógicos, o que significa que eles são variados nos seus tipos.
  - Desta forma eles são inúteis em um computador.
  - O computador somente pode processar formatos de dados concisos e simples.

Vocabulário

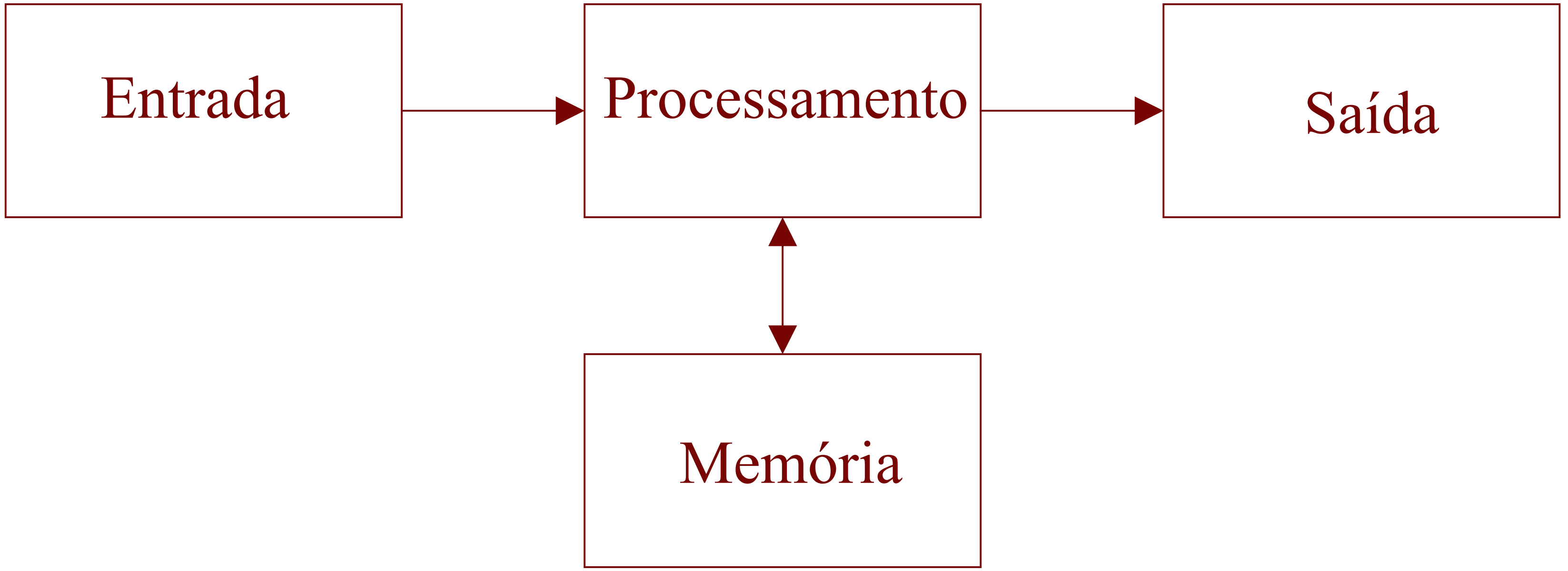
- Dados Digitais
  - O computador é uma unidade elétrica, então ele somente pode manipular dados, os quais são associados com eletricidade.
  - Isto pode ser associado a interruptores elétricos que podem estar ligados ou desligados. Se o interruptor estiver desligado, o computador obterá o valor numérico 0 (zero). Se o interruptor estiver ligado, será obtido o valor numérico 1.
  - Uma outra analogia pode ser feita para obter dados de saída: uma lâmpada acessa pode representar o valor numérico 1 (um), enquanto que uma lâmpada apagada representa o valor numérico 0 (zero).

Conceitos

- Computador: máquina capaz de receber, armazenar, recuperar, processar e exibir informações

DADOS → **COMPUTADOR** → **INFORMAÇÃO**

Fluxo de Informação



Computador

Unidades de Entrada

- Equipamentos utilizados para introduzir dados no computador
- Ex.:
  - teclado
  - mouse
  - scanner
  - unidade de fita magnética
  - unidade de disco magnético

Unidades de Saída

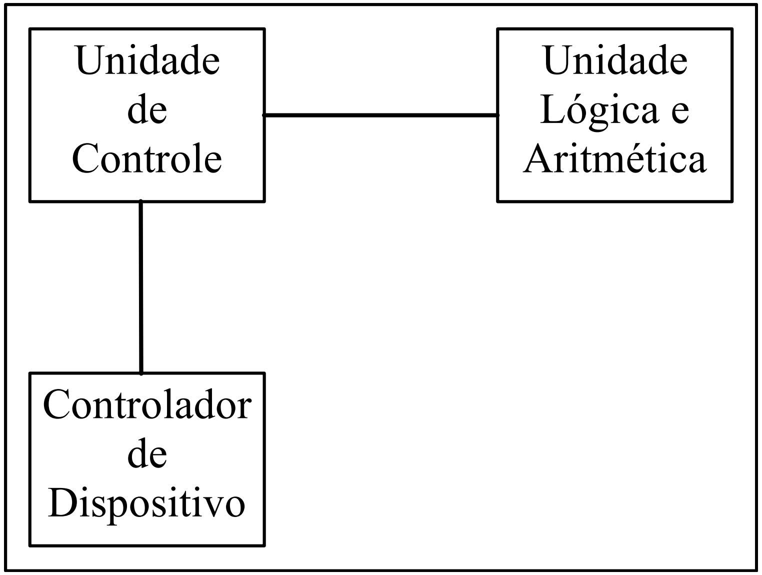
- Equipamentos utilizados para externar os resultados do processamento dos dados
- Ex.:
  - monitor de vídeo
  - impressora
  - unidade de fita magnética
  - disco magnético

Memória Principal

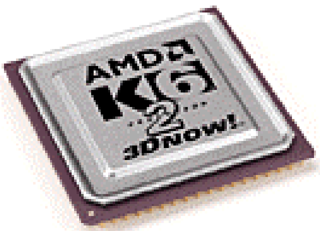
- Dispositivo eletrônico interno de armazenamento temporário
- Atua como se fosse uma área de trabalho para a CPU
- Armazena dados intermediários ou finais resultantes do processamento
- Memória Secundária:
  - Acesso lento;
  - Recurso abundante e barato;

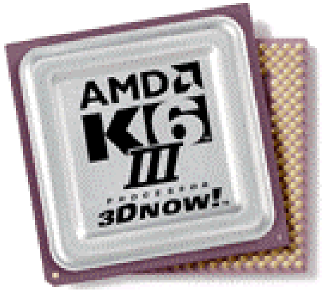
UCP ou CPU

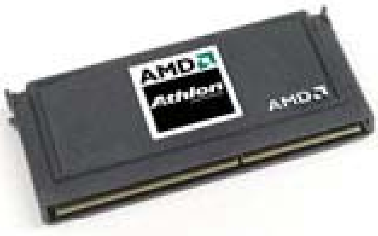
- Unidade Central de Processamento
- Contém os circuitos responsáveis pela interpretação/execução das instruções e pelo controle de fluxo de dados



CPU







Software

- Algoritmos
- Programas
- Linguagens de Programação
- Compilador
- Sistemas Operacionais

Algoritmos

- Seqüência de passos com objeto de realizar determinada tarefa
- “Receita de Bolo”
- Pode ser escrito sem preocupação com sintaxe, linguagem ou computador
- Deve prever todos os possíveis eventos e ocorrências

Algoritmo – Exemplo

- Exemplo: Calcular o salário médio em uma companhia.
  - Descubra quanto ganha cada pessoa
  - Conte quantos empregados você tem
  - Totalize os salários
  - Divida o total pelo número de empregados.

Linguagem de Programação

- Texto formal para representar o algoritmo
- Utiliza rígidas normas de escrita
- Impossibilidade de ambiguidade

Exemplos de Linguagem de Programação

- Fortran – destinado a aplicações matemáticas
- Cobol – destinado a aplicações comerciais e financeiras
- Basic – simples de implementar, sintaxe elementar
- Lisp – destinado a manipulação de listas e símbolos
- Pascal – linguagem simples destinada ao aprendizado de programação
- C – liguagem de uso geral, extremamente eficiente e rápida

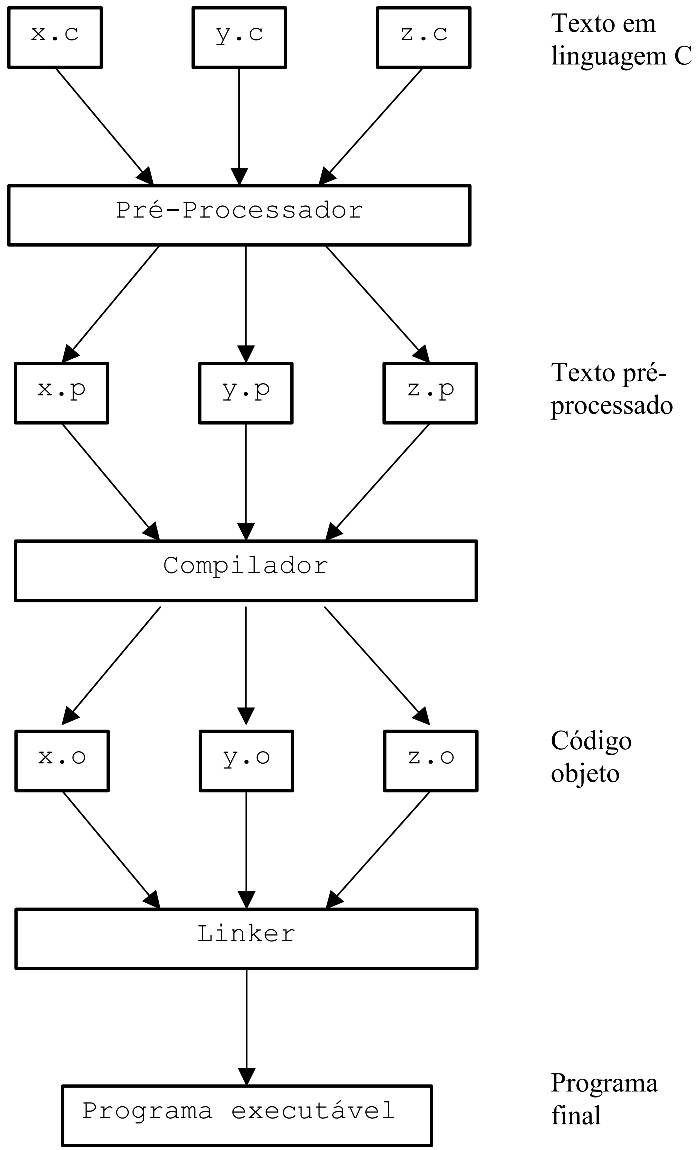
Programas

- Seqüência de comandos em código de máquina
- Texto em binário para a CPU
- Programa em linguagem de alto nível
  - a=10;
  - b=20;
  - c=a+b;
- Programa pronto para ser executado pela CPU
- Linguagem de máquina
  - 1100011100000110110001100000000100000101000000001100
    0111000001101100100000000010000101000000000010100001
    1100011000000010000000110000011011001000000000101010
    0011110010100000010

Compilador

- Converte um texto de um programa em um programa executável
- Adiciona bibliotecas
- Verifica erros de sintaxe
- Não verifica erros de lógica

Compilação


x.c
y.c
z.c
Texto em linguagem C
Pré-Processador
x.p
y.p
z.p
Texto pré-processado
Compilador
x.o
y.o
z.o
Código objeto
Linker
Programa executável
Programa final

Sistemas Operacionais

- O computador sempre está executando algum programa.
- Quando ele é ligado, o computador executa o programa de carga do Sistema Operacional.
- O Sistema Operacional é um programa que facilita a interface entre o operador do computador e o hardware.
- O sistema operacional tem como tarefas permitir que o usuário selecione programa para executar, gerenciar o armazenamento de dados nos discos, facilitar entrada e saída de dados dos programas, etc.

Introdução à Informática

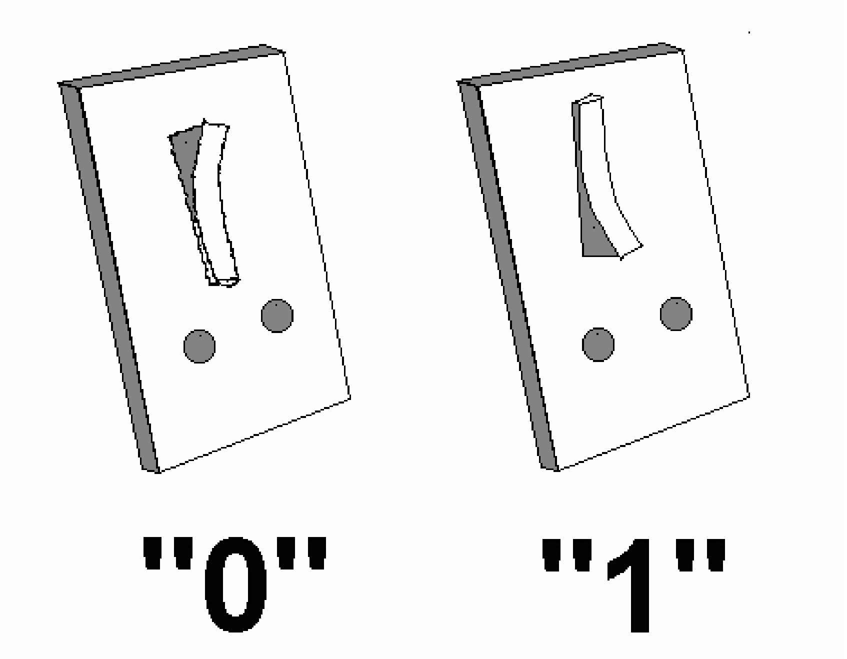
Alexandre Meslin
(meslin@nce.ufrj.br)

Aula 2

- Bit
- Byte
- Unidades e seus Multiplicadores
  - Armazenamento
  - Freqüência
  - Tempo

Unidades – bits

- Menor unidade de armazenamento
- Somente pode representar números de 0 até 1
- Poder estar em um entre dois estados
  - ❖Acesso – apagado
  - ❖Aberto – fechado
  - ❖Ligado – desligado
  - ❖0 – 1



Conjunto de bits

- 2 bits
  - 4 combinações
  - 00 01 10 11
  - Pode ser associado a 4 números
  - 0 até 3 (0, 1, 2, 3)
  - -2 até 1 (-2, -1, 0, 1)
- 3 bits
  - 8 combinações
  - 000 001 010 011 100 101 110 111
  - Pode ser associado a 8 números
  - 0 até 7 (0, 1, 2, 3, 4, 5, 6, 7)
  - -4 até 3 (-4, -3, -2, -1, 0, 1, 2, 3)

Unidades – byte

- Nome dado a um conjunto de 8 bits
- Pode assumir 256 valores diferentes.
- Usado como medida de capacidade de armazenamento de informações.
- Pode ser associado a 256 números
  - 0 até 255
  - -128 até 127

Multiplicadores e Divisores

- Multiplicadores convencionais

❖ 10^1	deca	da	decalitro (dal)
❖ 10^2	hecto	h	hectograma (hg)
❖ 10^3	quilo	k	quilometro (km)

- Multiplicadores não muito convencionais

❖ 10^6	mega	M
❖ 10^9	giga	G
❖ 10^{12}	tera	T
❖ 10^{15}	peta	P
❖ 10^{18}	exa	E
❖ 10^{21}	zeta	Z
❖ 10^{24}	yota	Y

Multiplicadores e Divisores

- Divisores convencionais
  - 10^{-1} deci d decigrama (dg)
  - 10^{-2} centi c centímetro (cm)
  - 10^{-3} mili m mililitro (ml)
- Divisores não muito convencionais
  - 10^{-6} micro μ
  - 10^{-9} nano η
  - 10^{-12} pico p
  - 10^{-15} fento f
  - 10^{-18} ato a
  - 10^{-21} zepto z
  - 10^{-24} yocto y

Unidades – Multiplicadores

- Atualmente
  - ❖ 1 bit
  - ❖ 1 byte = 8 bits
  - ❖ 1 Kbyte = 1024 bytes (4 algarismos)
  - ❖ 1 Mbyte = 1024 Kbytes = 1048576 bytes (7 algarismos)
  - ❖ 1 Gbyte = 1024 Mbytes (gigabyte) (10 algarismos)
  - ❖ 1 Tbyte = 1024 Gbytes (terabyte) (13 algarismos)
- Para o futuro
  - ❖ 1 Pbyte = 1024 Tbytes (petabyte) (16 algarismos)
  - ❖ 1 Ebyte = 1024 Pbytes (exabyte) (19 algarismos)
  - ❖ 1 Zbyte = 1024 Ebytes (zetabyte) (22 algarismos)
  - ❖ 1 Ybyte = 1024 Zbyte (yotabyte) (25 algarismos)

Capacidades Média de Armazenamento

- Disquete de 3½”
  - 1,44 Mbytes
- Zip disk
  - 100 Mbytes
- Memória de Computador Doméstico
  - 128 Mbytes
- Discos magnéticos
  - 40 Gbytes

Representação de Números Inteiros

- 1 byte – 8 bits
  - Números de 0 até 255
  - Números de –128 até 127
- 2 bytes – 16 bits
  - Números de 0 até 65535
  - Números de –32768 até 32767
- 4 bytes – 32 bits
  - Números de 0 até 4.294.967.295
  - Números de –2.147.483.648 até 2.147.483.647
- Mais detalhes no módulo 2

Representação de Números Reais

- Números reais representados usando notação mantissa/expoente

S	**Expoente**	**➉Parte Fracionária**

Representação de Símbolos de Grafia

- Necessidade de armazenar os símbolos de grafia em forma de dados binários
- Símbolos de grafia (caracteres): letras, números, pontuação, acentuação, espaço, etc
- Utilização de tabelas de conversão
  - EBCDIC
  - ASCII
  - UNICODE

Tabela ASCII

- Tabela que contém a maior parte dos caracteres ocidentais
- Possui 256 caracteres
- Problemas de compatibilidade de implementação entre países que posuem letras acentuadas e/ou diferentes do alfabeto inglês.
- Primeira parte (caracteres de 0 até 127) padronizada
- Segunda parte (caracteres de 128 até 255) com divergência entre implementações
- Para maiores informações, consulte: http://www.neurophys.wisc.edu/www/comp/docs/ascii.html

Divisões da Tabela ASCII

- Pode ser dividida em 2 grandes partes
- Primeira parte
  - ❖ Caracteres entre 0 e 127
  - ❖ Padrão mundial
  - ❖ Não possui letras acentuadas
  - ❖ Somente caracteres da lingua inglesa
- Segunda parte
  - ❖ Não está totalmente utilizada
  - ❖ Possui caracteres acentuados e outros símbolos
  - ❖ Foi acrescentada recentemente

Divisões da Tabela ASCII

- Primeira Parte: dividida em 4 áreas
  - Posições de 0 até 31
    - Caracteres de controle
  - Posições de 32 até 63
    - Alguns caracteres de pontuação !"#$%&'()*+,-./
    - Caracteres numéricos em ordem alfabética 0123456789
    - Mais caracteres de pontuação :;<=>?
  - Posições de 64 até 95
    - Caracter arroba @
    - Letras maiúsculas ABCDEFGHIJKLMNOPQRSTUVWXYZ
    - Mais símbolos [\]^_
  - Posições de 96 até 127
    - Crase `
    - Letras minúsculas abcdefghijklmnopqrstuvwxyz
    - Mais símbolos {|}~

Tabela ASCII

dec.	hex.	octal	ASCII	mnm.	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII
0	00	000	^@	NUL	32	20	040		64	40	100	@	96	60	140	`
1	01	001	^A	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	^B	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	^C	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	^D	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	05	005	^E	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	^F	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	07	007	^G	BELL	39	27	047	'	71	47	107	G	103	67	147	g
8	08	010	^H	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	09	011	^I	HTAB	41	29	051	)	73	49	111	I	105	69	151	i
10	0A	012	^J	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	^K	VTAB	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	^L	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	^M	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	^N	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	^O	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	^P	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	^Q	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	^R	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	^S	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	^T	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	^U	NACK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	^V	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	^W	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	^X	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	^Y	EN	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	^Z	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	^[	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	^\	FS	60	3C	074	<	92	5C	134	\	124	7C	174	|
29	1D	035	^]	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	^^	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	^_	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Tabela ASCII – Primeiro Grupo

dec.	hex.	octal	ASCII	mnm.	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII
0	00	000	^@	NUL	32	20	040		64	40	100	@	96	60	140	`
1	01	001	^A	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	^B	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	^C	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	^D	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	05	005	^E	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	^F	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	07	007	^G	BELL	39	27	047	'	71	47	107	G	103	67	147	g
8	08	010	^H	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	09	011	^I	HTAB	41	29	051	)	73	49	111	I	105	69	151	i
10	0A	012	^J	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	^K	VTAB	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	^L	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	^M	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	^N	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	^O	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	^P	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	^Q	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	^R	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	^S	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	^T	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	^U	NACK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	^V	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	^W	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	^X	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	^Y	EN	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	^Z	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	^[	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	^\	FS	60	3C	074	<	92	5C	134	\	124	7C	174	|
29	1D	035	^]	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	^^	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	^_	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Tabela ASCII – Segundo Grupo

dec.	hex.	octal	ASCII	mnm.	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII
0	00	000	^@	NUL	32	20	040		64	40	100	@	96	60	140	`
1	01	001	^A	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	^B	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	^C	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	^D	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	05	005	^E	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	^F	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	07	007	^G	BELL	39	27	047	'	71	47	107	G	103	67	147	g
8	08	010	^H	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	09	011	^I	HTAB	41	29	051	)	73	49	111	I	105	69	151	i
10	0A	012	^J	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	^K	VTAB	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	^L	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	^M	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	^N	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	^O	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	^P	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	^Q	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	^R	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	^S	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	^T	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	^U	NACK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	^V	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	^W	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	^X	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	^Y	EN	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	^Z	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	^[	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	^\	FS	60	3C	074	<	92	5C	134	\	124	7C	174	|
29	1D	035	^]	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	^^	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	^_	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Tabela ASCII – Terceiro Grupo

dec.	hex.	octal	ASCII	mnm.	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII
0	00	000	^@	NUL	32	20	040		64	40	100	@	96	60	140	`
1	01	001	^A	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	^B	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	^C	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	^D	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	05	005	^E	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	^F	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	07	007	^G	BELL	39	27	047	'	71	47	107	G	103	67	147	g
8	08	010	^H	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	09	011	^I	HTAB	41	29	051	)	73	49	111	I	105	69	151	i
10	0A	012	^J	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	^K	VTAB	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	^L	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	^M	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	^N	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	^O	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	^P	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	^Q	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	^R	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	^S	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	^T	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	^U	NACK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	^V	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	^W	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	^X	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	^Y	EN	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	^Z	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	^[	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	^\	FS	60	3C	074	<	92	5C	134	\	124	7C	174	|
29	1D	035	^]	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	^^	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	^_	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Tabela ASCII – Quarto Grupo

dec.	hex.	octal	ASCII	mnm.	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII	dec.	hex.	octal	ASCII
0	00	000	^@	NUL	32	20	040		64	40	100	@	96	60	140	`
1	01	001	^A	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	^B	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	^C	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	^D	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	05	005	^E	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	^F	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	07	007	^G	BELL	39	27	047	'	71	47	107	G	103	67	147	g
8	08	010	^H	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	09	011	^I	HTAB	41	29	051	)	73	49	111	I	105	69	151	i
10	0A	012	^J	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	^K	VTAB	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	^L	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	^M	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	^N	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	^O	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	^P	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	^Q	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	^R	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	^S	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	^T	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	^U	NACK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	^V	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	^W	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	^X	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	^Y	EN	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	^Z	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	^[	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	^\	FS	60	3C	074	<	92	5C	134	\	124	7C	174	|
29	1D	035	^]	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	^^	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	^_	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Unidades de Tempo

- Hertz: utilizado para explicitar a velocidade do processador, de barramentos e, atualmente, de memórias
- 1 Hz = 1 ciclo por segundo
- 1 Hz = 1/(1s)
- 1 KHz = 1000 Hz
- 1 MHz = 1000 KHz
- 1 GHz = 1000 MHz

Freqüências

- Rotação do motor do carro – 15 Hz (900 rpm)
- Energia elética – 60 Hz
- Rotação do disco do HD – 120 Hz (7200 rpm)
- Ciclo do 8088 (primeiro PC) – 4,77 MHz
- Ciclo do 286 – 16 MHz
- Ciclo do 386 – 40 MHz
- Rádio FM – 88 MHz – 108 MHz
- Ciclo do 486 – 120MHz
- Ciclo do Pentium – 233 MHz
- Ciclo do PII – 450 MHz
- Ciclo do PIII – 900 MHz
- Ciclo do PIV – 2GHz (2 bilhões de ciclos por segundo)

Unidades de Tempo

- Segundos
- Utilizado para marcar o tempo de acesso a dispositivos (memória, disco, etc)
- 1 s
- 1 ms = 0,001 s (milisegundo)
- 1 μs = 0,001 ms (microsegundo)
- 1 ηs = 0,001 μs (nanosegundo)
- 1 ps = 0,001 ηs (picosegundo)

Exemplos de Tempos

- Copo caindo de cima da mesa – pouco menos de 1 segundo
- Tempo para cabeça de HD mudar de trilha – 4 ms
- Leitura de 1 byte da memória – 60 ηs
- Leitura entre byte consecutivos da memória – 10 ηs

Período x Freqüência

- Comparação entre CPU e módulos de memória
- Processador
  - Velocidade medida em hertz (Hz)
  - Atualmente entre 100 MHz e 2 GHz
  - Ou seja, 1 ciclo = 10 ηs – 500 ps
- Memória de baixo custo
  - Memória lenta
  - 150 ηs - 60 ηs
  - Capacidade máxima: 512 Mbytes
  - US$1,00/Mbyte
  - Baixo consumo de energia
- Memória 6-120 vezes mais lenta que processador

Período x Freqüência

- Comparação entre CPU e memórias cache
- Processador
  - ❖ Velocidade medida em Hertz (Hz)
  - ❖ Atualmente entre 100 MHz e 2 GHz
  - ❖ Ou seja, 1 ciclo = 10 ηs – 500 ps
- Memória de alta velocidade
  - ❖ Memória rápida
  - ❖ 10 ns - 6ns
  - ❖ Capacidade máxima: 4 Mbytes
  - ❖ US$20,00/Mbyte
  - ❖ Alto consumo de energia
- Memória 12-20 vezes mais lenta que processador

Dificuldades e Soluções

- Dificuldades:
  - Interface do circuito em acionar sinais externos
  - Degradação do sinal ao percorrer a placa que interligas os circuitos integrados
  - Sensibilidade dos circuitos em receberem os sinais degradados
- Solução
  - Inclusão de memórias no interior do processador
  - Criação de diversos conjuntos de memórias para serem acessados em paralelo

Introdução à Informática

Alexandre Meslin
(meslin@nce.ufrj.br)

Aula 3

- Periféricos
  - Vídeo
  - Teclado
  - Disco
  - Mouse

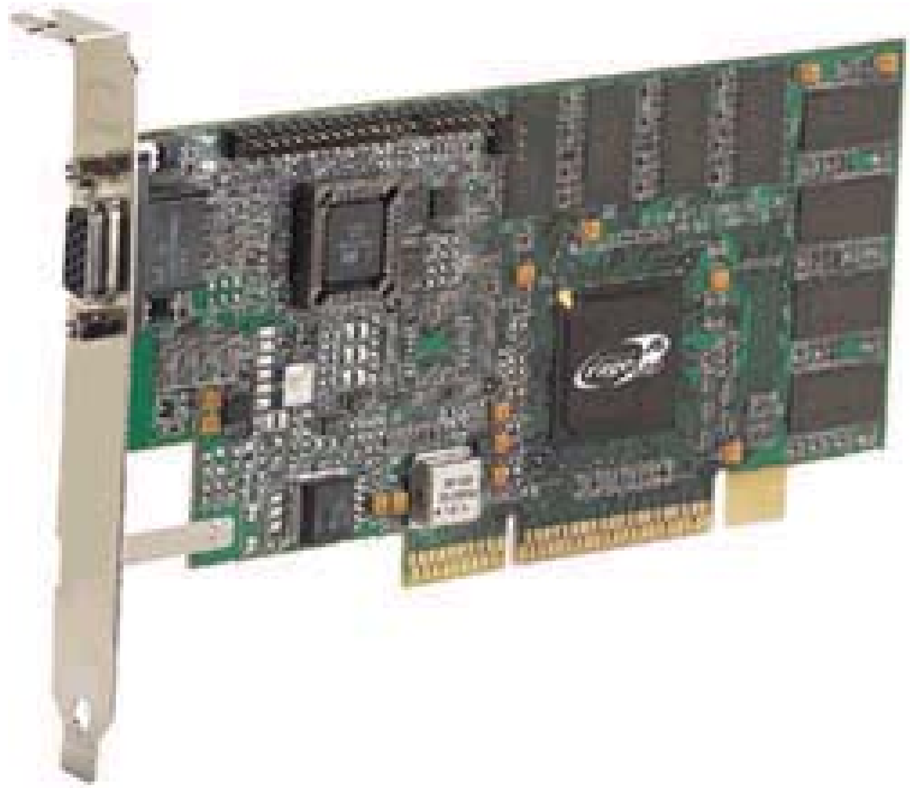
Periféricos

- Entrada
  - Digitalizam informações analógicas do mundo exterior para a CPU
- Saída
  - Convertem dados digitais da CPU para o formato analógico do mundo exterior

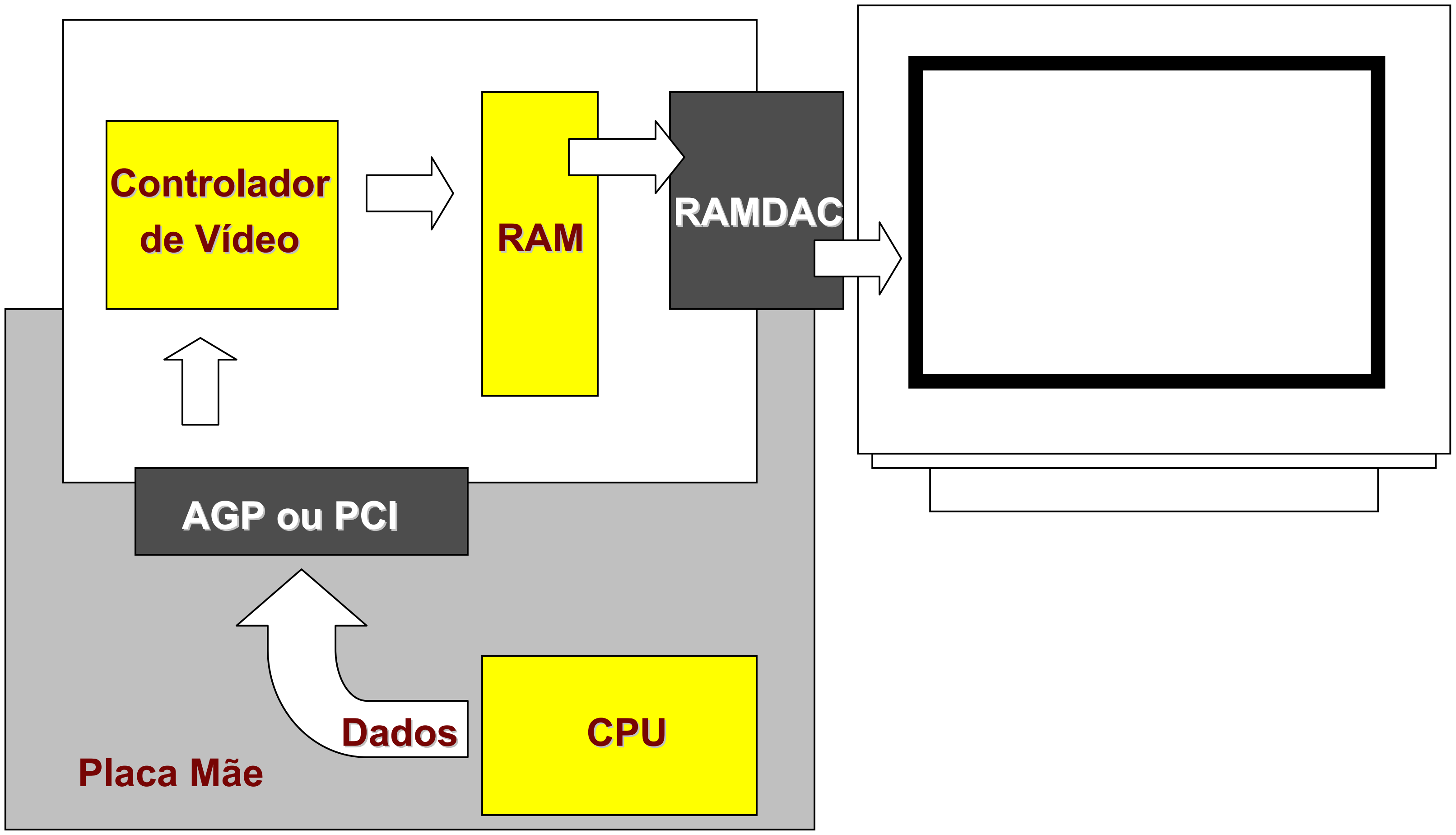
Interface de Vídeo

- Conecta-se no barramento da placa mãe
- Escolha uma placa de vídeo compatível com o monitor e com o barramento
- Possuem
  - ❖Um conector do barramento do PC
  - ❖Um conector de vídeo (15 ou 9 pinos fêmea)

Placa de Vídeo



Placa de Vídeo



Interface de Vídeo

- Característica
  - Padrão (CGA/EGA/VGA/SVGA)
  - Número de cores
  - Resolução
  - Entrelaçamento
  - Taxa de atualização (refresh rate)
  - Tamanho da Memória
  - Capacidade de Processamento
  - Barramento

Padrão CGA/EGA/VGA/SVGA

- CGA
  - ❖ Muito antigo
  - ❖ Gráficos de baixa resolução
  - ❖ Amplamente utilizados nos primeiros computadores PC
  - ❖ Até 16 cores
  - ❖ Resolução até 640 x 200 pontos
  - ❖ (fora do mercado)

Padrão CGA/EGA/VGA/SVGA

- EGA
  - Sucessor da interface CGA
  - Permite gráficos de mais alta resolução e mais cores
  - Vida muito curta
  - Até 256 cores
  - Resolução até 640 x 480 pontos
  - (fora do mercado)

Padrão CGA/EGA/VGA/SVGA

- VGA
  - Interface gráfica padrão de muitas máquinas
  - Permite resolução de até 800 por 600 pontos
  - Número maior de cores (até 256 cores)
  - Substituida pelo padrão SVGA
  - Configuração básica de toda placa de vídeo atual

Padrão CGA/EGA/VGA/SVGA

- SVGA
  - Implemetações particulares de diversos fabricantes
  - Tipo de interface de vídeo utilizada atualmente
  - Permite grande variedade de resoluções
  - Possui grande quantidade de cores possíveis simultaneamente
  - Não existe padrão de software para interface – necessita de driver específico

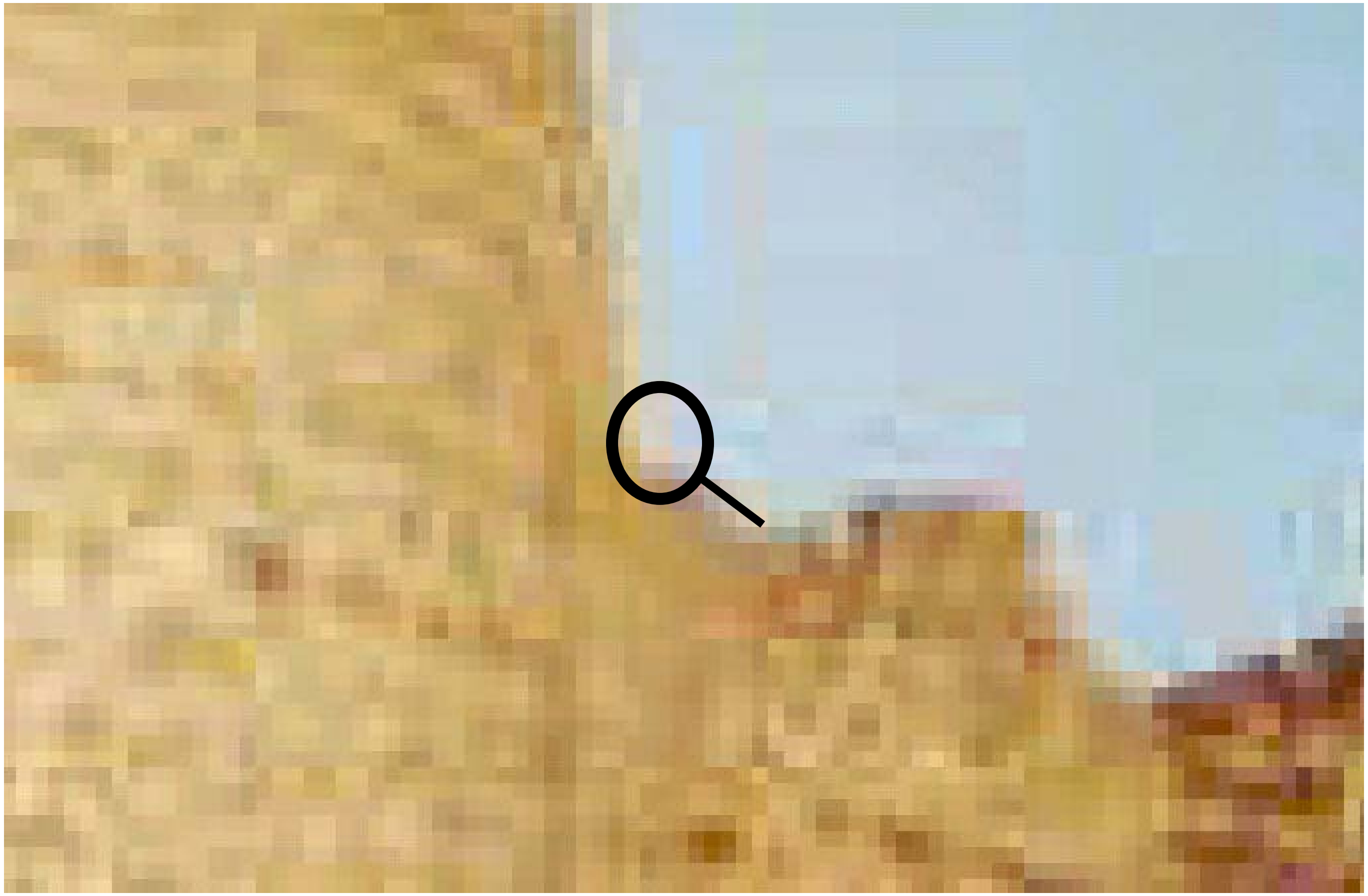
Cores e Resolução de Vídeo

- Imagem representada por pontos
- Tela formada por matriz de pontos

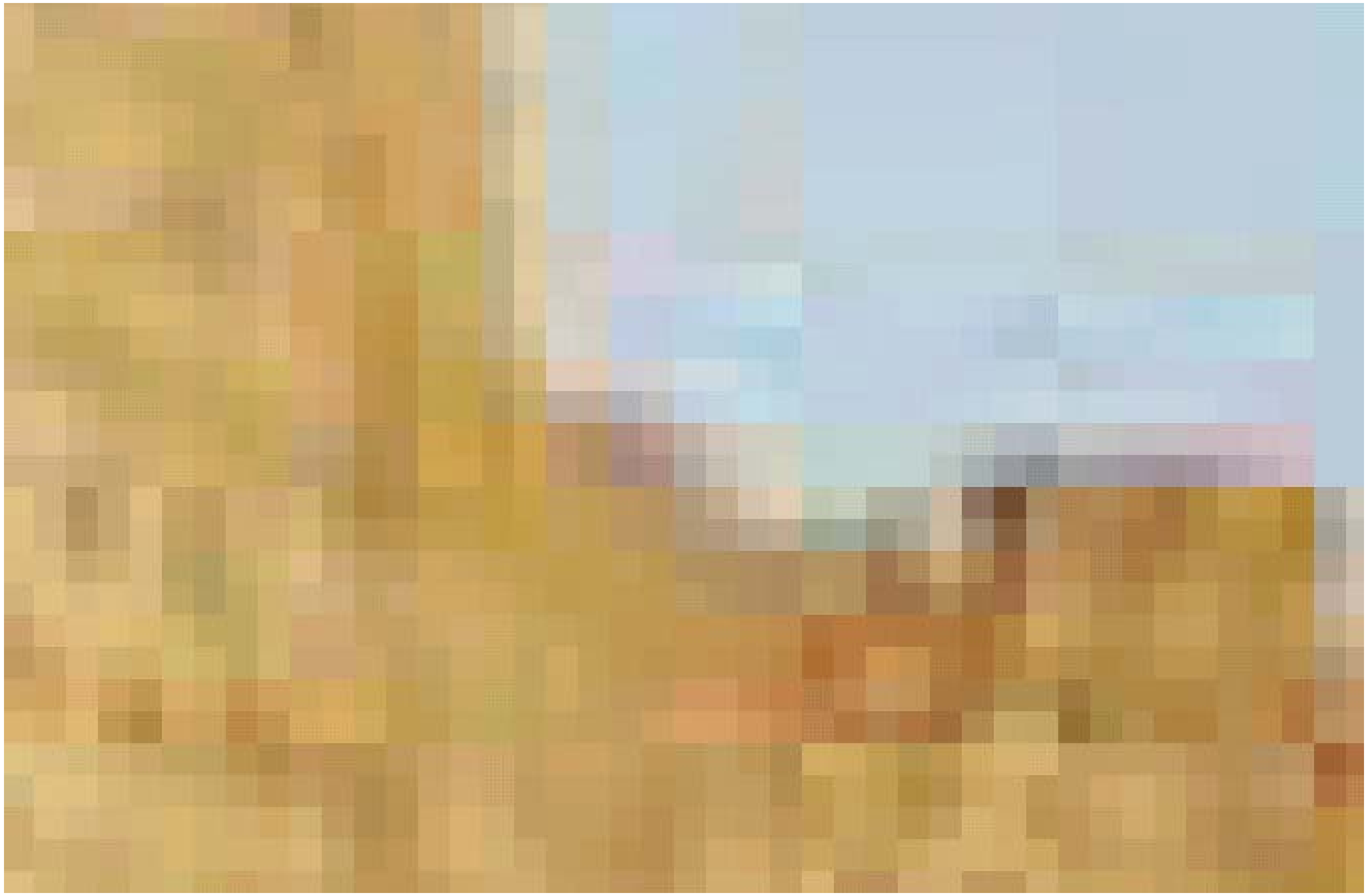
Cores e Resolução de Vídeo

Cores e Resolução de Vídeo

Cores e Resolução de Vídeo



Cores e Resolução de Vídeo

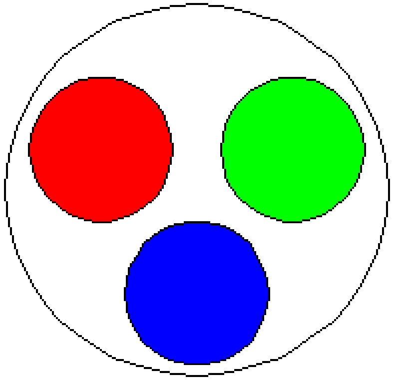


Cores e Resolução de Vídeo

- Cada ponto com características próprias
- Profundidade
  - Quantidade de cores por ponto
- Resolução
  - Quantidade de pontos horizontais e verticais

Cores de Vídeo

- Cada ponto em um monitor colorido é formado pela soma de 3 cores
  - Vermelho
  - Azul
  - Verde



Cores de Vídeo

- Bits por ponto
  - ❖1 bit – ponto acesso ou apagado
  - ❖2 bits – 4 cores
  - ❖4 bits – 16 cores
  - ❖8 bits – 256 cores
  - ❖16 bits – 65536 cores (64 k cores)
  - ❖24 bits – 16 Mega cores (8 bits para cada cor primária)
  - ❖32 bits – 4 Giga cores

Interface de Vídeo – Cores

- Relação entre cor/memória/resolução

Memória	Resolução	Número de Cores
1 Mbyte	640x480	256/64k/16M
	800x600	256k/64k
2 Mbytes	800x600	256k/64k/16M
	1024x768	256k/64k
3 Mbytes	1024x768	256k/64k/16M
4 Mbytes	1024x768	256k/64k/16M/4G

Interface de Vídeo - Resolução

⑩1280 x 1024 pontos = 1310720 pontos

⑩1024 x 768 pontos = 786432 pontos

⑩800 x 600 pontos =

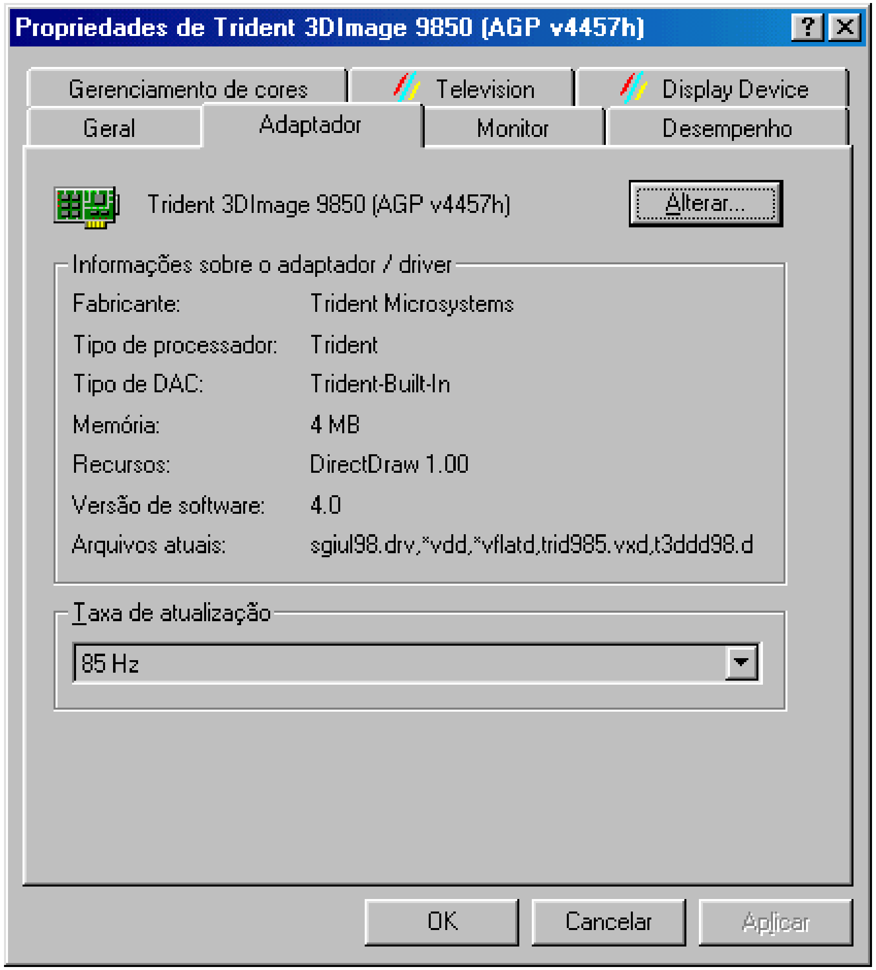
⑩480000 pontos

Interface de Vídeo

- Entrelaçamento
  - ❖ Apresentação de linhas em quadros alternados
  - ❖ Tela cintilante
- Atualização da Imagem
  - ❖ Atualização vertical
  - ❖ Atualização horizontal

Atualização da Imagem

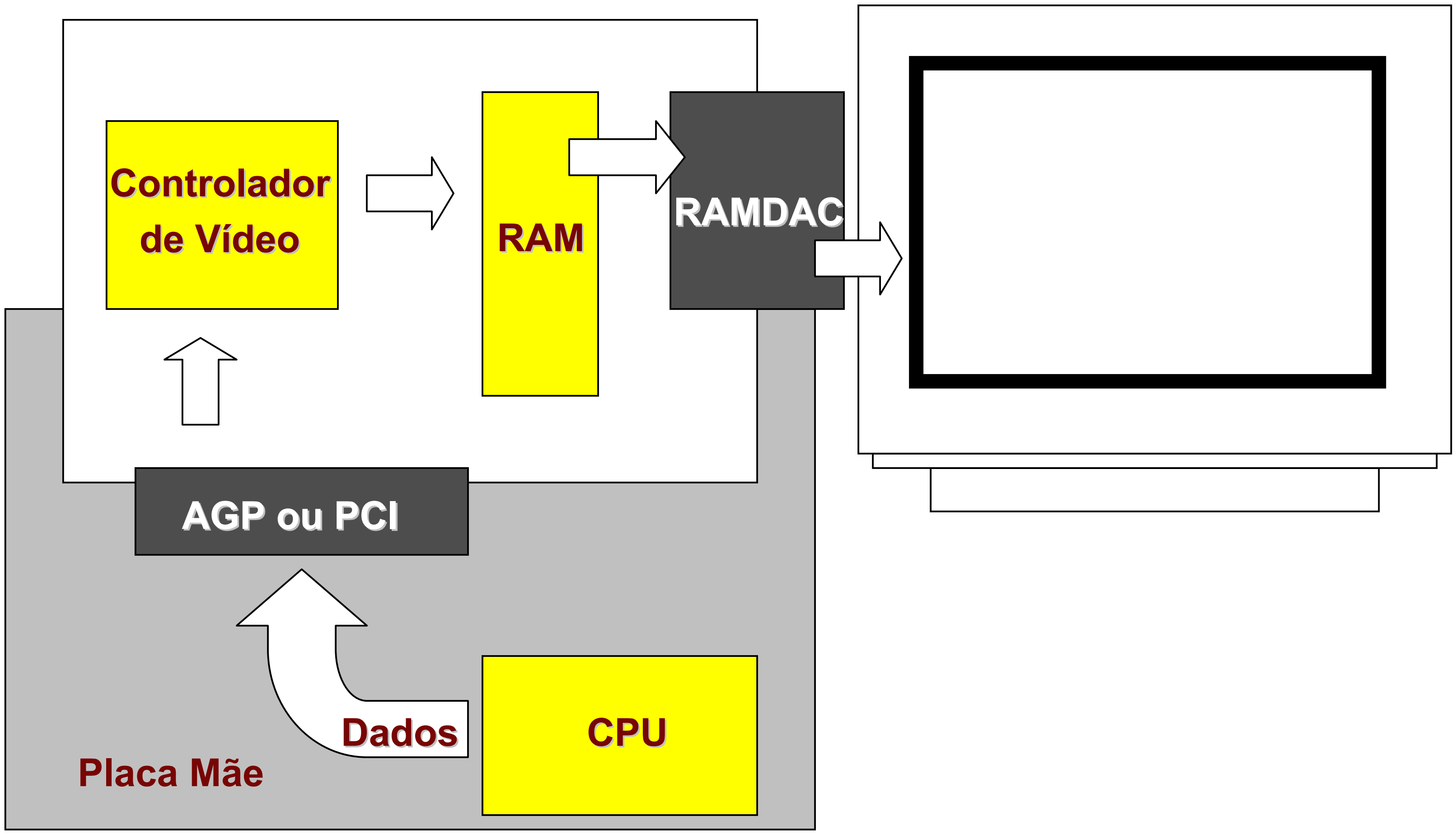
- A taxa de atualização pode ser configurada:
  - ❖ Painel de Controle →
  - ❖ Vídeo →
  - ❖ Configurações →
  - ❖ Avançadas →
  - ❖ Adaptador →
  - ❖ Taxa de atualização



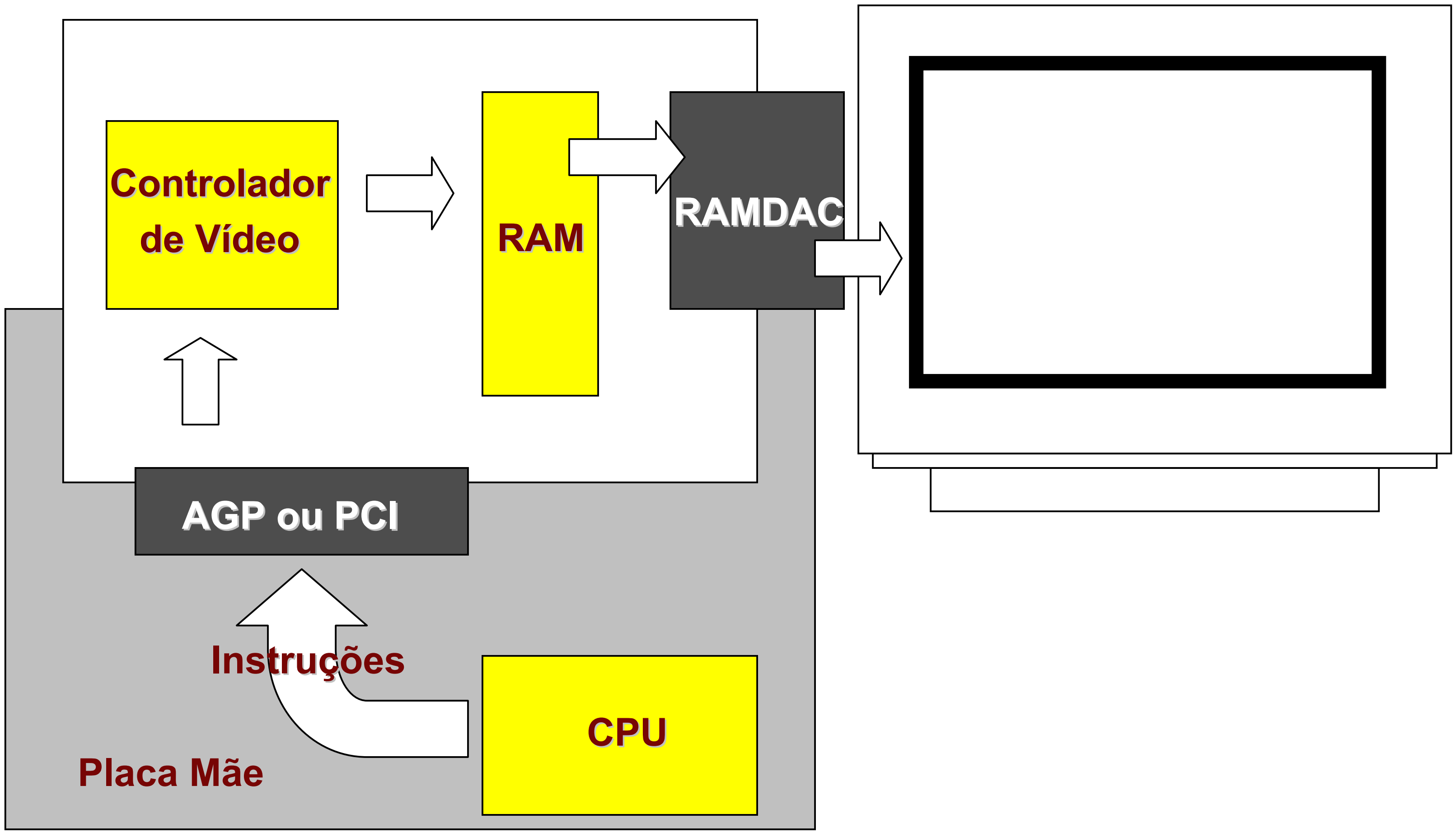
Interface de Vídeo

- Capacidade de Processamento
  - Imagens 2D
  - Imagens 3D

Capacidade de Processamento



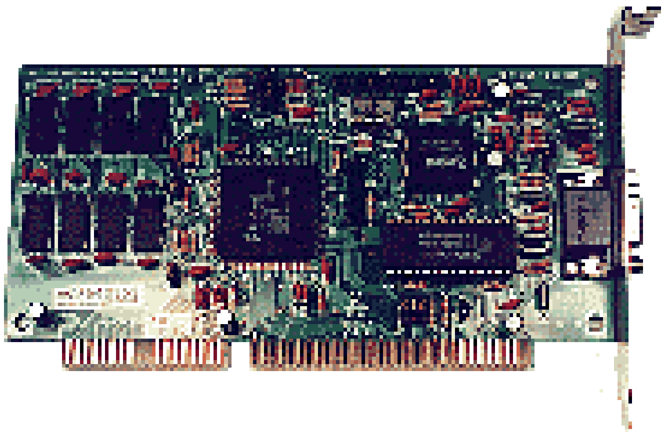
Capacidade de Processamento



Interface de Vídeo

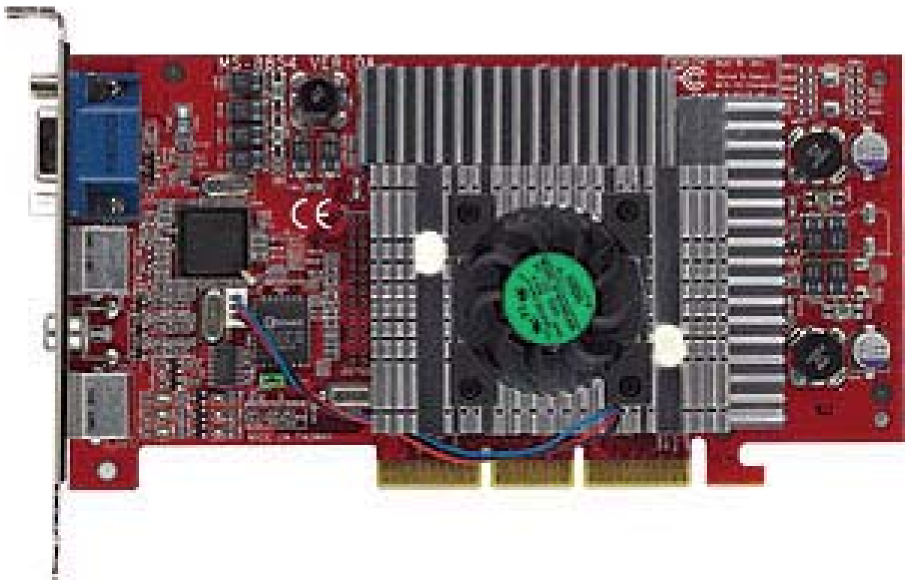
- Tipo de Barramento
  - ISA
  - VLB
  - PCI
  - AGP

Interface de Vídeo ISA

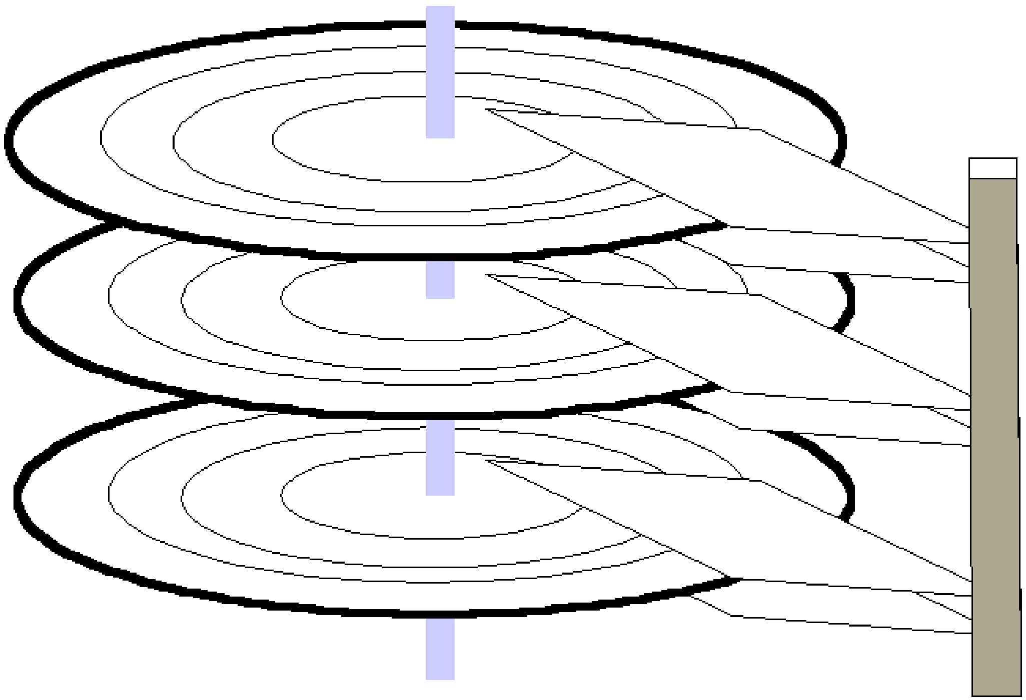


Interface de Vídeo PCI

Interface de Vídeo AGP



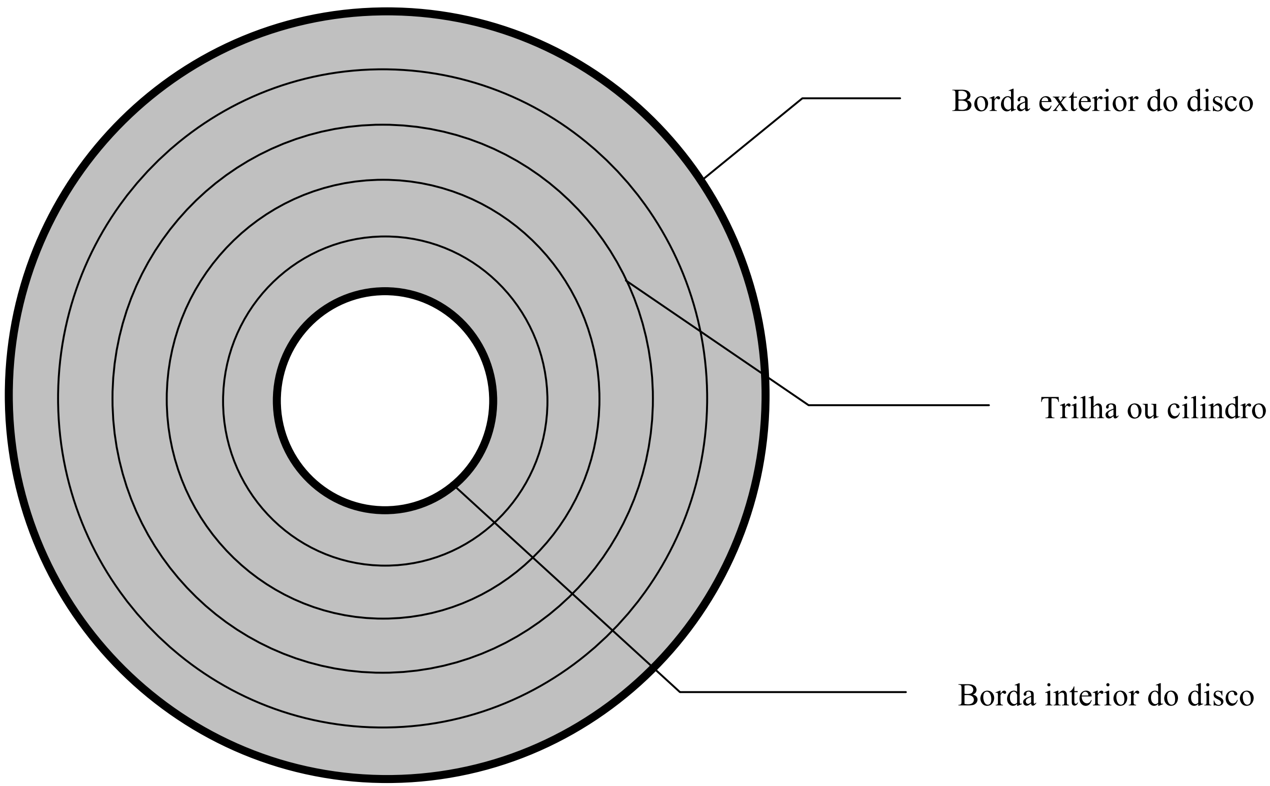
Disco



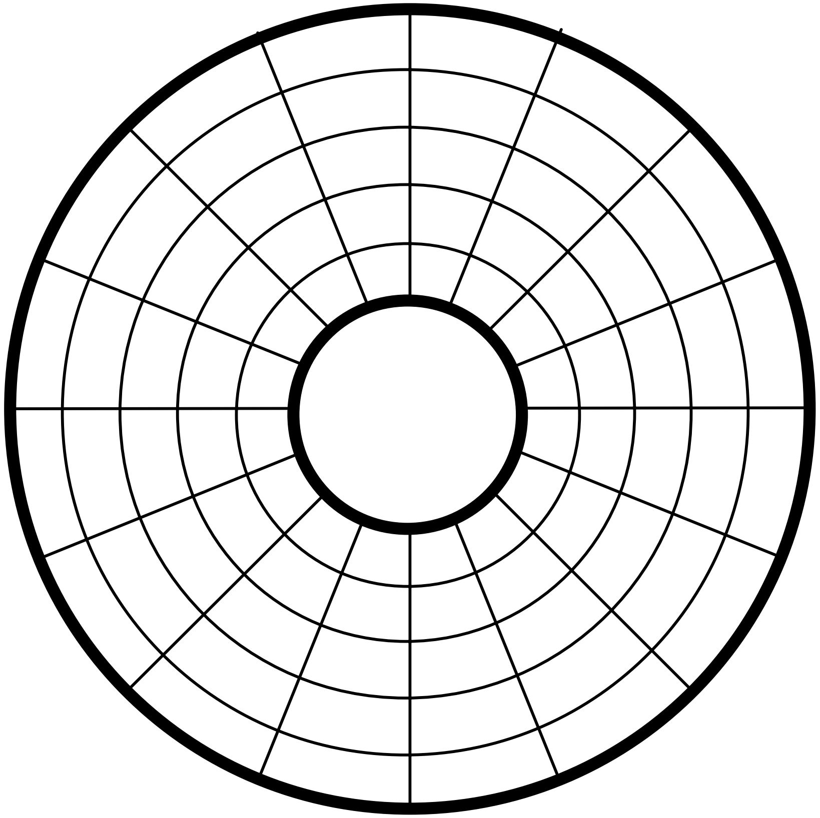
Disco – Interfaces

- MFM, RLL – fora do mercado
- IDE – substituida pela EIDE
- EIDE – melhoramento da interface IDE antiga, também conhecida por apenas IDE
- SCSI – interface de alto desempenho e custo. Utilizada em sistemas profissionais

Disco – Trilhas



Discos – Setores



Teclado

- 86 teclas (XT)
- 101 teclas (AT/XT)
- 102 teclas (AT/XT)
- Chave de comutação: AT-XT
- Conexão

Mouse

Final

- Exercícios

Introdução à Informática

Sistemas Numéricos
Representações Numéricas

Ageu Pacheco e Alexandre Meslin

cederj

Sistemas Numéricos
Representações Numéricas

- Objetivo do Módulo :

  Estudo de outros sistemas numéricos além do decimal visando entendimento e domínio de operações aritméticas do sistema binário (de base 2) e do seu relativo; hexadecimal ou de base16.

Sistemas Numéricos
Representações Numéricas

- Objetivo da Aula:

- Conhecer representações de números em outras bases

Sistemas Numéricos
Sistema Decimal

- Concebido pelos hindus cerca de 2000 anos atrás. Posteriormente foi adotado pelos árabes que o introduziram aos europeus.
- Também denominado sistema arábico porque utiliza símbolos arábicos para representar os dez algarismos ou dígitos (dedo em Latim) que a base suporta: (0,1,2,3,4,5,6,7,8,9).
- Base é a quantidade de símbolos disponíveis para representar os diferentes dígitos do sistema.

Sistemas Numéricos

Sistema Decimal

- A representação de qualquer número na base decimal é posicional; isto é cada dígito assume um valor ponderado à posição que ocupa.

  Ex: $638 = 6 \times 10^2 + 3 \times 10^1 + 8 \times 10^0$

- O valor que cada dígito assume na notação posicional é igual ao seu valor absoluto multiplicado pela base elevada à posição relativa do dígito – 1.

Sistemas Numéricos

Representações Numéricas

- Exemplo de sistema numérico não ponderado: Sistema Romano

  Algarismos romanos:

  I, V, X, L, C, D, M

  1, 5, 10, 50, 100, 500, 1000

  Exemplos de números romanos;

  MCMLXXXIX, MCMXCIX, MM, MMI

Outras Bases

- Outras bases ponderadas utilizando os mesmos símbolos arábicos:

  Exemplos:

- Base 3:

  0,1,2,10,11,12,20,21,22,100,101,102,110…

Sistemas Numéricos

Outras Bases

- Outras bases ponderadas utilizando os mesmos símbolos arábicos:

  Exemplos:

- Base 3:

  0,1,2,10,11,12, 20,21,22,100, 101,102,110…(base 3)

  0,1,2, 3, 4, 5, 6, 7, 8, 9 , 10, 11, 12….(base10)

cederj

Outras Bases

- Exemplos outras bases (cont):

- Base 5:
  0,1,2,3,4,10,11,12,13,14,20,21,22,23,24,30,...

Outras Bases

- Exemplos outras bases (cont):

- Base 5:

0,1,2,3,4,10,11,12,13,14,20,21,22,23,24,30...

↓ ↓ .... ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

0,1,2,3,4, 5, 6, 7, 8, 9, 10,11,12,13,14,15…

Outras Bases

- Exemplos outras bases (cont):

- Base 7:
  0,1,2,3,4,5,6,10,11,12,13,14,15,16,20,21,22,23,24...

Sistemas Numéricos
Outras Bases

- Exemplos outras bases (cont):

- Base 7:

$$
\begin{array}{cccccccccccccccccccc}
0, & 1, & 2,3,4,5,6, & 10, & 11, & 12, & 13, & 14, & 15, & 16, & 20, & 21, & 22, & 23, & 24\ldots \\
\downarrow & \downarrow & \ldots. & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
0, & 1, & 2,3,4,5,6, & 7, & 8, & 9, & 10, & 11, & 12, & 13, & 14, & 15, & 16, & 17, & 18\ldots
\end{array}
$$

Outras Bases

- Exemplos outras bases (cont):

- Base 2:
  0,1,10,11,100,101,110,111,1000,1001,1010,1011...

Sistemas Numéricos
Outras Bases

- Exemplos outras bases (cont):

- Base 2:

0,1,10,11,100,101,110,111,1000,1001,1010,1011,...

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...

Sistemas Numéricos
Outras Bases

- Exemplos outras bases (cont):

- Base 16:
  0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,11,12,13,…

Outras Bases

- Exemplos outras bases (cont):

- Base 16:

0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F, 10,11,12,13,…

↓ ↓ …. ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,…

Sistemas Numéricos
Outras Bases

Concluindo…

- Propriedades dos sistemas numéricos posicionais:
- O número de dígitos usados em qualquer sistema é sempre igual `a base
- O maior dígito é igual ao valor da base menos 1
- O valor que cada dígito assume na notação posicional é igual ao seu valor absoluto multiplicado pela base elevada à posição relativa do dígito menos 1
- O número que corresponde à base é sempre igual a 10 (um-zero)

Outras Bases

- Assim um número inteiro qualquer <u>N</u> de uma dada base <u>b</u> representado por sua notação posicional:

  $N_b = (A_n A_{n-1} \ldots A_2 A_1 A_0)_b$,

  pode ser expresso em termos quantitativos por:

  $N_b = A_n.b^n + A_{n-1}.b^{n-1} + \ldots\ldots + A_2.b^2 + A_1.b^1 + A_0.b^0$

  (expressão da expansão da notação posicional)

Sistemas Numéricos
Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$

cederj

Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$

Sistemas Numéricos
Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$
3) $7777_8 = 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 = 4095_{10}$

Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$
3) $7777_8 = 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 = 4095_{10}$
4) $4303_5 = 4 \times 5^3 + 3 \times 5^2 + 3 = 578_{10}$

Sistemas Numéricos
Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$
3) $7777_8 = 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 = 4095_{10}$
4) $4303_5 = 4 \times 5^3 + 3 \times 5^2 + 3 = 578_{10}$
5) $4303_{16} = 4 \times 16^3 + 3 \times 16^2 + 3 = 17155_{10}$

Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$
3) $7777_8 = 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 = 4095_{10}$
4) $4303_5 = 4 \times 5^3 + 3 \times 5^2 + 3 = 578_{10}$
5) $4303_{16} = 4 \times 16^3 + 3 \times 16^2 + 3 = 17155_{10}$
6) $21022_3 = 2 \times 3^4 + 1 \times 3^3 + 2 \times 3 + 2 = 197_{10}$

Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$
3) $7777_8 = 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 = 4095_{10}$
4) $4303_5 = 4 \times 5^3 + 3 \times 5^2 + 3 = 578_{10}$
5) $4303_{16} = 4 \times 16^3 + 3 \times 16^2 + 3 = 17155_{10}$
6) $21022_3 = 2 \times 3^4 + 1 \times 3^3 + 2 \times 3 + 2 = 116_{10}$
7) $1011010_2 = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2 = 90_{10}$

Sistemas Numéricos
Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$
3) $7777_8 = 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 = 4095_{10}$
4) $4303_5 = 4 \times 5^3 + 3 \times 5^2 + 3 = 578_{10}$
5) $4303_{16} = 4 \times 16^3 + 3 \times 16^2 + 3 = 17155_{10}$
6) $21022_3 = 2 \times 3^4 + 1 \times 3^3 + 2 \times 3 + 2 = 116_{10}$
7) $1011010_2 = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2 = 90_{10}$
8) $ABC_{16} = 10 \times 16^2 + 11 \times 16 + 12 = 2748_{10}$

cederj

Sistemas Numéricos
Outras Bases

- Exemplos:

1) $426_{10} = 4 \times 10^2 + 2 \times 10^1 + 6 \times 10^0 = 426_{10}$
2) $426_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 = 216_{10}$
3) $7777_8 = 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 = 4095_{10}$
4) $4303_5 = 4 \times 5^3 + 3 \times 5^2 + 3 = 578_{10}$
5) $4303_{16} = 4 \times 16^3 + 3 \times 16^2 + 3 = 17155_{10}$
6) $21022_3 = 2 \times 3^4 + 1 \times 3^3 + 2 \times 3 + 2 = 116_{10}$
7) $1011010_2 = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2 = 90_{10}$
8) $ABC_{16} = 10 \times 16^2 + 11 \times 16 + 12 = 2748_{10}$
9) 504176_7 = não é possível a representação na base 7

cederj

Sistemas Numéricos
Outras Bases

- Representação de números reais:

  Número real em uma dada base b:

  $N_R = (A_n A_{n-1} \ldots A_2 A_1 A_0 \,.\, A_{-1} A_{-2} A_{-3} \ldots A_{-m})$

  $N_R = N_I + N_F$, onde:

  $N_I = A_n . b^n + A_{n-1} . b^{n-1} + \ldots\ldots + A_1 . b^1 + A_0 . b^0$

  $N_F = A_{-1} . b^{-1} + A_{-2} . b^{-2} + \ldots\ldots + A_{-m+1} . b^{-m+1} + A_{-m} . b^{-m}$

Sistemas Numéricos
Outras Bases

- Exemplos:

1) $426.45_{10} = 426 + 4 \times 10^{-1} + 5 \times 10^{-2} = 426.45_{10}$

Sistemas Numéricos
Outras Bases

- Exemplos:

1) $426.45_{10} = 426 + 4 \times 10^{-1} + 5 \times 10^{-2} = 426.45_{10}$

2) $426.45_{7} = 4 \times 7^{2} + 2 \times 7^{1} + 6 \times 7^{0} + 4 \times 7^{-1} + 5 \times 7^{-2}$

$$= 216 + 4 \times \frac{1}{7} + 5 \times \frac{1}{49}$$

$$= 216 + 0.57 + 0.10 = 216.67_{10}$$

Sistemas Numéricos

Outras Bases

- Exemplos:

1) $426.45_{10} = 426 + 4 \times 10^{-1} + 5 \times 10^{-2} = 426.45_{10}$

2) $426.45_7 = 4 \times 7^2 + 2 \times 7^1 + 6 \times 7^0 + 4 \times 7^{-1} + 5 \times 7^{-2}$

$$= 216 + 4 \times \frac{1}{7} + 5 \times \frac{1}{49}$$

$$= 216 + 0.57 + 0.10 = 216.67_{10}$$

3) $1001.1011_2 = 9 + 2^{-1} + 2^{-3} + 2^{-4} =$

$$= 9 + 0.5 + 0.125 + 0.0625 = 9.6875_{10}$$

Sistemas Numéricos

Outras Bases

- Exemplos:

1) $426.45_{10} = 426 + 4 \times 10^{-1} + 5 \times 10^{-2} = 426.45_{10}$

2) $426.45_{7} = 4 \times 7^{2} + 2 \times 7^{1} + 6 \times 7^{0} + 4 \times 7^{-1} + 5 \times 7^{-2}$

$$= 216 + 4 \times \frac{1}{7} + 5 \times \frac{1}{49}$$

$$= 216 + 0.57 + 0.10 = 216.67_{10}$$

3) $1001.1011_{2} = 9 + 2^{-1} + 2^{-3} + 2^{-4} =$

$$= 9 + 0.5 + 0.125 + 0.0625 = 9.6875_{10}$$

4) $1A.1A_{16} = 26 + 0.0625 + 10 \times 0.0039 = 26.10156_{10}$

Introdução à Informática

Sistemas Numéricos
Conversão entre Bases

Ageu Pacheco e Alexandre Meslin

cederj

Sistemas Numéricos

- Objetivo da Aula:

  Conhecer e saber aplicar os métodos de conversão entre bases, com especial ênfase na conversão de números entre as bases 10, 2, e 16.

Conversão entre Bases

- Problema: Dado um número $\underline{N}_s$ expresso em uma base $\underline{s}$ (origem) achar sua representação $\underline{N}_r$ na base $\underline{r}$ (destino).

- Dois métodos:
  - desenvolvimento da notação posicional (polinomial)
  - divisões sucessivas

- Embora ambos os métodos possam ser utilizados para conversão direta entre quaisquer bases $\underline{s}$ e $\underline{r}$, é desejável que uma delas seja a 10. Senão vejamos:

Conversão entre Bases

- Desenvolvimento da notação posicional:

Dado $N_s = A_n A_{n-1} \ldots\ldots A_1 A_0$, N_r é obtido avaliando a expressão $N_r = A_n s^n + A_{n-1} s^{n-1} + \ldots\ldots + A_1 s + A_0$ no sistema de base r.

- Se $s < r$ a expressão é avaliada diretamente.

- Se $s > r$ é necessário primeiramente converter a base s e os dígitos A_i para a base r.

Sistemas Numéricos
Conversão entre Bases

- Exemplos:

1) $s = 2$, $N_s = 1110101$, $r = 10$, $N_r = ?$

$N_r = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 = 117_{10}$

cederj

Sistemas Numéricos
Conversão entre Bases

- Exemplos:

1) $s = 2$, $N_s = 1110101$, $r = 10$, $N_r = ?$

$N_r = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 = 117_{10}$

2) $s = 10$, $N_s = 117$, $r = 2$, $N_r = ?$ (caso de s > r)

$N_r = 1 \times (10_2)^2 + 1 \times (10_2)^1 + (7_2)$

$N_r = 1010^2 + 1010 + 111 = 1100100 + 10001$

$N_r = 1110101_2$

Conversão entre Bases

- Cálculos envolvidos no Exemplo 2:

```
     1010
   x 1010
   ------
     0000
    1010
   0000
 + 1010
 --------
  1100100
```

```
   1100100
      1010
 +     111
 ---------
   1110101
```

Conversão entre Bases

- Exemplos (cont.):

3) $s = 9$, $N_s = 857$, $r = 10$, $N_r = ?$

$N_r = 8 \times 9^2 + 5 \times 9 + 7 = 700_{10}$

Conversão entre Bases

- Exemplos:

3) $s = 9$, $N_s = 857$, $r = 10$, $N_r = ?$

$N_r = 8 \times 9^2 + 5 \times 9 + 7 = 700_{10}$

4) $s = 16$, $N_s = BF7$, $r = 10$, $N_r = ?$ (caso de $s > r$)

$N_r = (B_{10}) \times 16^2 + (F_{10}) \times 16 + 7$

$N_r = 11 \times 256 + 15 \times 16 + 7 = 3063_{10}$

Conversão entre Bases

- Exemplos:

3) $s = 9, \; N_s = 857, \; r = 10, \; N_r = ?$

   $N_r = 8 \times 9^2 + 5 \times 9 + 7 = 700_{10}$

4) $s = 16, \; N_s = BF7, \; r = 10, \; N_r = ?$

   $N_r = (B_{10}) \times 16^2 + (F_{10}) \times 16 + 7$

   $N_r = 11 \times 256 + 15 \times 16 + 7 = 3063_{10}$

5) $s = 7, \; N_s = 100, \; r = 2, \; N_r = ?$

   No caso, nem a base origem nem a destino é a 10.

   A solução é transformar primeiramente 100_7 para a base10 e depois desta para a base 2 pelo método das divisões sucessivas a seguir.

Sistemas Numéricos

Conversão entre Bases

- Método das divisões sucessivas:

  $N_s = A_n A_{n-1} \ldots\ldots A_1 A_0$ (nº na base origem)

  $N_r = B_m B_{m-1} \ldots\ldots B_1 B_0$ (nº na base destino)

- Consiste em dividir sucessivamente o número dado N_s da base origem <u>s</u> pela base destino <u>r</u>.

  - Se s > r os restos B obtidos já são os dígitos procurados, ou seja, $N_r = B_m B_{m-1} \ldots\ldots B_1 B_0$..
  - Se s < r os restos B devem ser primeiramente convertidos para a base r.

cederj

Sistemas Numéricos
Conversão entre Bases

- Divisões sucessivas:

$$
\begin{array}{lllll}
N_s & \underline{|\ r\ } & & & \\
B_0 & N_1 & \underline{|\ r\ } & & \\
 & B_1 & N_2 & & \\
 & & \vdots & & \\
 & & N_{m-1} & \underline{|\ r\ } & \\
 & & B_{m-1} & N_m & \underline{|\ r\ } \\
 & & & B_m & 0
\end{array}
$$

$(B_m = N_m)$

Sistemas Numéricos
Conversão entre Bases

- Exemplos:

6) $s = 10$, $N_s = 70$, $r = 4$, $N_r = ?$

$$
\begin{array}{lllll}
70 & \multicolumn{1}{|l}{4} & & & \\
\cline{2-2}
2 & 17 & \multicolumn{1}{|l}{4} & & \\
\cline{3-3}
 & 1 & 4 & \multicolumn{1}{|l}{4} & \\
\cline{4-4}
 & & 0 & 1 & \multicolumn{1}{|l}{4} \\
\cline{5-5}
 & & & 1 & 0
\end{array}
$$

$$70_{10} = 1012_4$$

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont):

7) $s = 10,\ N_s = 70,\ r = 6,\ N_r = ?$

$$\begin{array}{llll}
70 & \multicolumn{1}{|l}{6} & & \\ \cline{2-2}
4 & 11 & \multicolumn{1}{|l}{6} & \\ \cline{3-3}
 & 5 & 1 & \multicolumn{1}{|l}{6} \\ \cline{4-4}
 & & 1 & 0
\end{array}$$

$70_{10} = 154_6$

Sistemas Numéricos
Conversão entre Bases

- Algoritmo da divisão sucessiva:

$$
\begin{array}{llllll}
N_s & \underline{|\ r} & & & & \\
B_0 & N_1 & \underline{|\ r} & & & \\
 & B_1 & N_2 & & & \\
 & & \vdots & & & \\
 & & N_{m-1} & \underline{|\ r} & & \\
 & & B_{m-1} & N_m & \underline{|\ r} & \\
 & & & B_m & 0 &
\end{array}
$$

$(B_m = N_m)$

Sistemas Numéricos
Conversão entre Bases

- Demonstração:

$$N_s = N_1 \,.\, r + B_0$$

$$N_1 = N_2 \,.\, r + B_1$$

$$N_m = 0 \,.\, r + B_m$$

$$N_s = (N_2 \,.\, r + B_1)\, r + B_0 = N_2 \,.\, r^2 + B_1 \,.\, r + B_0$$

$$N_s = (N_3 \,.\, r + B_2)\, r^2 + B_1 \,.\, r + B_0 \quad \ldots\ldots\ldots$$

$$N_s = B_m \,.\, r^m + B_{m-1} \,.\, r^{m-1} + \ldots\ldots + B_1 \,.\, r + B_0 = N_r$$

cederj

Sistemas Numéricos

Conversão entre Bases

- Exemplos (cont.):

8) s = 10, N_s = 69, r = 2, N_r = ?

$$
\begin{array}{llllllll}
69 & \underline{|\,2} & & & & & & \\
1 & 34 & \underline{|\,2} & & & & & \\
 & 0 & 17 & \underline{|\,2} & & & & \\
 & & 1 & 8 & \underline{|\,2} & & & \\
 & & & 0 & 4 & \underline{|\,2} & & \\
 & & & & 0 & 2 & \underline{|\,2} & \\
 & & & & & 0 & 1 & \underline{|\,2} \\
 & & & & & & 1 & 0
\end{array}
$$

$69_{10} = 1000101_2$

Sistemas Numéricos

Conversão entre Bases

- Exemplos (cont.):

9) $s = 2$, $N_s = 1000101$, $r = 10$, $N_r = ?$ (caso $s < r$)

A base destino r não tem representaçao direta na base s origem. É preciso antes achar representação da base destino na de origem para depois realizar a conversão.

$$r = 10_{10} = 1010_2$$

Sistemas Numéricos

Conversão entre Bases

- Exemplos (cont.):

9)

```
1000101 | 1010
- 1010    110 | 1010
  1110          0
- 1010
  1001
              110
```

$Nr = (110)_2\ (1001)_2 = 69_{10}$

Conversão entre Bases

- Apenas relembrando a sequência inicial da base 2:

- Base 2:

0,1,10,11,100,101,110,111,1000,1001,1010,1011,...

$\downarrow$

0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...

Conversão entre Bases

- Exemplos (cont.):

9a) $s = 2$, $N_s = 1000101$, $r = 10$, $N_r = ?$

A solucão do exemplo 4, em que $s < r$, é muito mais facilmente obtida através da decomposição posicional do número fornecido.

$$N_r = 1 \times 2^6 + 1 \times 2^2 + 1 = 64 + 4 + 1 = 69_{10}$$

Sistemas Numéricos
Conversão entre Bases

- Considerando s (base origem) e r (base destino) temos na prática que:

  - Quando $s < r$ e $r = 10$ ⟶ aplicar o método do desenvolvimento da notação posicional do número N_s
  - Quando $s > r$ e $s = 10$ ⟶ aplicar o método das divisões sucessivas.
  - Quando $s \neq 10$ e $r \neq 10$ ⟶ converter N_s para a base 10 (desenvolvimento posicional) e depois converter para a base r (divisões sucessivas).

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont):

10) $s = 7$, $N_s = 100$, $r = 2$, $N_r = ?$ (Ex5 não resolvido)

$N_s = 1 \times 7^2 = 49_{10}$

$$
\begin{array}{llllll}
49 & \underline{|\,2} & & & & \\
1 & 24 & \underline{|\,2} & & & \\
 & 0 & 12 & \underline{|\,2} & & \\
 & & 0 & 6 & \underline{|\,2} & \\
 & & & 0 & 3 & \underline{|\,2} \\
 & & & & 1 & 1 \;\; \underline{|\,2} \\
 & & & & & 1 \;\; 0
\end{array}
$$

$100_7 = 110001_2$

Conversão entre Bases

- Exemplos (cont):

11) $s = 9,\ N_s = 87,\ r = 4,\ N_r = ?$

$N_s = 8 \times 9 + 7 = 79_{10}$

$$
\begin{array}{lllll}
79 & \underline{|\,4} & & & \\
3 & 19 & \underline{|\,4} & & \\
 & 3 & 4 & \underline{|\,4} & \\
 & & 0 & 1 & \underline{|\,4} \\
 & & & 1 & 0
\end{array}
$$

$87_9 = 1033_4$

Conversão entre Bases

- Exemplos (cont):

12) $s = 3,\ N_s = 2120,\ r = 9,\ N_r = ?$

$N_s = 2 \times 3^3 + 3^2 + 2 \times 3 = 54 + 9 + 6 = 69_{10}$

$$\begin{array}{r|l}
69 & 9 \\ \cline{2-2}
6 & 7 \;\; \begin{array}{|l} 9 \\ \hline \end{array} \\
 & 7 \;\;\;\; 0
\end{array}$$

$2120_3 = 76_9$

Sistemas Numéricos

Conversão entre Bases

- Exemplos (cont):

11) $s = 9, N_s = 76, r = 3, N_r = ?$

$N_s = 7 \times 9 + 6 = 63 + 6 = 69_{10}$

$$
\begin{array}{lllll}
69 & \underline{|\,3} & & & \\
0 & 23 & \underline{|\,3} & & \\
 & 2 & 7 & \underline{|\,3} & \\
 & & 1 & 2 & \underline{|\,3} \\
 & & & 2 & 0
\end{array}
$$

$76_9 = 2120_3$

cederj

Sistemas Numéricos

Conversão entre Bases

- Os dois exemplos anteriores podem ser resolvidos de forma mais rápida se atentarmos para o fato de que a base 9 é potência da base 3. Senão, vejamos a sequência inicial da base 3:

0,1,2,10,11,12, 20,21,22,100,101,102,110…(base 3)

0,1,2, 3, 4, 5, 6, 7, 8, 10 , 11, 12, 13….(base 9)

Sistemas Numéricos

Conversão entre Bases

- Repare que cada dois dígitos em sequência de um dado número na base 3 corresponde diretamente a um dígito da base 9. Assim teríamos:

$$21\ 20_3 = (2 \times 3) + 1 \quad (2 \times 3) = 7\ 6_9$$

- É esta relação de potências entre as bases que tornará, com o passar do tempo, a conversão entre as bases 2, 4, 8, e principalmente entre a 2 e a 16 (e vice-versa), muito mais simples.

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont.):

12) $s = 2, \; N_s = 1011011, \; r = 4, \; N_r = ?$

$01 \; 01 \; 10 \; 11 = 1 \; 1 \; 2 \; 3 = 1123_4$

cederj

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont.):

12) $s = 2,\ N_s = 1011011,\ r = 4,\ N_r = ?$

$$01\ 01\ 10\ 11 = 1\ 1\ 2\ 3 = 1123_4$$

13) $s = 4,\ N_s = 1123,\ r = 2,\ N_r = ?$

$$1\ 1\ 2\ 3 = 01\ 01\ 10\ 11 = 1011011_2$$

cederj

Sistemas Numéricos

Conversão entre Bases

- Exemplos (cont.):

14) $s = 2,\ N_s = 1011011_2,\ r = 8,\ N_r = ?$

$001\ 011\ 011 = 1\ 3\ 3 = 133_8$

15) $s = 2,\ N_s = 1011011,\ r = 16,\ N_r = ?$

$0101\ 1011 = 5\ B = 5B_{16}$

cederj

Sistemas Numéricos
Conversão entre Bases

- Mudança de base de números reais:

$$N_R = (A_n A_{n-1} \ldots A_2 A_1 A_0 \,.\, A_{-1} A_{-2} A_{-3} \ldots)$$

$$N_R = N_I + N_F \text{, onde:}$$

$$N_I = A_n.b^n + A_{n-1}.b^{n-1} + \ldots\ldots + A_1.b^1 + A_0.b^0$$

$$\boxed{N_F = A_{-1}.b^{-1} + A_{-2}.b^{-2} + A_{-3}.b^{-3} + \ldots\ldots}$$

Sistemas Numéricos
Conversão entre Bases

- A parte fracionária N_F de um número em uma dada base corresponde sempre `a parte fracionária de sua representação em outra base.

$$N_F = A_{-1}.b^{-1} + A_{-2}.b^{-2} + A_{-3}.b^{-3} + \ldots\ldots$$

- O problema se reduz ao cálculo dos dígitos A_{-1} , A_{-2} , A_{-3} , ...

Sistemas Numéricos

Conversão entre Bases

- Multiplicando a expressão de N_F por b temos:

$$b.N_F = (A_{-1}.b^{-1} + A_{-2}.b^{-2} + A_{-3}.b^{-3} + \ ......).b$$

$$b.N_F = A_{-1} + A_{-2}.b^{-1} + A_{-3}.b^{-3} + \ ......$$

Conversão entre Bases

- Repare que na expressão anterior A_{-1} é a parte inteira de $b.N_F$. Subtraindo-se A_{-1} de $b.N_F$ e multiplicando novamente por b a expressão resultante temos:

$$b.(b.N_F - A_{-1}) = A_{-2} + A_{-3}.b^{-1} + \ldots\ldots$$

- O processo deve continuar até alcançar o número de dígitos desejado na parte fracionária. A aplicação do método na prática é bem mais simples que as equações anteriores parecem sugerir.

Sistemas Numéricos

Conversão entre Bases

- Exemplos:

1) Converter 69.71_{10} para a base 2.

N_I = 1000101 de um exemplo anterior.

Para N_F:

$2 \times (0.71) = 1.42 \longrightarrow A\text{-}1 = 1$

$2 \times (0.42) = 0.84 \longrightarrow A\text{-}2 = 0$

$2 \times (0.84) = 1.68 \longrightarrow A\text{-}3 = 1$

$2 \times (0.68) = 1.36 \longrightarrow A\text{-}4 = 1$

$2 \times (0.36) = 0.72 \longrightarrow A\text{-}5 = 0$

$2 \times (0.72) = 1.44 \longrightarrow A\text{-}6 = 1$

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont.):

1) Continuando …N_F = 101101…… e o resultado fica:

$69.71_{10} = (1000101.101101\ldots\ldots)_2$

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont.):

1) Continuando ...N_F = 101101...... e o resultado fica:
$69.71_{10} = (1000101.101101......)_2$

2) Converter $(1000101.101101)_2$ para a base 10.
Aqui como s < r, a solução é desenvolver a notação posicional do número:
$1000101.101101 = 2^6 + 2^2 + 1 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} =$
$= 64 + 4 + 1 + 0.5 + 0.125 + 0.0625 + 0.015625 =$
$= 69.703125_{10}$

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont.):

3) $s = 2$, $N_s = 1001011.0110011$, $r = 4$, $N_r = ?$

$01\ 00\ 10\ 11\ .\ 01\ 10\ 01\ 10 = 1\ 0\ 2\ 3\ .\ 1\ 2\ 1\ 2$

$1001011.0110011_2 = 1023.1212_4$

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont.):

3) $s = 2$, $N_s = 1001011.0110011$, $r = 4$, $N_r = ?$

01 00 10 11 . 01 10 01 10 = 1 0 2 3 . 1 2 1 2

$1001011.0110011_2 = 1023.1212_4$

4) $s = 2$, $N_s = 10101110.10011111$, $r = 8$, $N_r = ?$

010 101 110 . 100 111 110 = 2 5 6 . 4 7 6

$10101110.100111110_2 = 256.476_8$

cederj

Sistemas Numéricos

Conversão entre Bases

- Exemplos (cont.):

5) $s = 2$, $N_s = 101011.1100111$, $r = 16$, $N_r = ?$

$$0010 \; 1011 \; . \; 1100 \; 1110 = 2 \; B \; . \; C \; E$$

$$101011.1100111_2 = 2B.CE_{16}$$

cederj

Sistemas Numéricos
Conversão entre Bases

- Exemplos (cont.):

5) $s = 2$, $N_s = 101011.1100111$, $r = 16$, $N_r = ?$

$0010\ 1011\ .\ 1100\ 1110 = 2\ B\ .\ C\ E$

$101011.1100111_2 = 2B.CE_{16}$

6) $s = 16$, $N_s = AB.CD$, $r = 8$, $N_r = ?$

$N_s = \underline{A}\ \underline{B}\ .\ \underline{C}\ \underline{D} = (\underline{1010}\ \ \underline{1011}\ \ .\ \underline{1100}\ \ \underline{1101})_2$

$N_s = (\underline{010}\ \ \underline{101}\ \ \underline{011}\ \ .\ \underline{110}\ \ \underline{011}\ \ \underline{010})_2$

$N_s = (\underline{2}\ \underline{5}\ \underline{3}\ .\ \underline{6}\ \underline{3}\ \underline{2})_8$

$AB.CD_{16} = 253.632_8$

cederj

Introdução à Informática

Sistemas Numéricos

Operações Aritméticas em Diversas Bases

Ageu Pacheco e Alexandre Meslin

cederj

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Objetivo da Aula:

- Partindo da base 10, ver como operações aritméticas são efetuadas em outras bases; em especial a 2.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 10:

$$\begin{array}{cccc}
 & (1) & (1) & \\
 & & 6 & 7_{10} \\
+ & & 8 & 4_{10} \\
\hline
 & 1 & 5 & 1_{10}
\end{array}$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 10:

$$
\begin{array}{rccc}
 & (1) & (1) & \\
 & & 6 & 7_{10} \\
+ & & 8 & 4_{10} \\
\hline
 & 1 & 5 & 1_{10}
\end{array}
\qquad\qquad
\begin{array}{rccc}
 & (1) & (1) & \\
 & & 6 & 7_{10} \\
+ & & 8 & 4_{10} \\
\hline
 & 1 & (15) & (11) \\
- & & 10 & 10 \\
\hline
 & 1 & 5 & 1_{10}
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 9:

$$
\begin{array}{cccc}
 & (1) & (1) & \\
 & & 6 & 7_{9} \\
+ & & 8 & 4_{9} \\
\hline
 & 1 & (15) & (11) \\
- & & 9 & 9 \\
\hline
 & 1 & 6 & 2_{9}
\end{array}
$$

cederj

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 9:

$$
\begin{array}{rccc}
 & (1) & (1) & \\
 & & 6 & 7_9 \\
+ & & 8 & 4_9 \\
\hline
 & 1 & (15) & (11) \\
- & & 9 & 9 \\
\hline
 & 1 & 6 & 2_9
\end{array}
$$

- Na base 8:

$$
\begin{array}{rccc}
 & (1) & (1) & \\
 & & 7 & 7_8 \\
+ & & 7 & 7_8 \\
\hline
 & 1 & (15) & (14) \\
- & & 8 & 8 \\
\hline
 & 1 & 7 & 6_8
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 7:

$$
\begin{array}{cccc}
 & (1) & (1) & \\
 & & 2 & 5_7 \\
+ & & 5 & 4_7 \\
\hline
 & 1 & (8) & (9) \\
- & & 7 & 7 \\
\hline
 & 1 & 1 & 2_7
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 7:

$$
\begin{array}{cccc}
 & (1) & (1) & \\
 & & 2 & 5_7 \\
+ & & 5 & 4_7 \\
\hline
 & 1 & (8) & (9) \\
- & & 7 & 7 \\
\hline
 & 1 & 1 & 2_7
\end{array}
$$

- Na base 16:

$$
\begin{array}{ccccc}
 & (1) & & (1) & \\
 & & F & 1 & A_{16} \\
+ & & E & 0 & 9_{16} \\
\hline
 & 1 & (29) & 2 & (19) \\
- & & 16 & & 16 \\
\hline
 & 1 & D & 2 & 3_{16}
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 16:

	(1)	(1)	
	2	F	F_{16}
+	1	F	E_{16}
	4	(31)	(29)
-		16	16
	4	F	D_{16}

cederj

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 16:

$$
\begin{array}{cccc}
 & (1) & (1) & \\
 & 2 & F & F_{16} \\
+ & 1 & F & E_{16} \\
\hline
 & 4 & (31) & (29) \\
- & & 16 & 16 \\
\hline
 & 4 & F & D_{16}
\end{array}
$$

- Na base 2:

$$
\begin{array}{ccccccc}
 & (1) & (1) & (1) & (1) & (1) & \\
 & & 1 & 0 & 1 & 1 & 1_2 \\
+ & & 1 & 1 & 0 & 1 & 1_2 \\
\hline
 & & (3) & (2) & (2) & (3) & (2) \\
- & & 2 & 2 & 2 & 2 & 2 \\
\hline
 & 1 & 1 & 0 & 0 & 1 & 0_2
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 2 (cont.):

$$
\begin{array}{rcccccc}
 & (1) & (1) & (1) & & & \\
 & & 1 & 0 & 1 & 0 & 0_2 \\
+ & & 1 & 1 & 1 & 0 & 1_2 \\
\hline
 & 1 & 1 & 0 & 0 & 0 & 1_2
\end{array}
$$

cederj

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Adição na base 2 (cont.):

$$
\begin{array}{ccccccc}
 & (1) & (1) & (1) & & & \\
 & & 1 & 0 & 1 & 0 & 0_2 \\
+ & & 1 & 1 & 1 & 0 & 1_2 \\
\hline
 & 1 & 1 & 0 & 0 & 0 & 1_2
\end{array}
$$

$$
\begin{array}{cccccccc}
 & & (10) & (1) & (1) & (1) & (1) & \\
 & & & 1 & 1 & 0 & 1 & 1_2 \\
 & & & 1 & 0 & 1 & 0 & 1_2 \\
+ & & & 1 & 1 & 0 & 1 & 0_2 \\
\hline
 & 1 & 0 & 0 & 1 & 0 & 1 & 0_2
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 10:

$$
\begin{array}{rcc}
 & 4 & 7_{10} \\
- & 2 & 4_{10} \\
\hline
 & 2 & 3_{10}
\end{array}
\qquad\qquad
\begin{array}{rcc}
 & (4) & (+10) \\
 & \cancel{5} & 2_{10} \\
- & 1 & 7_{10} \\
\hline
 & 3 & 5_{10}
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 10 (cont.):

$$\begin{array}{rcc} & 5 & 2_{10} \\ - & 1_{(+1)} & 7_{10} \\ \hline & 3 & 5_{10} \end{array} \qquad\qquad \begin{array}{rccc} & 3 & 0 & 0_{10} \\ - & 1_{(+1)} & 4_{(+1)} & 7_{10} \\ \hline & 1 & 5 & 3_{10} \end{array}$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 8:

$$
\begin{array}{rcc}
 & 5 & 2_8 \\
- & 1_{(+1)} & 7_8 \\
\hline
 & 3 & 3_8
\end{array}
$$

- Subtração na base 8 (cont.):

$$
\begin{array}{rcc}
 & 5 & 2_8 \\
- & 1_{(+1)} & 7_8 \\
\hline
 & 3 & 3_8
\end{array}
\qquad\qquad
\begin{array}{rccc}
 & 3 & 0 & 0_8 \\
- & 1_{(+1)} & 4_{(+1)} & 7_8 \\
\hline
 & 1 & 3 & 1_8
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 16:

$$\begin{array}{rcc} & 5 & 2_{16} \\ - & 1\,(+1) & 7_{16} \\ \hline & 3 & B_{16} \end{array}$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 16 (cont.):

$$
\begin{array}{rcc}
 & 5 & 2_{16} \\
- & 1_{(+1)} & 7_{16} \\
\hline
 & 3 & B_{16}
\end{array}
\qquad\qquad
\begin{array}{rccc}
 & 3 & 0 & 0_{16} \\
- & 1_{(+1)} & 4_{(+1)} & 7_{16} \\
\hline
 & 1 & B & 9_{16}
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 16 (cont):

$$
\begin{array}{cccc}
 & 3 & D & C_{16} \\
- & 1_{(+1)} & F_{(+1)} & E_{16} \\
\hline
 & 1 & (13) & (14) \\
 & \downarrow & \downarrow & \downarrow \\
 & 1 & D & E_{16}
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 16 (cont):

$$
\begin{array}{cccc}
 & 3 & D & C_{16} \\
- & 1_{(+1)} & F_{(+1)} & E_{16} \\
\hline
 & 1 & (13) & (14) \\
 & \downarrow & \downarrow & \downarrow \\
 & 1 & D & E_{16}
\end{array}
\qquad\qquad
\begin{array}{cccc}
 & C & 0 & A_{16} \\
- & 2_{(+1)} & C_{(+1)} & C_{16} \\
\hline
 & 9 & 3 & (14) \\
 & \downarrow & \downarrow & \downarrow \\
 & 9 & 3 & E_{16}
\end{array}
$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtração na base 2:

$$\begin{array}{rcccccc} & 1 & 1 & 0 & 0 & 1 & 0_2 \\ - & 1 & 0 & 1 & 1 & 0 & 1_2 \\ \hline & 0 & 0 & 0 & 1 & 0 & 1_2 \end{array}$$

$$\begin{array}{rcccccccc} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1_2 \\ - & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0_2 \\ \hline & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1_2 \end{array}$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Representação em "complemento à base"

  - O complemento de um número **N** em uma dada base **B** é igual a diferença entre o número e a próxima potência da base.

Ex: Complemento a 10 de 734

próxima potência $\longrightarrow 10^3$

$$\begin{array}{r} 1000 \\ -\ \ 734 \\ \hline 266 \end{array}$$

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

Ex: Complemento a 2 de 1011

próxima potência $\longrightarrow$ $2^4 = 16 = 10000$

$$\begin{array}{rr} & 10000 \\ - & 1011 \\ \hline & 101 \end{array}$$

101 $\longrightarrow$ é o complemento a 2 de 1011

Ex: Complemento a 2 de 101101

próxima potência $\longrightarrow$ 1000000

$$
\begin{array}{r}
1000000 \\
-\quad 101101 \\
\hline
10011
\end{array}
$$

10011 $\longrightarrow$ é o comp. a 2 de 101101

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Representação em “complemento à base” (cont.)

  - O cálculo do complemento à base em qualquer base é tedioso por causa dos “vem-um”.

  - Uma alternativa mais confortável é calcular o “complemento à base menos 1” e depois somar 1 ao resultado para obter o complemento à base.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

Ex: Base 10 ⟶ "complemento a 9" de 734

$$
\begin{array}{r l}
 & 9\ 9\ 9 \\
- & 7\ 3\ 4 \\
\hline
 & 2\ 6\ 5 \\
+ & 1 \\
\hline
 & 2\ 6\ 6
\end{array}
$$

2 6 5 ⟶ (complemento a 9 de 734)

2 6 6 ⟶ (complemento a 10 de 734)

cederj

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

Ex: Base 2 → "complemento a 1" de 1011

$$
\begin{array}{rr}
 & 1\,1\,1\,1 \\
- & 1\,0\,1\,1 \\
\hline
 & 1\,0\,0 \\
+ & 1 \\
\hline
 & 1\,0\,1
\end{array}
$$

1 0 0 → (complemento a 1 de 1011)

1 0 1 → (complemento a 2 de 1011)

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Observação importante:

  Para obter o “complemento a 1” de um número binário basta invertê-lo bit a bit.

  Ex: complemento a 1 de 1 0 1 1

  0 1 0 0

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Operações de subtração em qualquer base podem ser feitas utilizando complemento à base.

Ex: Subtração 913 – 734 $\longrightarrow$ na base 10

913	999	913
- 734	- 734	+ 266
179	265	1)179
	+ 1	
	266	

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

Ex: Subtração 11001 – 10011 na base 2

complemento a 1 do subtraendo = 01100

comp. a 2:

01100 → (comp.a 1)		11001 (minuendo)
+ 1		+ 01101
01101 → (comp.a 2)		1)00110 (resultado)

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Regra prática para obtenção do complemento a 2:

  Para se obter diretamente o complemento a 2 de um número basta percorrer o número da direita para a esquerda repetindo-se os dígitos zeros até encontrar o primeiro dígito 1 (um), o qual deve ser mantido. A partir daí, todos os dígitos (zeros ou uns) a esquerda desse primeiro 1 deverão ser invertidos.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

Exemplos:

1) 1010 ⇒ 0110 (comp. a 2)
2) 11001 ⇒ 00111 (comp. a 2)
3) 111000 ⇒ 001000 (comp. a 2)
4) 1100110 ⇒ 0011010 (comp. a 2)

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Representação de números com sinal:

  Para simplificar, vamos supor que os números sejam representados internamente ao computador no formato de 8 bits (1 byte), ou seja eles são operados e armazenados em 8 bits.

7	6	5	4	3	2	1	0
MSB							**LSB**

Bit de sinal
- **0 – nº positivo**
- **1 – nº negativo**

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Representação de números com sinal (cont.):

  Em 8 bits é possível representar 256 números diferentes : de 00000000 a 11111111, já que $2^8 = 256$.

  Com o bit mais significativo representando o sinal, a gama de números possíveis de serem representados permanece a mesma, só que agora metade negativa e metade positiva.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Exemplos:

+ 127	0	1	1	1	1	1	1	1
+ 1	0	0	0	0	0	0	0	1
(+) 0	0	0	0	0	0	0	0	0
- 1	1	1	1	1	1	1	1	1
- 128	1	0	0	0	0	0	0	0

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Mais exemplos:

+ 74	⇒	0	1	0	0	1	0	1	0
+ 27	⇒	0	0	0	1	1	0	1	1
- 10	⇒	1	1	1	1	0	1	1	0
- 27	⇒	1	1	1	0	0	1	0	1
- 74	⇒	1	0	1	1	0	1	1	0

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Operações aritméticas no computador:

Números positivos são submetidos na forma "normal".

Ex: + 22 ⇒

0	0	0	1	0	1	1	0

Números negativos na forma complemento a 2.

Ex: - 22 ⇒

1	1	1	0	1	0	1	0

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Desta forma, todas as operações de soma e subtração envolvendo números com bits de sinal produzem diretamente resultados consistentes, ou seja, positivos na forma "normal" e negativos em complemento a 2.

- A vantagem de representar números negativos em complemento a 2 internamente ao computador é que todas as operações de subtração ficam transformadas em simples somas.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Exemplos:

  Tomemos como exemplo os números 9 e 4. Vamos ver como ficam todas as possibilidades de somas e subtrações envolvendo suas formas positivas e negativas.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Somas:

+ 4 00000100 (+) + 9 00001001 +13 00001101	+4 00000100 (+) - 9 11110111 - 5 11111011
- 4 11111100 (+) + 9 00001001 + 5 1)00000101	- 4 11111100 (+) - 9 11110111 - 13 1)11110011

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtrações:

	+ 4	00000100		00000100
(-)	+ 9	00001001	→	11110111 (+)
	- 5			11111011
	+ 4	00000100		00000100
(-)	- 9	11110111	→	00001001 (+)
	+13			00001101

cederj

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Subtrações:

- 4	11111100		11111100
(-) + 9	00001001	→	11110111 (+)
-13			1) 11110011

- 4	11111100		11111100
(-) - 9	11110111	→	00001001 (+)
+ 5			1) 00000101

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- "Overflow" em operações aritméticas

  - O maior número positivo que pode ser carregado (armazenado) em um registro de 8 bits é +127 (01111111).

  - Nas mesmas condições o menor número negativo é –128 (10000000).

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- “Overflow” em operações aritméticas (cont.)

  - Apenas para ilustrar, supondo ser o computador de 8 bits, temos que, caso o resultado de qualquer operação aritmética exceda um dos valores anteriores, é dito que uma condição de overflow ocorreu, o que normalmente acarreta erro.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- “Overflow” em operações aritméticas (cont.)

  - A detecção de overflow é simples e consiste em:

    1. Há um “vai-um” propagado para o bit de sinal sem “vai-um” saindo deste.
    2. Há um “vai-um” propagado pelo bit de sinal sem este ter recebido “vai-um”.

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Exemplos:

```
1)      + 88            0 1011000
    (+) + 46        (+) 0 0101110
    --------        -------------
       + 134            1 0000110
```

overflow

Sistemas Numéricos
Operações Aritméticas em Diversas Bases

- Exemplos (cont):

1) + 88   0 1011000

(+) + 46   (+) 0 0101110

+ 134   1 0000110

overflow

2) - 76   (01001100) $\longrightarrow$ 1 0110100

(-) + 68   (01000100) $\longrightarrow$ 1 0111100

- 144   1 0 1110000

overflow

Introdução à Informática

Álgebra de Boole

Ageu Pacheco e Alexandre Meslin

cederj

Álgebra de Boole

- Objetivo da Aula:

- Estudar os conceitos e regras que regem o projeto e funcionamento dos circuitos lógicos dos computadores digitais.

Álgebra de Boole

- Álgebra de Boole:

  Criada em 1854 por George Boole com o intuito de formalizar matematicamente o pensamento lógico.

Álgebra de Boole

- Variável lógica ou booleana:

  Uma variável lógica só pode assumir dois valores (estados):

  verdadeiro ou falso; ligado ou desligado; aceso ou apagado; fechado ou aberto; branco ou preto; sim ou não; “1” ou “0”.

Álgebra de Boole

- Operações lógicas básicas (primitivas):

  São 3 as operações lógicas básicas:

  1. Produto lógico ⇒ porta AND ($\cdot$)
  2. Soma lógica ⇒ porta OR (+)
  3. Negação ⇒ porta NOT ($-$)

cederj

Álgebra de Boole

- Todas as operações internas a um computador podem ser descritas por combinações destas 3 operações básicas.

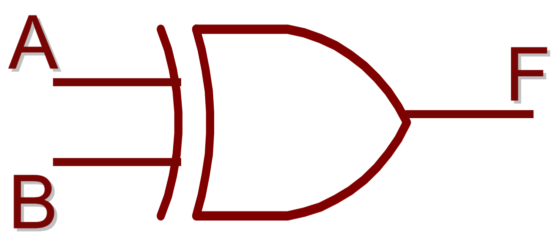
- Na realidade bastaria utilizar o AND com o NOT ou o OR com o NOT (conjuntos funcionalmente completos).

Álgebra de Boole

- Uma função lógica pode ser representada pela sua expressão algébrica, pela sua tabela verdade, pelo seu símbolo ou circuito lógico.

  Exemplo:

$$F(A,B) = \overline{A}.B + A.\overline{B}$$



A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

cederj

Álgebra de Boole

- Função AND

Definição:

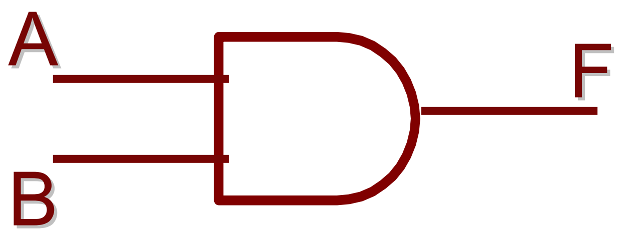
$$F(A,B,C,\ldots,N) = A.B.C\ldots.N$$

$$F(A,B,C,\ldots,N) = 1 \text{ se e somente se } A=B=C=\ldots=N=1$$

Álgebra de Boole

- AND de duas variáveis:

$$F(A,B) = A.B$$



Símbolo lógico

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

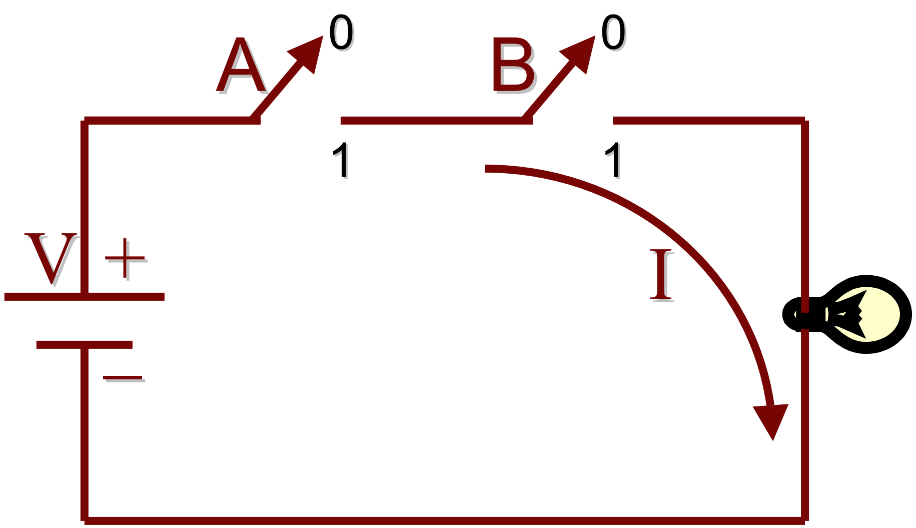
Tabela verdade

cederj

Álgebra de Boole

- AND de duas variáveis: (cont.)

$$F(A,B) = A.B$$



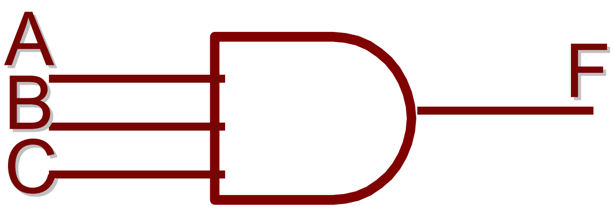
Circuito elétrico equivalente

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Álgebra de Boole

- AND de três variáveis:

$$F(A,B,C) = A.B.C$$



A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Álgebra de Boole

- Cada linha da tabela verdade relaciona uma combinação específica das variáveis de entrada ao valor assumido pela função na saída.

- O número total de linhas é igual a 2^n, onde n é o número de variáveis.

Álgebra de Boole

- Função OR

Definição:

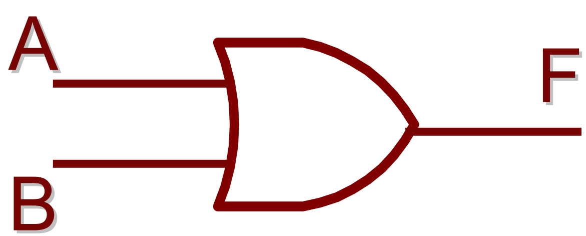
$$F(A,B,C,...,N) = A+B+C+...+N$$

$$F(A,B,C,...,N) = 1 \text{ se e somente se } A=1 \text{ ou } B=1 \text{ ou } ... \text{ ou } N=1$$

Álgebra de Boole

- OR de duas variáveis:

$$F(A,B) = A+B$$



Símbolo lógico

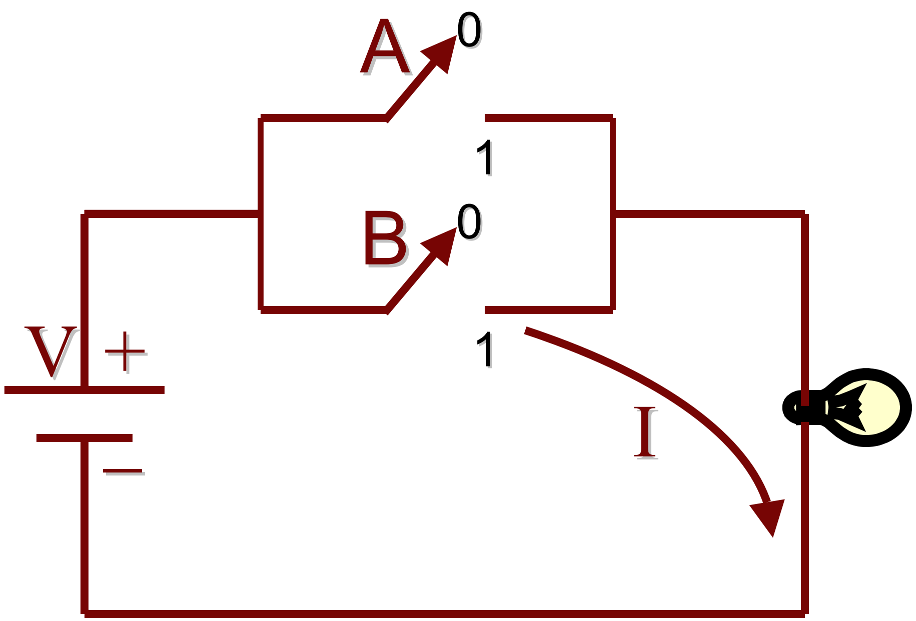
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Tabela verdade

Álgebra de Boole

- OR de duas variáveis: (cont.)

$$F(A,B) = A+B$$



Circuito elétrico equivalente

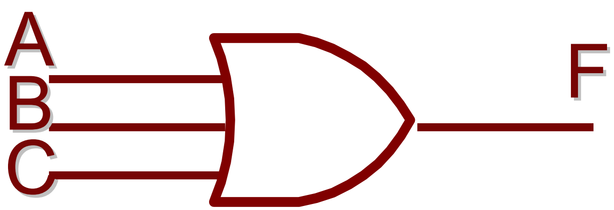
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

cederj

Álgebra de Boole

- OR de três variáveis:

$$F(A,B,C) = A+B+C$$



A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

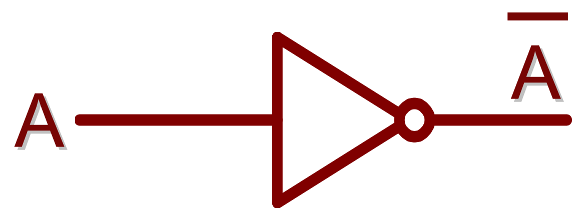
Álgebra de Boole

- Função NOT

Definição: $F(A) = \overline{A}$

$F(A) = 0$ se $A = 1$
$F(A) = 1$ se $A = 0$

A	F
0	1
1	0



Álgebra de Boole

- Outras funções lógicas importantes:
- Função NAND

Definição:

$$F(A,B,C,...,N) = \overline{A.B.C....N}$$

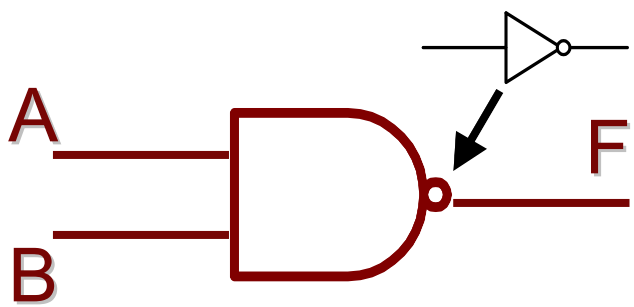
$F(A,B,C,...,N) = 0$ se e somente se

$A=B=C=...=N=1$

Álgebra de Boole

- NAND de duas variáveis:

$$F(A,B) = \overline{A.B}$$



Símbolo lógico

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Tabela verdade

cederj

Álgebra de Boole

- Função NOR

Definição:

$$F(A,B,C,...,N) = \overline{A+B+C+...+N}$$

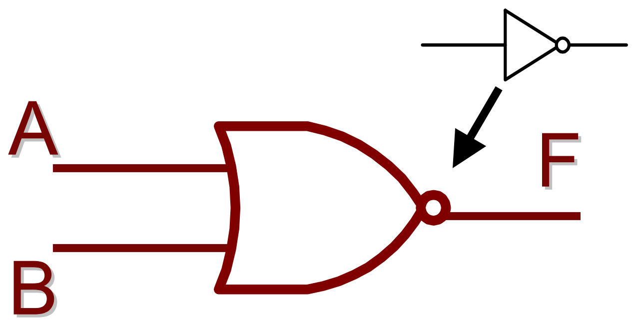
$F(A,B,C,...,N) = 1$ se e somente se $A=B=C=...=N=0$

cederj

Álgebra de Boole

- NOR de duas variáveis:

$$F(A,B) = \overline{A+B}$$



Símbolo lógico

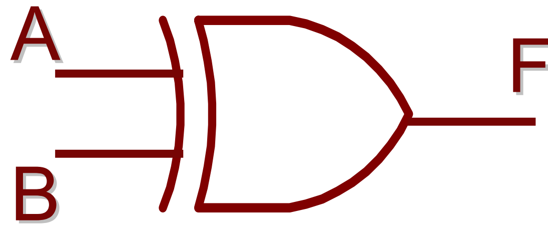
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Tabela verdade

Álgebra de Boole

- Função OU-EXCLUSIVO (XOR)

$F(A,B) = A \oplus B$



A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Por inspeção na tabela verdade:

$$\text{F=1 se } \underbrace{\text{A=0 e B=1}}_{\overline{A}.B} \text{ ou se } \underbrace{\text{A=1 e B=0}}_{A.\overline{B}}$$

$$F(A,B) = \overline{A}.B + A.\overline{B}$$

Álgebra de Boole

- Relações da álgebra booleana:
- Postulados:

1a. $A = 1$ (se $A \neq 0$)
2a. $0.0 = 0$
3a. $1.1 = 1$
4a. $1.0 = 0$
5a. $\overline{1} = 0$

1b. $A = 0$ (se $A \neq 1$)
2b. $0+0 = 0$
3b. $1+1 = 1$
4b. $1+0 = 1$
5b. $\overline{0} = 1$

Álgebra de Boole

- Relações da álgebra booleana (cont.):

- Teoremas:

6a.	$A.0 = 0$	6b.	$A+0 = A$
7a.	$A.1 = A$	7b.	$A+1 = 1$
8a.	$A.A = A$	8b.	$A+A = A$
9a.	$A.\overline{A} = 0$	9b.	$A+\overline{A} = 1$
10a.	$\overline{\overline{A}} = A$	10b.	$A = \overline{\overline{A}}$

Álgebra de Boole

- Propriedades algébricas:

  Comutativa:

  11a. $AB = BA$ 11b. $A+B = B+A$

  Associativa:

  12a. $A(BC) = AB(C)$

  12b. $A+(B+C) = (A+B)+C$

  Distributiva:

  13a. $A(B+C) = AB + AC$

  13b. $A + BC = (A+B)(A+C)$

Álgebra de Boole

- Teorema da absorção:

14a. $A(A+B) = A$ 14b. $A+AB = A$

15a. $A(\overline{A}+B) = AB$ 15b. $A+\overline{A}B = A+B$

cederj

Álgebra de Boole

- Teoremas de De Morgan:

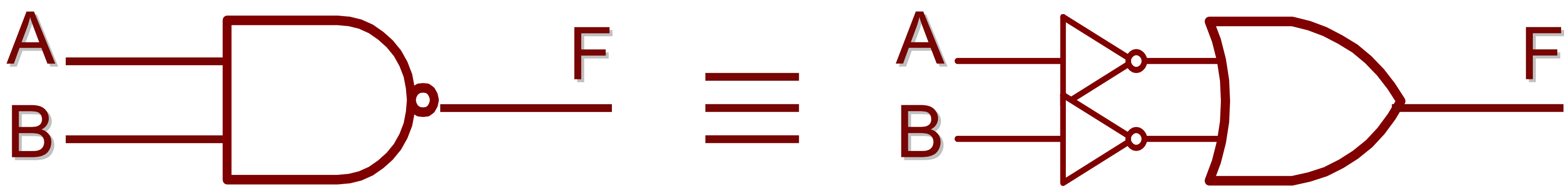
16a. $\overline{A.B.C...N} = \overline{A} + \overline{B} + \overline{C} +...+\overline{N}$

16b. $\overline{A+B+C+...+N} = \overline{A} . \overline{B} . \overline{C} ... \overline{N}$

Álgebra de Boole

- Consequências diretas das leis De Morgan:

1. $\overline{A.B} = \overline{A} + \overline{B}$



cederj

Álgebra de Boole

- Consequências diretas das leis De Morgan:

1. Prova pela tabela verdade:

$$F1(A,B) = \overline{A.B}$$

$$F2(A,B) = \overline{A} + \overline{B}$$

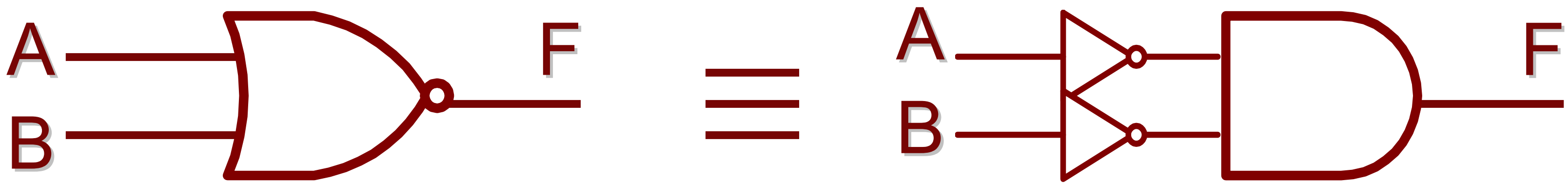
$$\overline{A.B} = \overline{A} + \overline{B}$$

A	B	F1	F2
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Álgebra de Boole

- Consequências diretas das leis De Morgan:

2. $\overline{A+B} = \overline{A} \cdot \overline{B}$

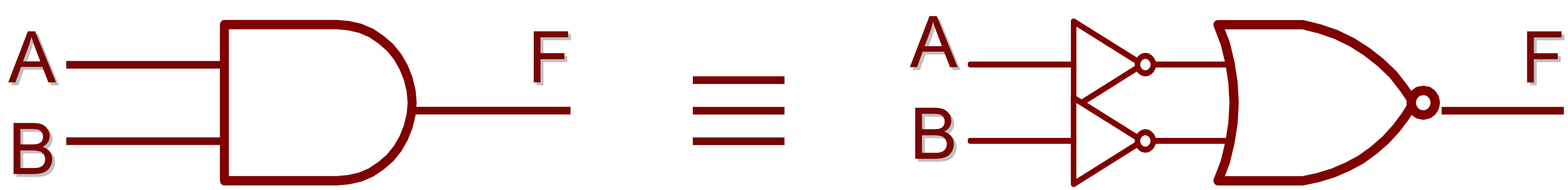


Álgebra de Boole

- Consequências diretas das leis De Morgan:

3. $\overline{A.B} = \overline{A}+\overline{B} \Rightarrow \overline{\overline{A.B}} = \overline{\overline{A} + \overline{B}} \Rightarrow$

$$A.B = \overline{\overline{A} + \overline{B}}$$

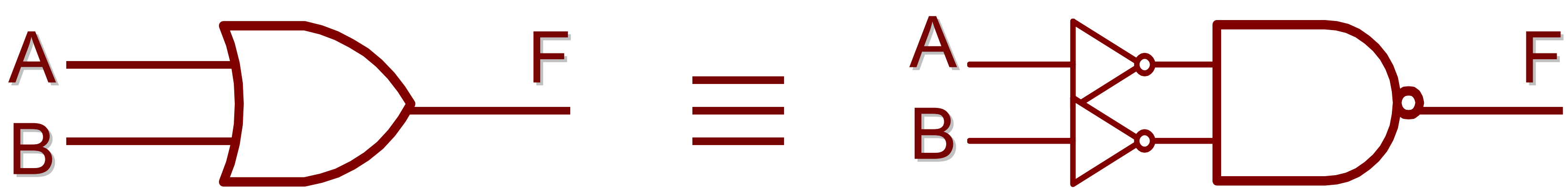


Álgebra de Boole

- Consequências diretas das leis De Morgan:

$$4.\quad \overline{A+B} = \overline{A}.\overline{B} \;\Rightarrow\; \overline{\overline{A+B}} = \overline{\overline{A}\,.\,\overline{B}} \;\Rightarrow$$

$$A+B = \overline{\overline{A}\,.\,\overline{B}}$$



Álgebra de Boole

- Exercícios:

1. Mostrar que $A + BC = (A+B)(A+C)$ 13b

$$
\begin{aligned}
(A+B)(A+C) &= AA+AC+AB+BC = \\
&= A+AC+AB+BC = \\
&= A(\cancelto{1}{1+C}+B)+BC \\
&= A + BC
\end{aligned}
$$

Álgebra de Boole

- Exercícios: (cont.)

2. Mostrar que $A + \overline{A}B = A + B \Rightarrow$ 15b

14b $\Rightarrow A+AB = A(1+B) = A$

$$A + \overline{A}B = \overbrace{A+AB}^{A} + \overline{A}B = A+(A+\overline{A})B =$$
$$= A + B$$

cederj

Álgebra de Boole

- Exercícios: (cont.)

3. Mostrar que $\overline{A \oplus B} = \bar{A} \oplus B$

$$\overline{A \oplus B} = \overline{\bar{A}B + A\bar{B}} \overset{M}{=} \overline{\bar{A}B} \,.\, \overline{A\bar{B}} \overset{M}{=}$$

$$\overset{M}{=} (A+\bar{B})(\bar{A}+B) = \underbrace{\cancel{A\bar{A}}}_{0}+AB+\bar{A}\bar{B}+\underbrace{\cancel{B\bar{B}}}_{0}$$

$$= AB + \bar{A}\bar{B}$$

Álgebra de Boole

- Exercícios: (cont.)

3. Mostrar que $\overline{A \oplus B} = \overline{A} \oplus B$ (cont.)

$$X \oplus B = \overline{X}B + X\overline{B}$$

Fazendo $X=\overline{A}$, temos:

$$\overline{A} \oplus B = \overline{\overline{A}}B + \overline{A}\,\overline{B} = AB + \overline{A}\,\overline{B} = A \oplus \overline{B}$$

Álgebra de Boole

- Exercícios: (cont.)

4. Mostrar que $AB + \overline{A}C + BC = AB + \overline{A}C$

$$AB+\overline{A}C+BC = AB+\overline{A}C+BC\underbrace{(A+\overline{A})}_{1} =$$

$$= AB + \overline{A}C + ABC + \overline{A}BC =$$

$$= AB(1+C)+\overline{A}C(1+B) = AB + \overline{A}C$$

cederj

Álgebra de Boole

- Exercícios: (cont.)

5. Simplifique a expressão lógica de F:

$$F(x,y,z) = \bar{x}yz + x\bar{y}z + \underbrace{xy\bar{z} + xyz}$$

$$= \bar{x}yz+x\bar{y}z+xy(\bar{z}+z) = xy+\bar{x}yz+x\bar{y}z =$$

$$= y(\underbrace{x+\bar{x}z}_{15b})+x\bar{y}z = y(x+z)+x\bar{y}z =$$

Álgebra de Boole

- Exercícios: (cont.)

5. (cont.)

$$F = \underbrace{xy + yz + x\bar{y}}z = yz + x(\underbrace{y+\bar{y}}_{15b}z) =$$

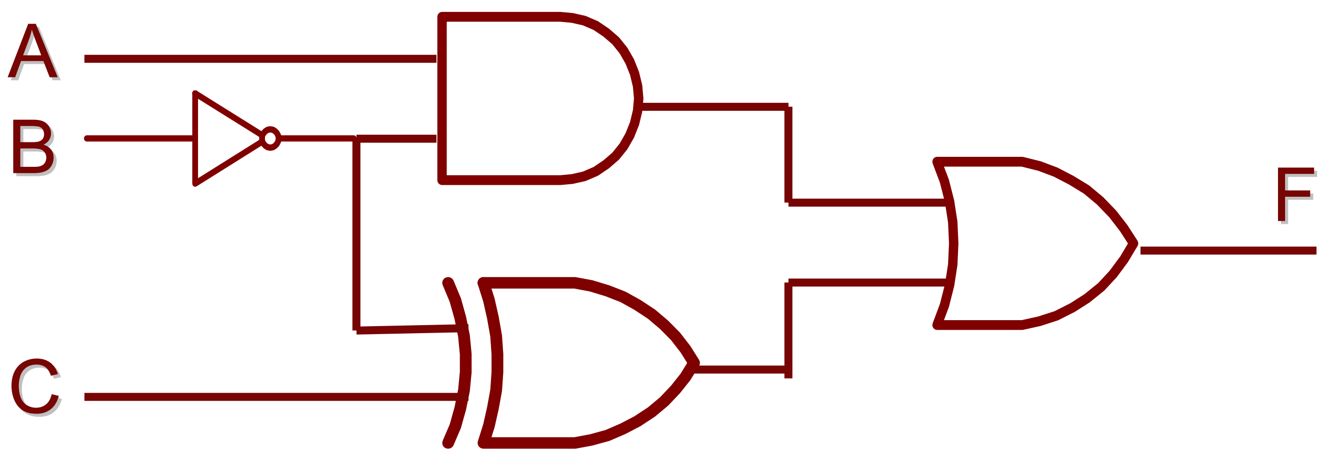
$$= yz + x(y+z)$$

$$F(x,y,z) = xy + xz + yz$$

Álgebra de Boole

- Exercícios: (cont.)

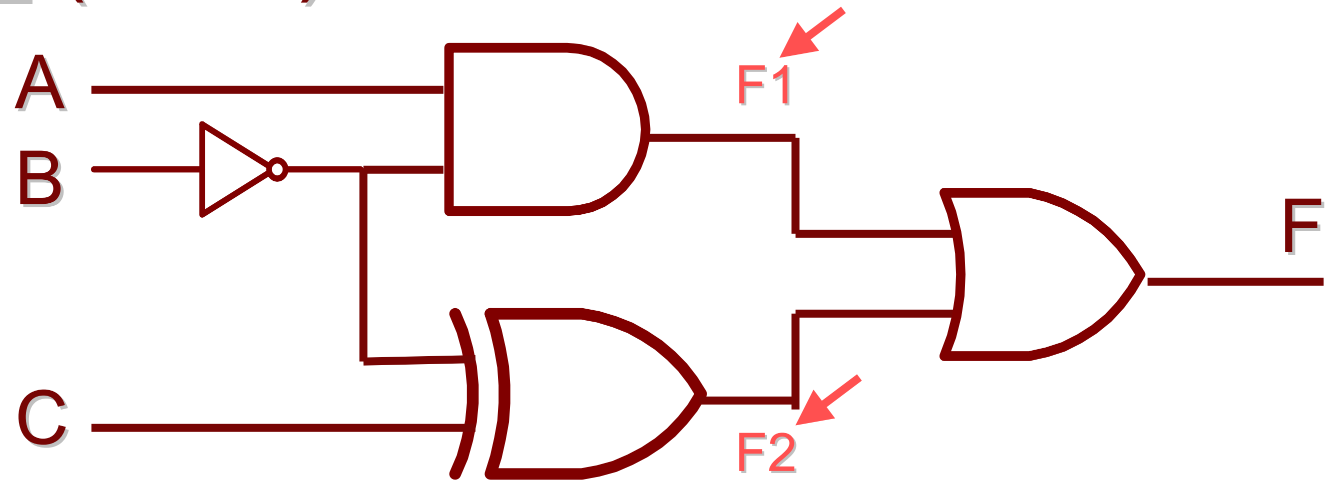
  6. Determine a expressão lógica para a saída F no circuito abaixo:



Álgebra de Boole

- Exercícios: (cont.)

6. (cont.)



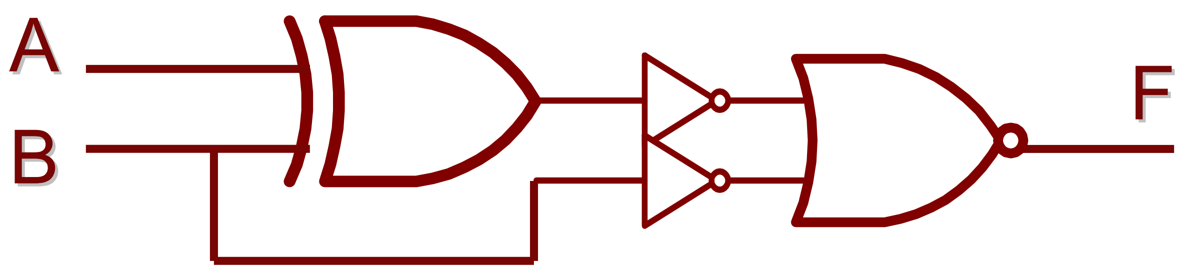
$$F1 = A\overline{B}\ ,\quad F2 = \overline{B} \oplus C = \overline{B}\,\overline{C} + BC$$

$$F = F1 + F2 = A\overline{B} + \overline{B}\,\overline{C} + BC$$

Álgebra de Boole

- Exercícios: (cont.)

  7. Dado o circuito abaixo, obtenha a expressão lógica mais simples que você puder para a saída F:

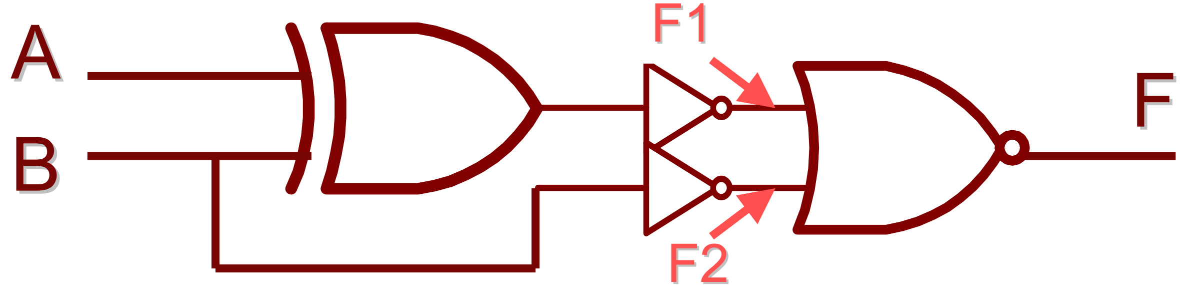


cederj

Álgebra de Boole

- Exercícios: (cont.)

7. (cont.)



$$F1 = \overline{A \oplus B} = \bar{A}\bar{B} + AB \,, \quad F2 = \bar{B}$$

$$F = \overline{F1+F2} = \overline{\bar{A}\bar{B}+AB+\bar{B}} = \overline{\bar{B}(\overset{1}{\cancel{1+\bar{A}}})+AB} =$$

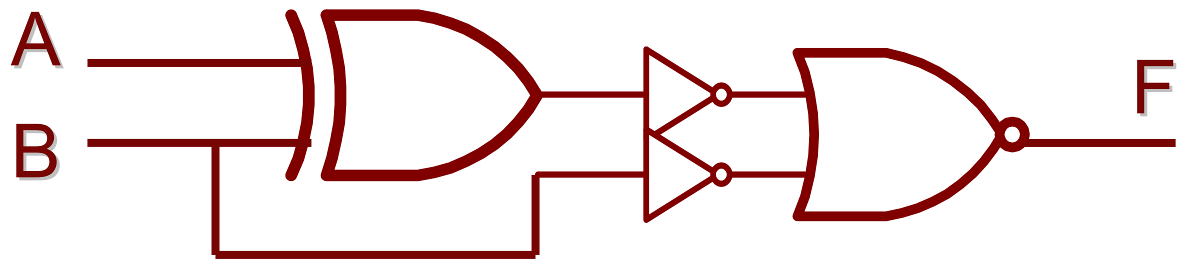
$$= \overline{\bar{B}+AB} = \overline{\bar{B}+A} \overset{M}{=} \bar{\bar{B}}.\bar{A} = \bar{A}B$$

cederj

Álgebra de Boole

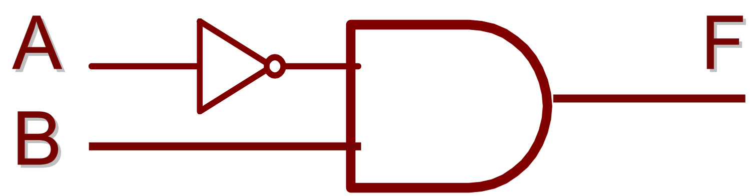
- Exercícios: (cont.)

7. (cont.)



$\equiv$

$$F(A,B) = \overline{A}B$$

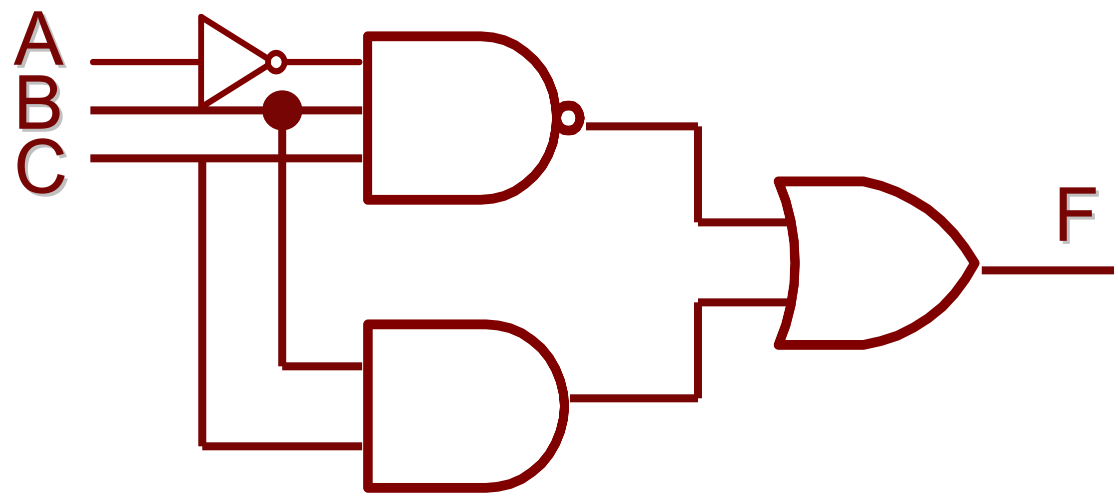


Álgebra de Boole

- Exercícios: (cont.)

8. Desenhe o circuito correspondente a expressão abaixo:

$$F = \overline{\overline{A}BC} + BC$$



Álgebra de Boole

- Exercícios: (cont.)

9. Simplifique a expressão de F do exemplo anterior:

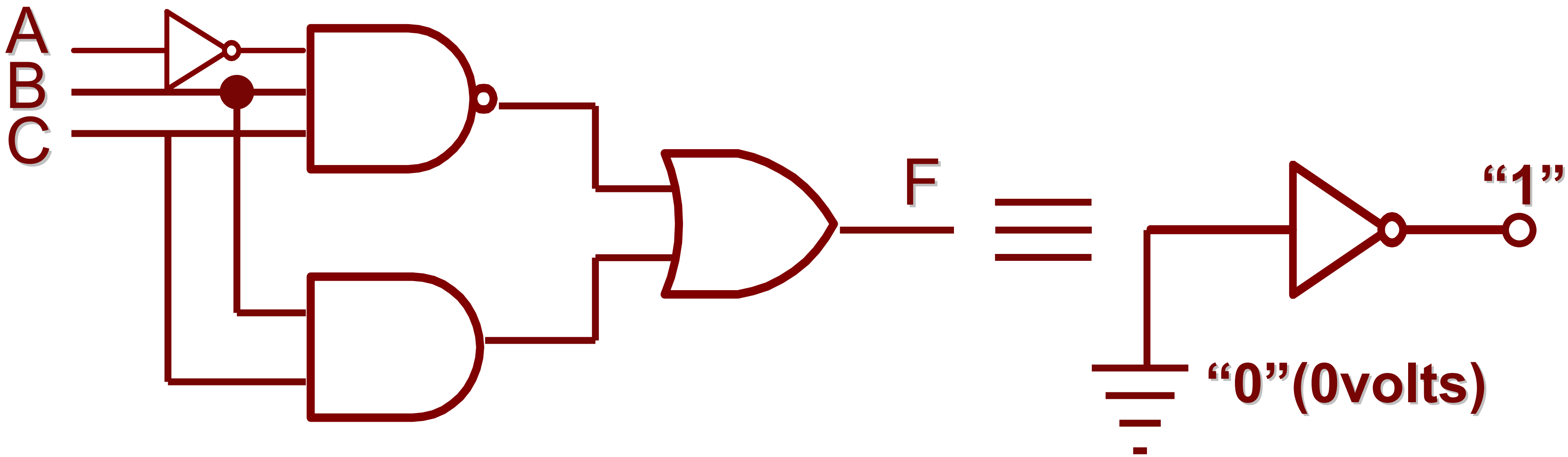
$$F = \overline{\overline{A}BC} + BC \overset{M}{=} A+\overline{B}+\underbrace{\overline{C} + BC}_{} =$$

$$= A+\underbrace{\overline{B}+\overline{C}+B}_{1} = A+1+\overline{C} = 1$$

Álgebra de Boole

- Exercícios: (cont.)

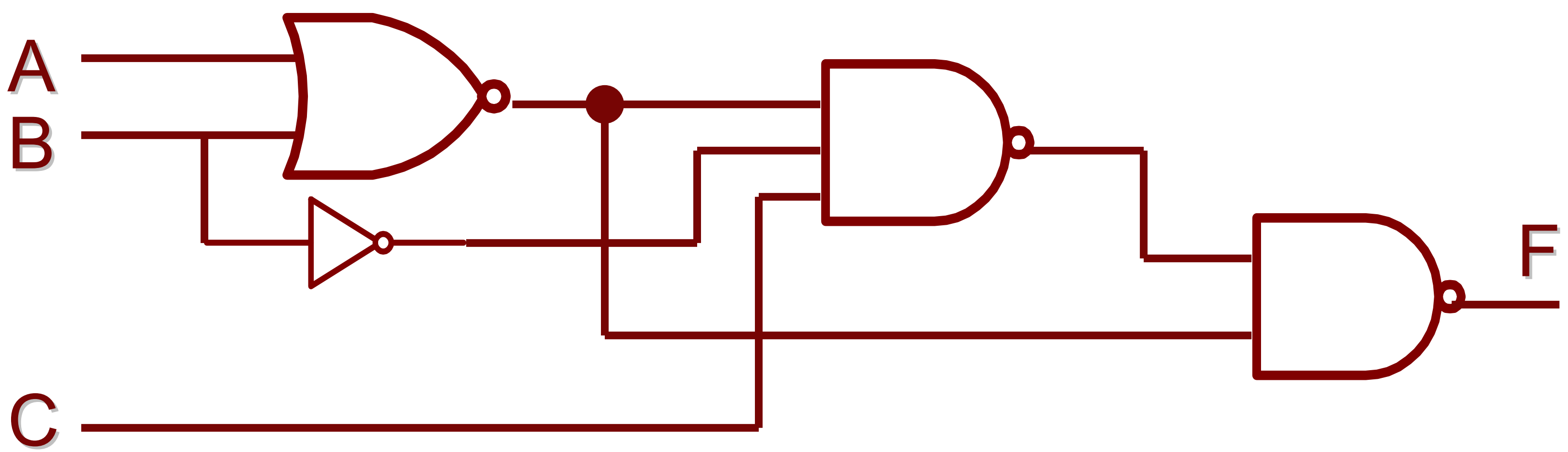
9. (cont.)



Álgebra de Boole

- Exercícios: (cont.)

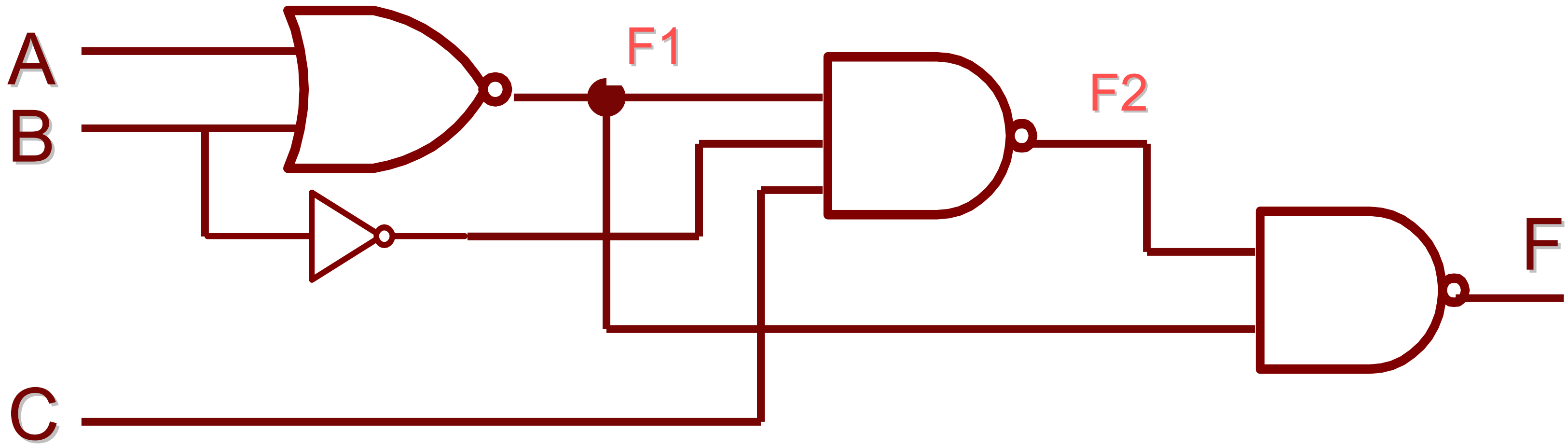
  10. No circuito abaixo, obtenha a expressão lógica mais simples que você puder para a saída F:



Álgebra de Boole

- Exercícios: (cont.)

10.



$$F1 = \overline{A+B} \overset{M}{=} \overline{A}.\overline{B}$$

$$F2 = \overline{F1\,\overline{B}\,C} = \overline{\overline{A}\,\overline{B}\,\overline{B}\,C} = \overline{\overline{A}\,\overline{B}\,C} \overset{M}{=} A+B+\overline{C}$$

$$F = \overline{F1F2} \overset{M}{=} \overline{F1}+\overline{F2}$$

Álgebra de Boole

- Exercícios: (cont.)

10. (cont.)

$$F1 = \overline{A+B} \overset{M}{=} \bar{A}.\bar{B}$$

$$F2 = \overline{F1\bar{B}\,C} = \overline{\bar{A}\bar{B}\bar{B}C} = \overline{\bar{A}\bar{B}C} \overset{M}{=} A+B+\bar{C}$$

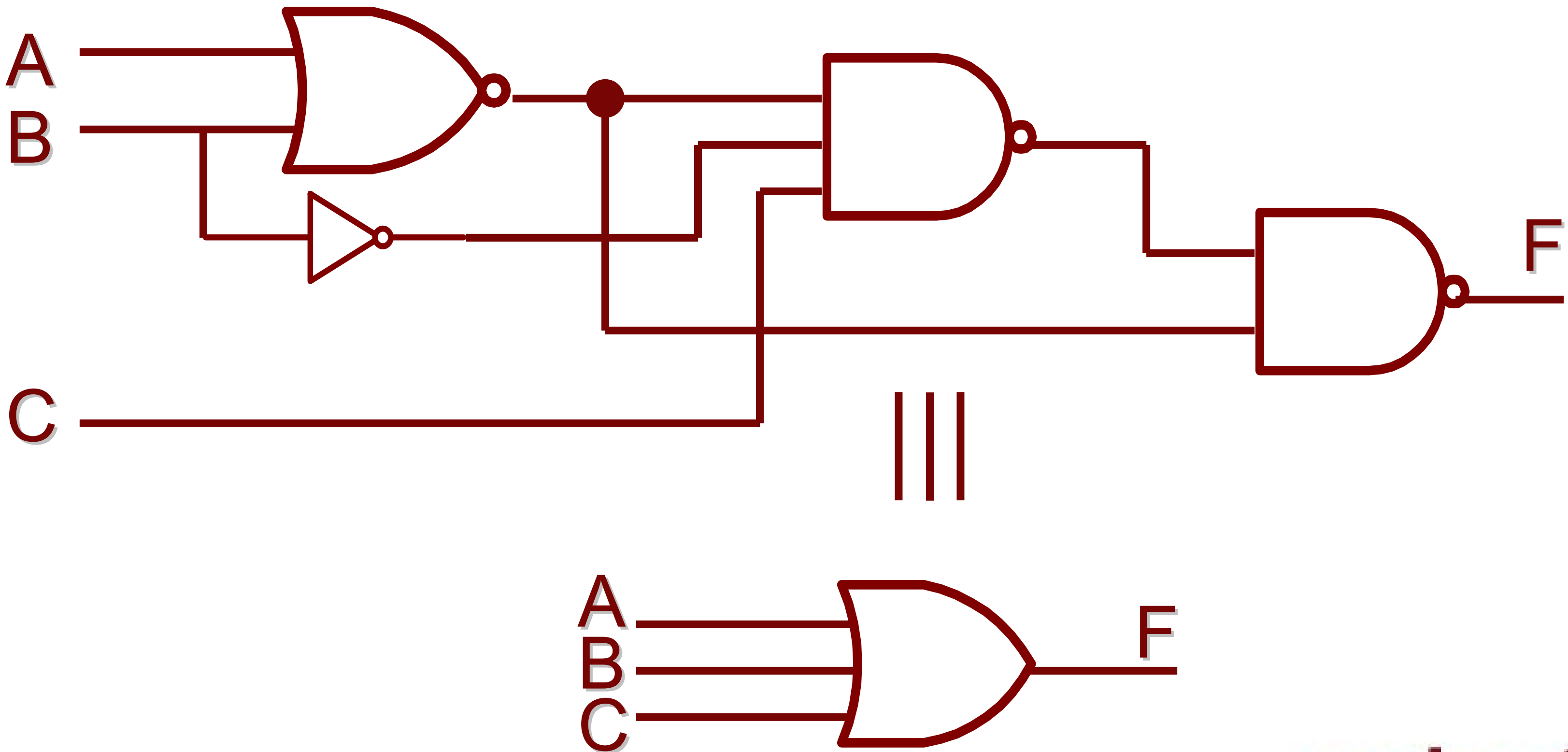
$$F = \overline{F1F2} \overset{M}{=} \overline{F1}+\overline{F2}$$

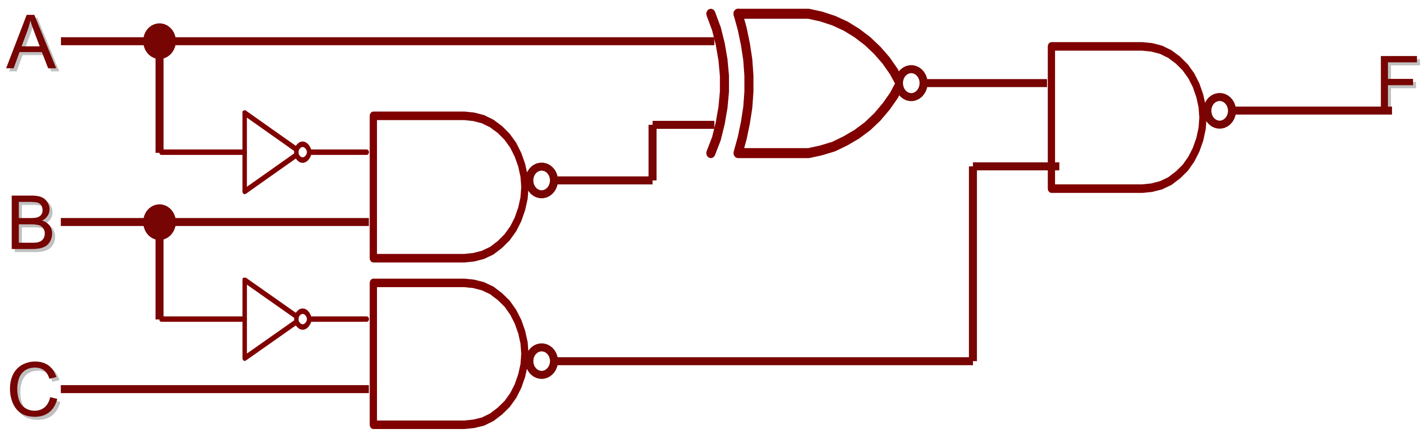
$$F = \overline{F1}+\overline{F2} = \overline{\overline{A+B}} + \overline{\overline{\bar{A}\bar{B}C}} = \underbrace{A+B+\bar{A}}\bar{B}C$$

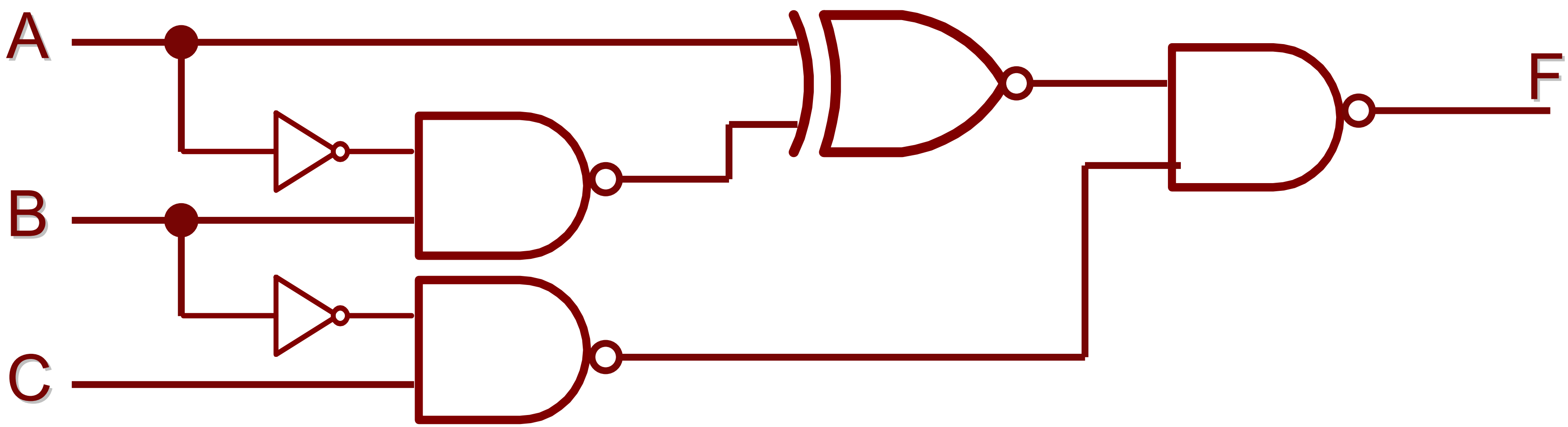
$$F = A+\underbrace{\bar{B}C+B} = A+B+C$$

- Exercícios: (cont.)

10. (cont.)




A
B
C
F


A
B
C
F

Introdução à Informática

Funções Lógicas

Ageu Pacheco e Alexandre Meslin

Funções Lógicas

- Objetivo da Aula:

- Estudar os principais métodos empregados na simplificação/minimização de funções lógicas (booleanas).

Funções Lógicas

- Conceito de mintermos e maxtermos:

  Considere a tabela verdade a seguir que expressa a função votador majoritário para 3 votantes:

Funções Lógicas

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Por inspeção na tabela podemos escrever:

$$F = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

cederj

Funções Lógicas

$$F(A,B,C) = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

- Repare que a expressão de F é constituída por uma soma de produtos lógicos, cada um composto pelas 3 variáveis de que F depende.

- Cada um destes produtos “completos” é denominado mintermo.

Funções Lógicas

- A descrição de uma função por meio de soma de produtos (mintermos) representa a função implementada nos pontos em que ela é "1"
- Notação compacta:

$$F = \underbrace{\overline{A}BC}_{} + \underbrace{A\overline{B}C}_{} + \underbrace{AB\overline{C}}_{} + \underbrace{ABC}_{}$$

$$F = m_3 + m_5 + m_6 + m_7$$

$$F = \sum(3,5,6,7) \Longrightarrow \text{(notação compacta)}$$

cederj

Funções Lógicas

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Função votador majoritário:

$$F = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$F = m_3 + m_5 + m_6 + m_7$$

$$F = \sum(3,5,6,7)$$

Funções Lógicas

- Invertendo F e aplicando a Lei de Morgan à equação resultante temos:

$$\overline{F} = \overline{\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC} \stackrel{M}{=}$$

$$\overline{F} \stackrel{M}{=} (\overline{\overline{A}BC}).(\overline{A\overline{B}C}).(\overline{AB\overline{C}}).(\overline{ABC}) \stackrel{M}{=}$$

$$\overline{F} \stackrel{M}{=} (A+\overline{B}+\overline{C})(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})$$

Funções Lógicas

$$\overline{F} = (A+\overline{B}+\overline{C})(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})$$

- $\overline{F}$ acha-se representada por um produto de somas onde cada soma contém as 3 variáveis de que $\overline{F}$ depende.

- Cada um destas somas “completas” é denominada maxtermo.

Funções Lógicas

- A descrição de uma função por meio de produtos de somas (maxtermos) representa a função implementada nos pontos em que ela é "0"
- Notação compacta:

$$\overline{F} = \underbrace{(A+\overline{B}+\overline{C})}\underbrace{(\overline{A}+B+\overline{C})}\underbrace{(\overline{A}+\overline{B}+C)}\underbrace{(\overline{A}+\overline{B}+\overline{C})}$$

$$\overline{F} = M_3 \cdot M_5 \cdot M_6 \cdot M_7$$

$$\overline{F} = \P\,(3,5,6,7) \Rightarrow \text{(notação compacta)}$$

cederj

Funções Lógicas

	A	B	C	F	$\overline{F}$
0	0	0	0	0	1
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

$$\overline{F} = (A+\overline{B}+\overline{C}).(\overline{A}+B+\overline{C}).$$
$$.(\overline{A}+\overline{B}+C).(\overline{A}+\overline{B}+\overline{C})$$

$$\overline{F} = M_3.M_5.M_6.M_7$$

$$\overline{F} = ¶\,(3,5,6,7)$$

Funções Lógicas

- Para achar F representada por maxtermos aplicamos Morgan à expressão de $\overline{F}$ descrita por mintermos:

$$\overline{F} = \overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+A\overline{B}\,\overline{C}$$

$$\overline{F} = m_0+m_1+m_2+m_4$$

Funções Lógicas

A	B	C	F	$\overline{F}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

$$\overline{F} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$$

$$\overline{F} = m_0 + m_1 + m_2 + m_4$$

Pela tabela já dá para perceber que F por maxtermos será:

$$F = M_0.M_1.M_2.M_4$$

Funções Lógicas

- Aplicando Morgan a $\overline{F}$:

$$\overline{\overline{F}} = \overline{\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}C+\bar{A}B\bar{C}+A\bar{B}\bar{C}} \overset{M}{=}$$

$$F \overset{M}{=} \overline{(\bar{A}\bar{B}\bar{C})}.\overline{(\bar{A}\bar{B}C)}.\overline{(\bar{A}B\bar{C})}.\overline{(A\bar{B}\bar{C})} \overset{M}{=}$$

Funções Lógicas

- Aplicando Morgan a F (cont.):

$$F=(\overline{\overline{A}}+\overline{\overline{B}}+\overline{\overline{C}})(\overline{\overline{A}}+\overline{\overline{B}}+\overline{C})(\overline{\overline{A}}+\overline{B}+\overline{\overline{C}})(\overline{A}+\overline{\overline{B}}+\overline{\overline{C}})$$

$$F=(A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(\overline{A}+B+C)$$

$$F = M_0.M_1.M_2.M_4 = \P(0,1,2,4)$$

Funções Lógicas

	A	B	C	F	$\overline{F}$
0	0	0	0	0	1
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

$$F = m_3+m_5+m_6+m_7$$

$$F = M_0.M_1.M_2.M_4$$

$$\overline{F} = m_0+m_1+m_2+m_4$$

$$\overline{F} = M_3.M_5.M_6.M_7$$

Funções Lógicas

Linha	x	y	z	Mintermo	Maxtermo
0	0	0	0	$m_0 = \overline{x}\ \overline{y}\ \overline{z}$	$M_0 = x + y + z$
1	0	0	1	$m_1 = \overline{x}\ \overline{y}\ z$	$M_1 = x + y + \overline{z}$
2	0	1	0	$m_2 = \overline{x}\ y\ \overline{z}$	$M_2 = x + \overline{y} + z$
3	0	1	1	$m_3 = \overline{x}\ y\ z$	$M_3 = x + \overline{y} + \overline{z}$
4	1	0	0	$m_4 = x\ \overline{y}\ \overline{z}$	$M_4 = \overline{x} + y + z$
5	1	0	1	$m_5 = x\ \overline{y}\ z$	$M_5 = \overline{x} + y + \overline{z}$
6	1	1	0	$m_6 = x\ y\ \overline{z}$	$M_6 = \overline{x} + \overline{y} + z$
7	1	1	1	$m_7 = x\ y\ z$	$M_7 = \overline{x} + \overline{y} + \overline{z}$

Funções Lógicas

- Exercícios: (mintermos e maxtermos)

1)Represente $F(A,B,C)=\sum 1,3,5,7$ por meio de produtos de maxtermos.

$$F(A,B,C) = ¶\, 0,2,4,6 = M_0.M_2.M_4.M_6$$

$$F = (A+B+C)(A+\overline{B}+C)(\overline{A}+B+C)(\overline{A}+\overline{B}+C)$$

Funções Lógicas

2) Represente a função F da tabela por meio de soma de mintermos e produtos de maxtermos.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Funções Lógicas

2)

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$F = \sum 1,2,4 = m_1+m_2+m_4$

$F = ¶\, 0,3,5,6,7$

$F = M_0.M_3.M_5.M_6.M_7$

Funções Lógicas

2) $F = m_1 + m_2 + m_4$

$$F = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$$

$$F = M_0 . M_3 . M_5 . M_6 . M_7$$

$$F = (A+B+C).(A+\overline{B}+\overline{C}).(\overline{A}+B+\overline{C}).$$
$$(\overline{A}+\overline{B}+C).(\overline{A}+\overline{B}+\overline{C})$$

Funções Lógicas

- Simplificação de expressões lógicas:

  Métodos:
  - Por manipulações algébricas
  - Por mapas de Karnaugh
  - Pelo método de Quine-McCluskey

Funções Lógicas

- Manipulações algébricas (exemplos):

  1) Simplificar a função
  $F(A,B,C) = \sum 3,5,6,7$
  (votador majoritário de 3 votantes)

  $$F = \overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$$

Funções Lógicas

1) (cont.)

$$F = \bar{A}BC+A\bar{B}C+\underbrace{AB\bar{C}+ABC}$$

$$F = \bar{A}\underbrace{BC+A\bar{B}C+AB}(\cancelto{1}{\bar{C}+C})$$

$$F = B(\underbrace{A+\bar{A}C})+A\bar{B}C = B(A+C)+A\bar{B}C$$

Funções Lógicas

1) (cont.)

$$F = B(A+C)+A\overline{B}C$$

$$F = AB+B\underbrace{C+A\overline{B}C} = AB+C(B+\overline{B}A)$$

$$F = AB+C(B+A) \Rightarrow F = AB+AC+BC$$

Funções Lógicas

2) $F(A,B,C) = \sum 3,7$

$$F = \bar{A}BC+ABC = BC(\bar{A}+A) = BC$$

Funções Lógicas

$$2)\ F(A,B,C) = \sum 3,7$$

$$F = \overline{A}BC + ABC = BC(\overline{A}+A) = BC$$

$$3)\ F(A,B,C) = A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$

$$F = A\overline{B}(\overline{C}+C) + AB(\overline{C}+C) = A\overline{B} + AB$$

$$F = A(\overline{B}+B) \Rightarrow F = A$$

Funções Lógicas

3) (cont.)

$F = A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC$

$F = \sum 4,5,6,7$

(mintermos adjacentes)

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Funções Lógicas

$$4)\ F(A,B,C) = \underbrace{\bar{A}\bar{B}\bar{C}+\bar{A}B\bar{C}}_{}+\underbrace{A\bar{B}\bar{C}+AB\bar{C}}_{}$$

$$F = \bar{A}\bar{C}(\bar{B}+B)+A\bar{C}(\bar{B}+B)$$

$$F = \bar{A}\bar{C}+A\bar{C} = \bar{C}(\bar{A}+A) = \bar{C}$$

Funções Lógicas

4)

$$F=\overline{A}\,\overline{B}\,\overline{C}+\overline{A}B\overline{C}+A\overline{B}\,\overline{C}+AB\overline{C}$$

$$F = \sum 0,2,4,6$$

(mintermos adjacentes)

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Funções Lógicas

5)

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

A	B	C	D	F
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Funções Lógicas

5) $F(A,B,C,D) = \sum 0,1,2,4,8,9,10,12,14$

$$F = \underbrace{\bar{A}\bar{B}\bar{C}\bar{D}}_{0}+\underbrace{\bar{A}\bar{B}\bar{C}D}_{1}+\underbrace{\bar{A}\bar{B}C\bar{D}}_{2}+\underbrace{\bar{A}B\bar{C}\bar{D}}_{4}+$$

$$+\underbrace{A\bar{B}\bar{C}\bar{D}}_{8}+\underbrace{A\bar{B}\bar{C}D}_{9}+\underbrace{A\bar{B}C\bar{D}}_{10}+\underbrace{AB\bar{C}\bar{D}}_{12}+\underbrace{ABC\bar{D}}_{14}$$

(a simplificação será feita mais adiante por meio do mapa de Karnaugh)

cederj

Funções Lógicas

- Mapas de Karnaugh (Maurice Karnaugh ~~1950)

  O mapa de Karnaugh é uma representação gráfica espacial da tabela verdade onde cada quadrado representa um mintermo de tal maneira que quadrados adjacentes contêm mintermos adjacentes.

Funções Lógicas

- Exemplos de mapas:

1)

	A	B	F
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

A \ B	0	1
0	0	1
1	1	0

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

Funções Lógicas

2) Mapa de 3 variáveis:

B

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
A 1	m_4	m_5	m_7	m_6

C

inversão na sequência

	A	B	C	F
m0	0	0	0	1
m1	0	0	1	0
m2	0	1	0	0
m3	0	1	1	1
m4	1	0	0	1
m5	1	0	1	0
m6	1	1	0	0
m7	1	1	1	1

Funções Lógicas

2) Mapa de 3 variáveis:

A \ BC		$\overline{B}$ 00	$\overline{B}$ 01	B 11	B 10
$\overline{A}$	0	1	0	1	0
A	1	1	0	1	0
		$\overline{C}$	C	C	$\overline{C}$

$\overline{B}\,\overline{C}$

BC

	A	B	C	F
m0	0	0	0	1
m1	0	0	1	0
m2	0	1	0	0
m3	0	1	1	1
m4	1	0	0	1
m5	1	0	1	0
m6	1	1	0	0
m7	1	1	1	1

Funções Lógicas

2) cont:

Pela tabela temos:

$F(A,B,C) = \sum 0,3,4,7$

$F = m_0 + m_3 + m_4 + m_7$

	A	B	C	F
m0	0	0	0	1
m1	0	0	1	0
m2	0	1	0	0
m3	0	1	1	1
m4	1	0	0	1
m5	1	0	1	0
m6	1	1	0	0
m7	1	1	1	1

Funções Lógicas

2) cont:

$$F = m_0+m_3+m_4+m_7$$

$$F = \bar{A}\bar{B}\bar{C}+\bar{A}BC+A\bar{B}\bar{C}+ABC$$

$$F = \bar{B}\bar{C}(\cancelto{1}{\bar{A}+A})+BC(\cancelto{1}{\bar{A}+A}) = \bar{B}\bar{C}+BC$$

Funções Lógicas

3) Mapa de 4 variáveis:

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1

A	B	C	D	F
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Funções Lógicas

3) $F(A,B,C,D) = \sum m_1, m_2, m_3, m_6, m_7, m_9, m_{12}$

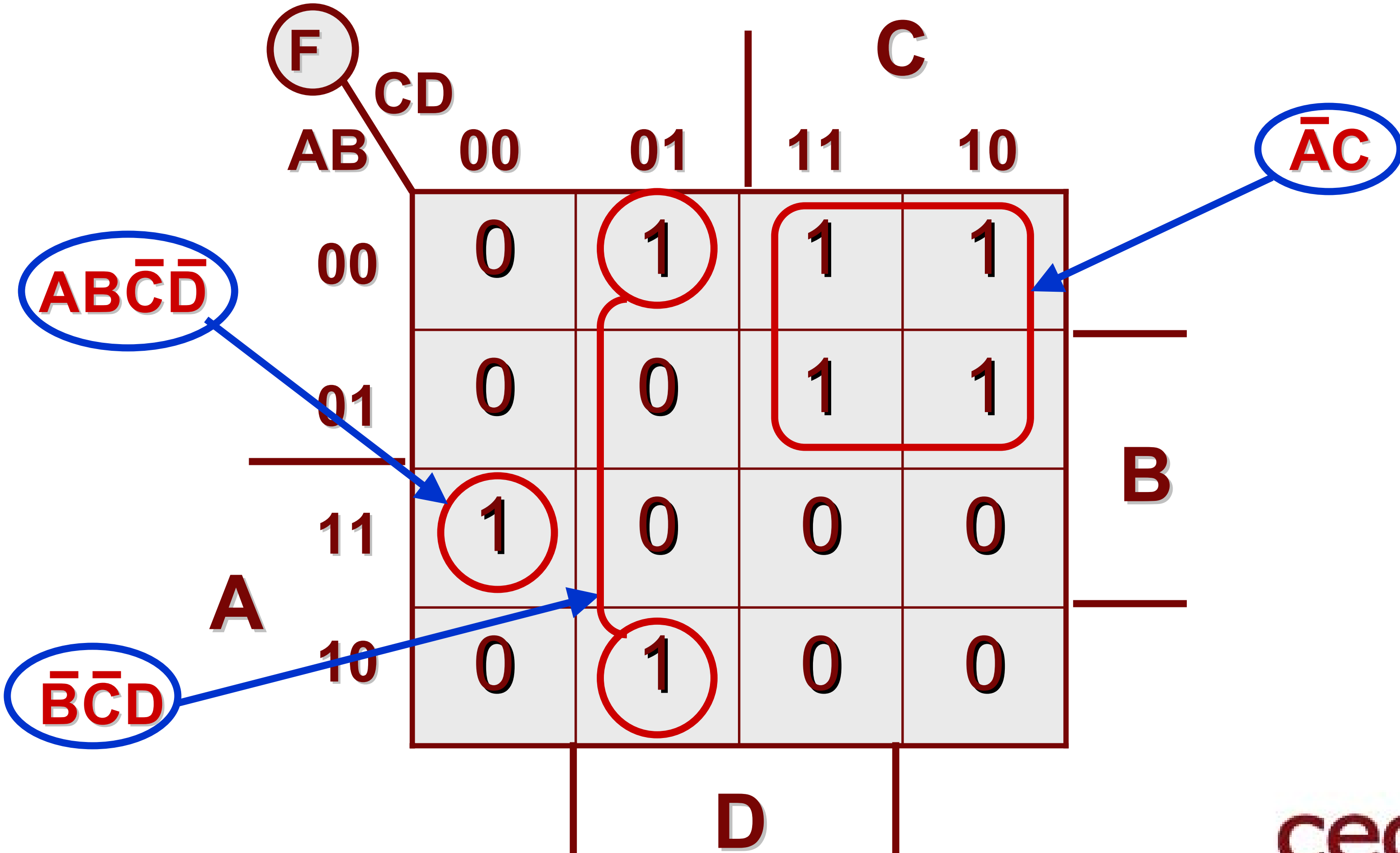
AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

F

AB \ CD	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	1	0	0	0
10	0	1	0	0

Funções Lógicas

3) $F(A,B,C,D) = \sum m_1, m_2, m_3, m_6, m_7, m_9, m_{12}$



Funções Lógicas

3) $F(A,B,C,D) = \sum m_1, m_2, m_3, m_6, m_7, m_9, m_{12}$

$$F = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\,\overline{C}D + AB\overline{C}\,\overline{D}$$

$$F = \overline{A}C + \overline{B}\,\overline{C}D + AB\overline{C}\,\overline{D}$$

- **Regras para simplificação com o mapa de Karnaugh:**

  1. Começar pelos quadrados isolados, isto é, que não tenham adjacentes. Eles representam mintermos que não podem ser simplificados mas que fazem parte da função.

Funções Lógicas

- **Regras (cont.):**

2. Procurar por quadrados que só tenham uma possibilidade de combinação.
3. Daí em diante procurar visualizar combinações envolvendo o máximo de quadrados.

- **Observação importante:**

  Combinações onde **todos** quadrados participantes já tenham sido utilizados em combinações prévias, não geram simplificações adicionais. Na verdade tais combinações geram termos redundantes.

Funções Lógicas

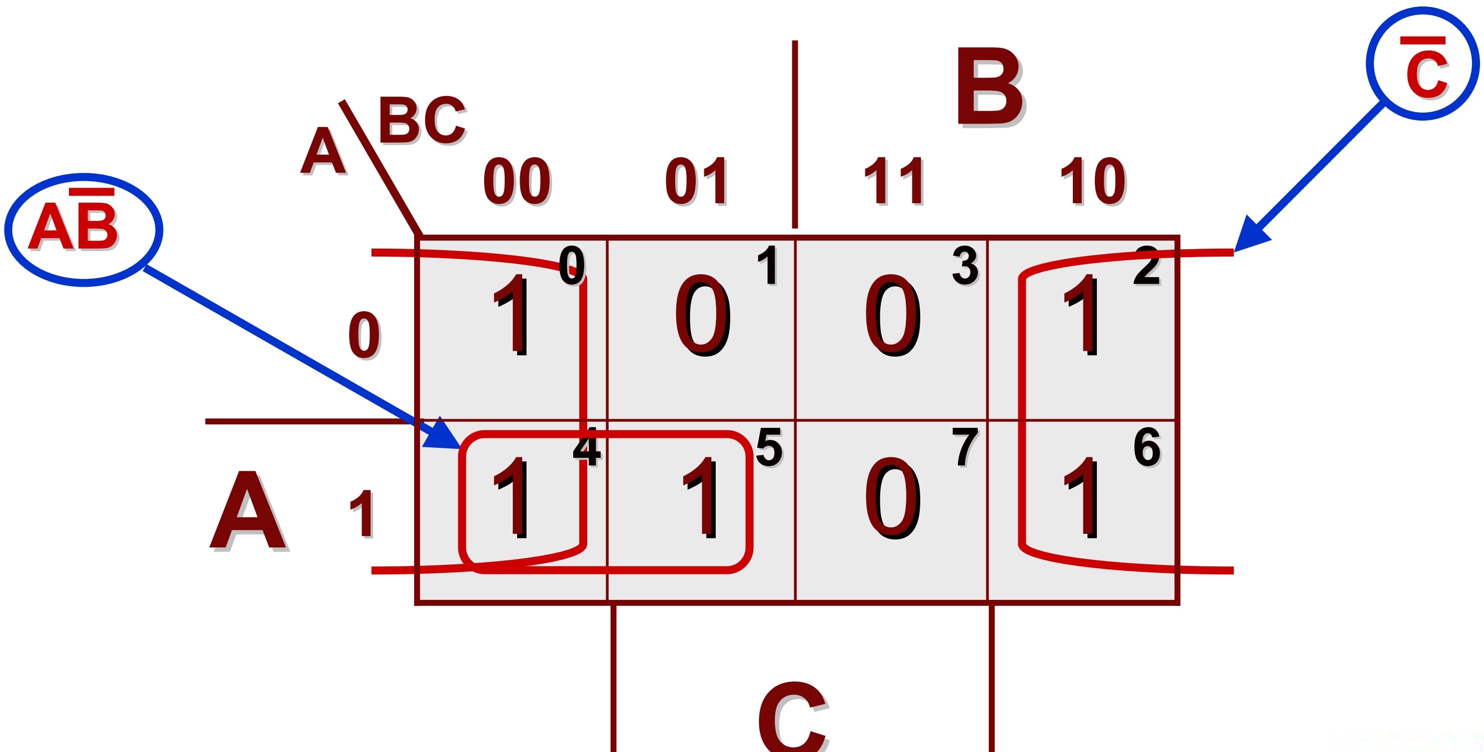
- “Sentidos” de mapas:

AB \ CD			
0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

CD \ AB			
0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

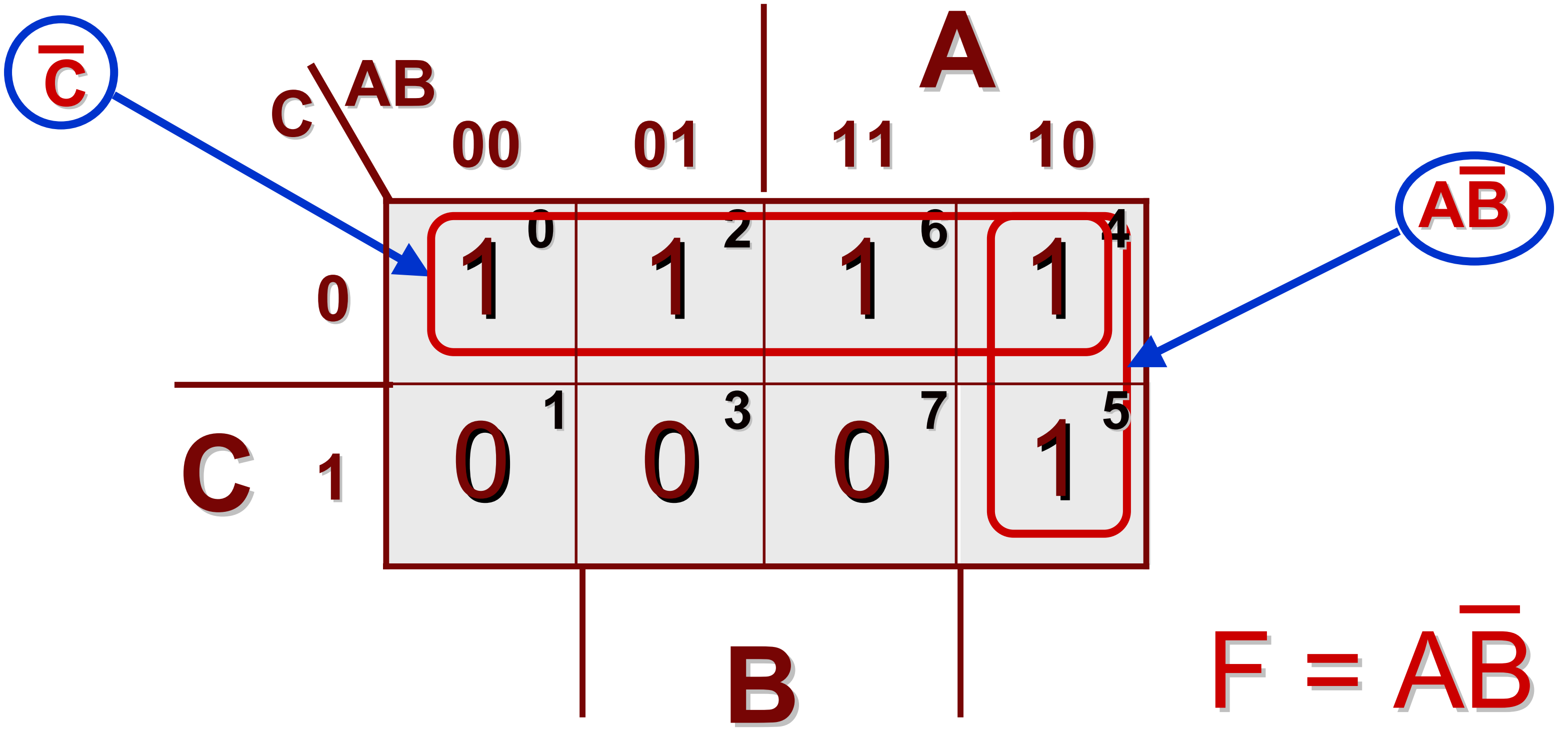
- Exercícios:

1. $F(A,B,C) = \sum 0,2,4,5,6$



Funções Lógicas

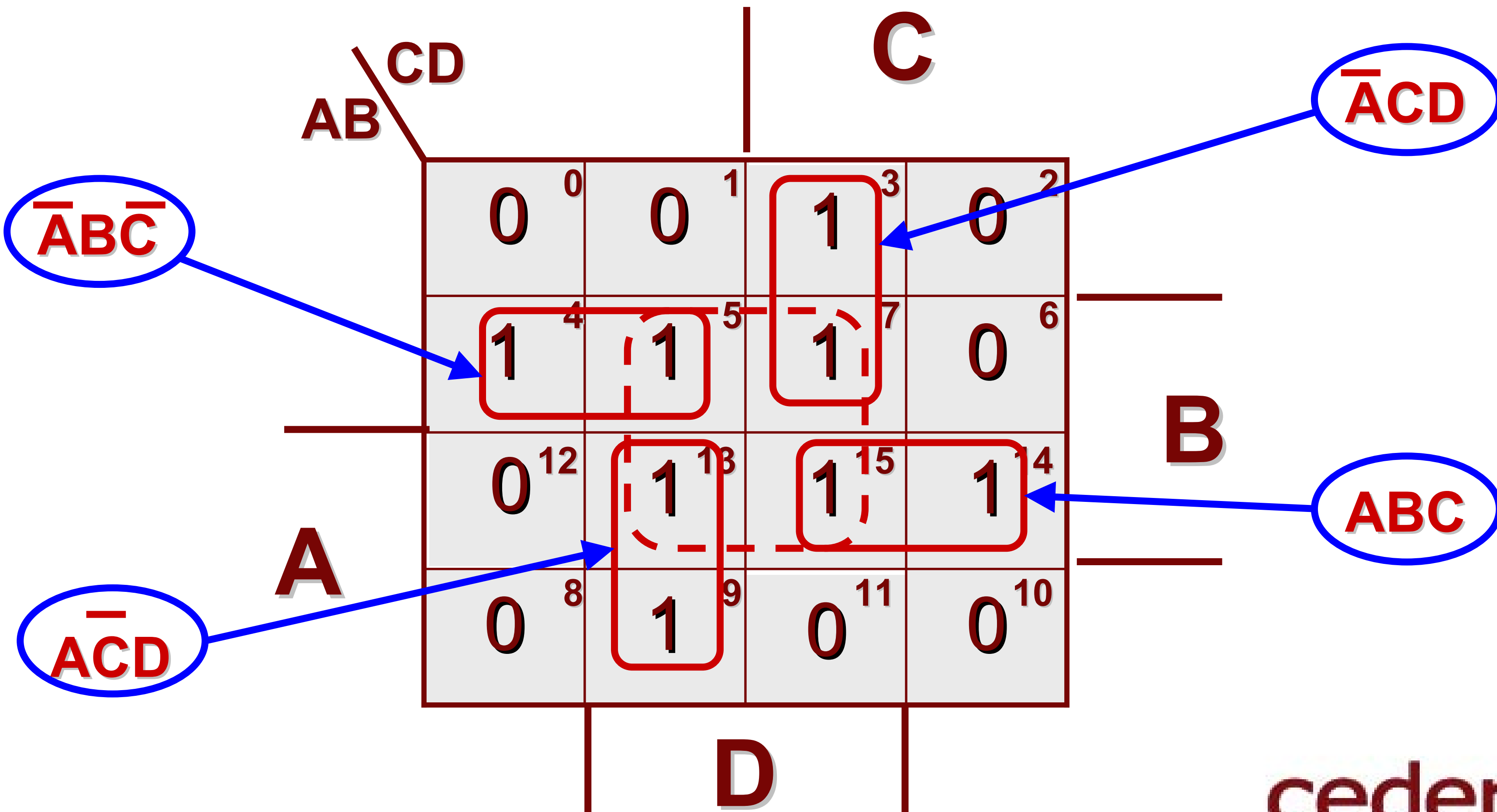
1a. $F(A,B,C) = \sum 0,2,4,5,6$



$$F = A\overline{B} + \overline{C}$$

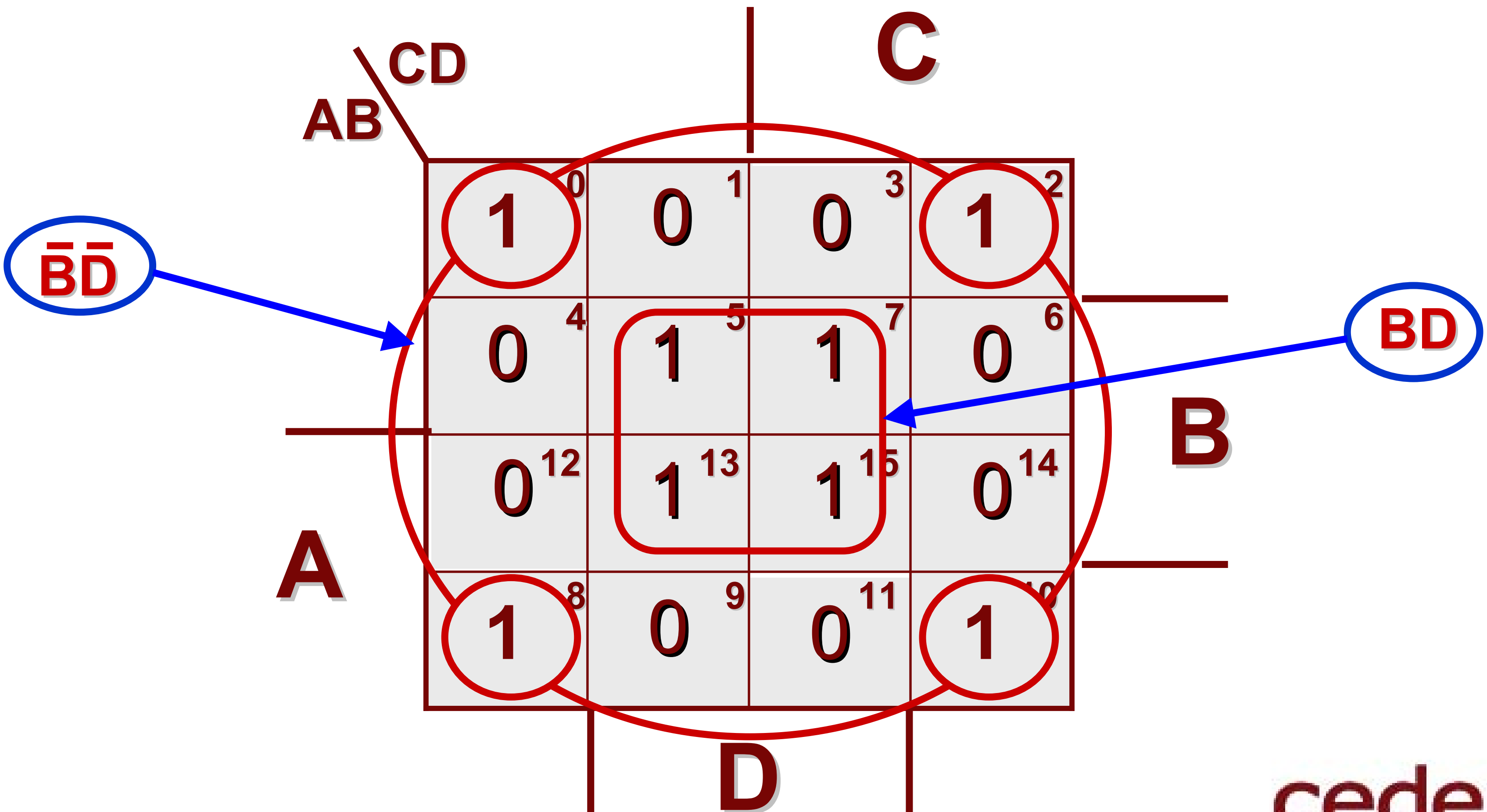
Funções Lógicas

2. $F(A,B,C,D) = \sum 3,4,5,7,9,13,14,15$



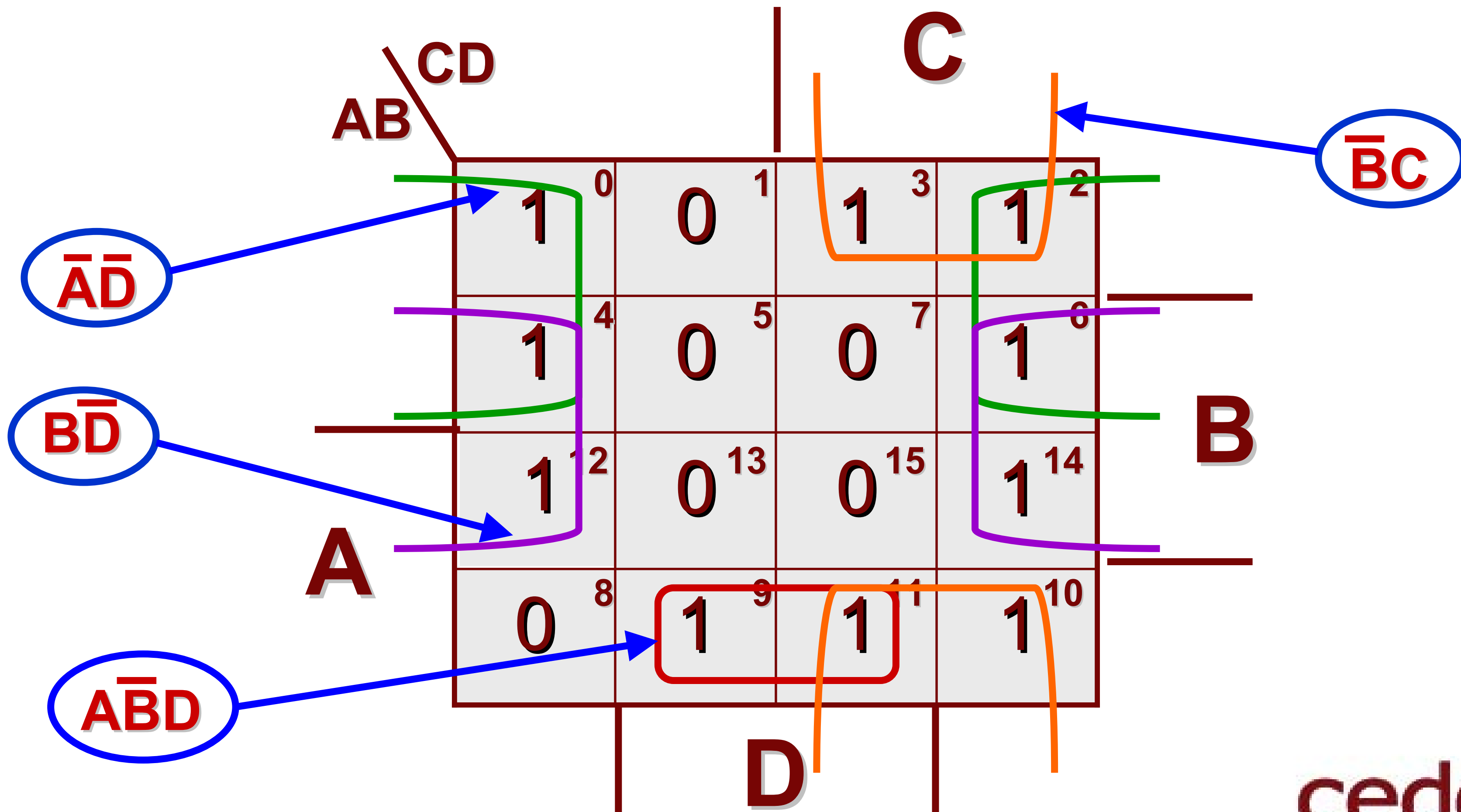
Funções Lógicas

3. $F(A,B,C,D) = \sum 0,2,5,7,8,10,13,15$



Funções Lógicas

4. $F(A,B,C,D) = \sum 0,2,3,4,6,9,10,11,12,14$



Funções Lógicas

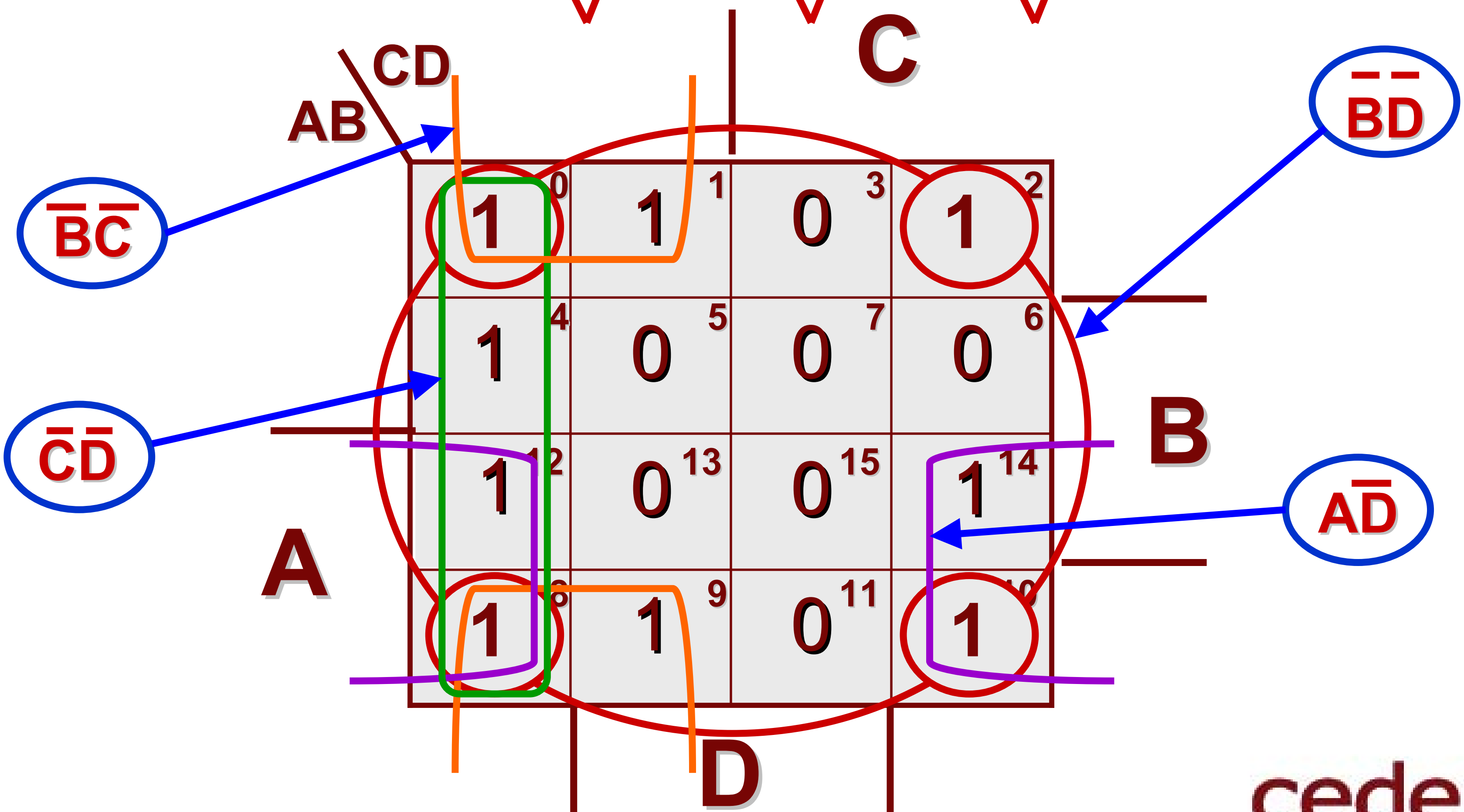
4. $F(A,B,C,D) = \sum 0,2,3,4,6,9,10,11,12,14$

$$F = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}\,\overline{D} +$$
$$\overline{A}BC\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD +$$
$$AB\overline{C}\,\overline{D} + ABC\overline{D}$$

$$F = \overline{A}\,\overline{D} + \overline{B}C + B\overline{D} + A\overline{B}D$$

Funções Lógicas

$$5.F(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\overline{D} + \overline{A}B\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D +$$
$$+ AC\overline{D} + A\overline{C}\,\overline{D} + \overline{B}\,\overline{C}D$$



Funções Lógicas

5) $F(A,B,C,D) = \sum 0,1,2,4,8,9,10,12,14$

$$F = \underbrace{\bar{A}\bar{B}\bar{C}\bar{D}}_{0}+\underbrace{\bar{A}\bar{B}\bar{C}D}_{1}+\underbrace{\bar{A}\bar{B}C\bar{D}}_{2}+\underbrace{\bar{A}B\bar{C}\bar{D}}_{4}+$$

$$+\underbrace{A\bar{B}\bar{C}\bar{D}}_{8}+\underbrace{A\bar{B}\bar{C}D}_{9}+\underbrace{A\bar{B}C\bar{D}}_{10}+\underbrace{AB\bar{C}\bar{D}}_{12}+\underbrace{ABC\bar{D}}_{14}$$

(a simplificação será feita mais adiante por meio do mapa de Karnaugh)

Funções Lógicas

5) $F(A,B,C,D) = \sum 0,1,2,4,8,9,10,12,14$

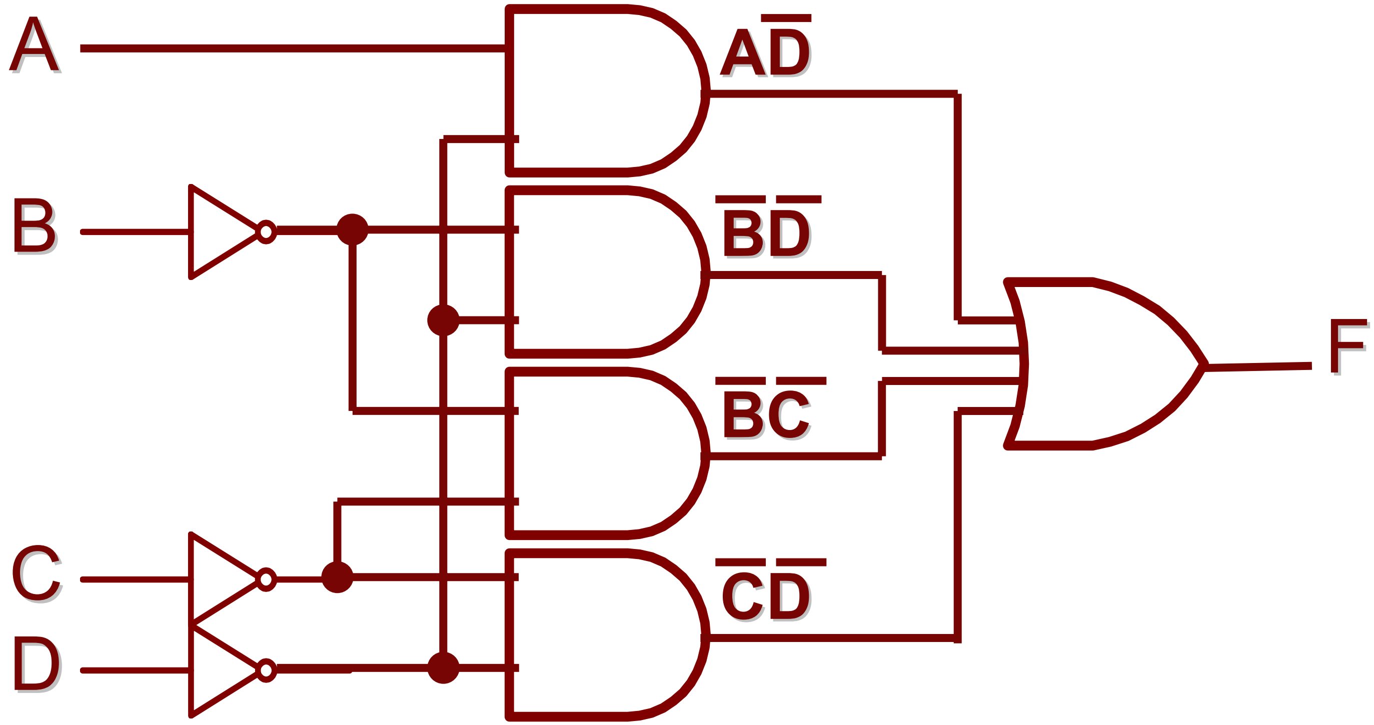
$$F = \underbrace{\bar{A}\bar{B}\bar{C}\bar{D}}_{0} + \underbrace{\bar{A}\bar{B}\bar{C}D}_{1} + \underbrace{\bar{A}\bar{B}C\bar{D}}_{2} + \underbrace{\bar{A}B\bar{C}\bar{D}}_{4} +$$

$$+ \underbrace{A\bar{B}\bar{C}\bar{D}}_{8} + \underbrace{A\bar{B}\bar{C}D}_{9} + \underbrace{A\bar{B}C\bar{D}}_{10} + \underbrace{AB\bar{C}\bar{D}}_{12} + \underbrace{ABC\bar{D}}_{14}$$

$$\mathbf{F(A,B,C,D) = A\bar{D} + \bar{B}\bar{C} + \bar{B}\bar{D} + \bar{C}\bar{D}}$$

Álgebra de Boole

- $F(A,B,C,D) = A\overline{D} + \overline{B}\,\overline{C} + \overline{B}\,\overline{D} + \overline{C}\,\overline{D}$



Introdução à Informática

Alexandre Meslin
(meslin@nce.ufrj.br)

Módulo 4 – Objetivos

- Aula 1
  - Componentes principais em um computador pessoal
  - Fluxo de informações em um computador idealizado
  - Componentes do PC
- Aula 2
  - Exercícios de fixação
- Aula 3
  - A instalação de uma placa em um PC
  - Componentes do Laptop
- Aula 4
  - Exercícios de fixação

Esta Aula

- Componentes principais em um computador pessoal
- Fluxo de informações em um computador idealizado
- Componentes do PC

Computador Pessoal

Computador Pessoal

- Monitor
  - Saída de Informação
  - Visualização
    - Imagens
    - Textos
  - Disponível em diversos tamanhos
    - 9”, 14”, 15”, 17”, 21”, etc.
  - Colorido ou monocromático

Computador Pessoal

Computador Pessoal

- Teclado
  - ❖ Entrada de dados

Computador Pessoal

Computador Pessoal

- Mouse
  - ❖ Dispositivo apontador
  - ❖ Mice
  - ❖ Track ball

Computador Pessoal

Computador Pessoal

- Caixas de Som
  - ❖ Passivas
  - ❖ Amplificadas
  - ❖ Sistemas com até 6 caixas

Computador Pessoal

Computador Pessoal

- Microfone
  - ❖ Dispositivo de entrada
  - ❖ Dispositivo analógico

Computador Pessoal

Computador Pessoal

- Gabinete
  - Interliga a maior parte dos componentes
  - Placa mãe
  - Memória
  - Discos
  - Interface serial
  - Interface paralela
  - Interface USB

Dentro do Gabinete

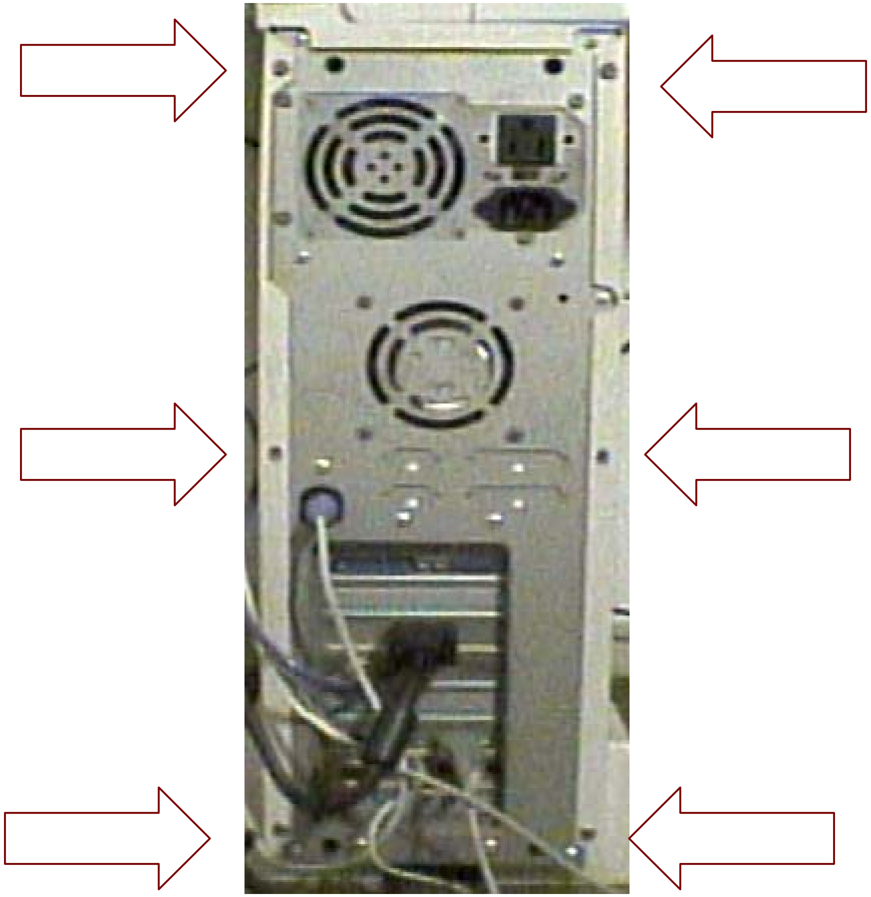
- Abrindo o gabinete
  - Ferramentas necessárias
    - Chave de fenda
    - Chave Philips
    - Chave de boca sextavada
    - Pinça ou alicate
  - Ferramentas que dificilmente serão necessárias
    - Multímetro
    - Ferro de soldar

Abrindo o Gabinete

- Grande variedade de tipos
- Não existe um padrão
- Necessidade de usar o bom senso

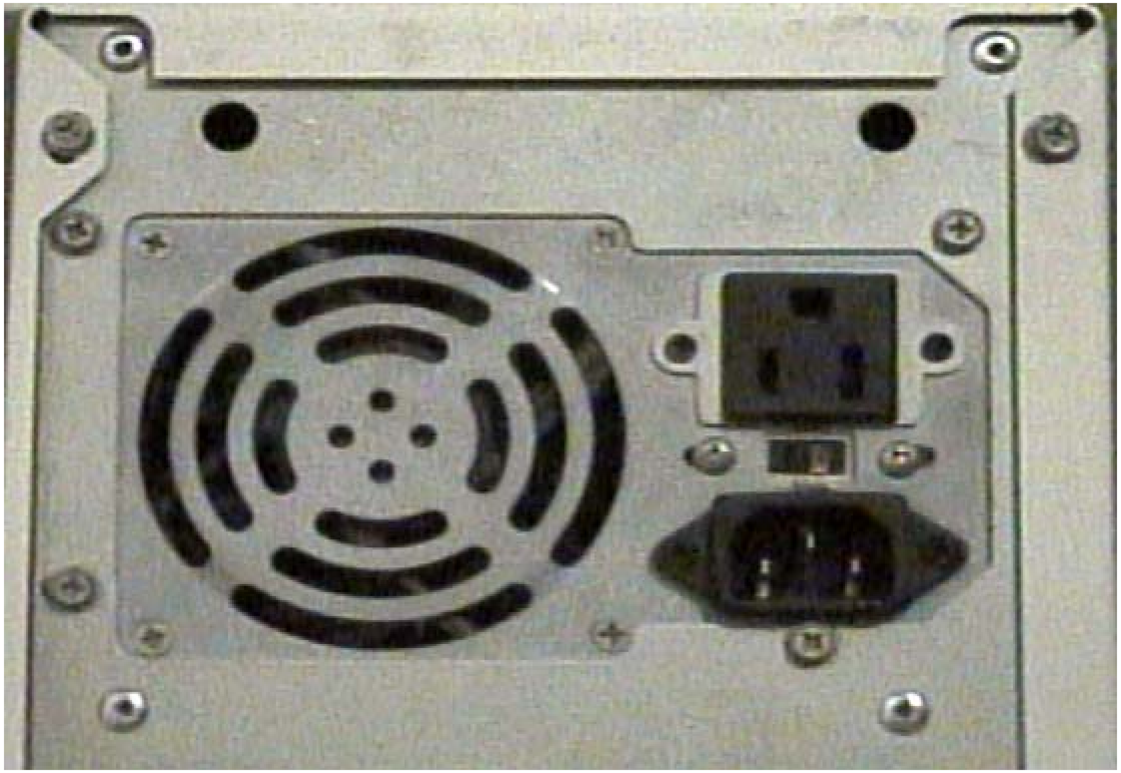
Um Exemplo de Gabinete

Um Exemplo de Gabinete



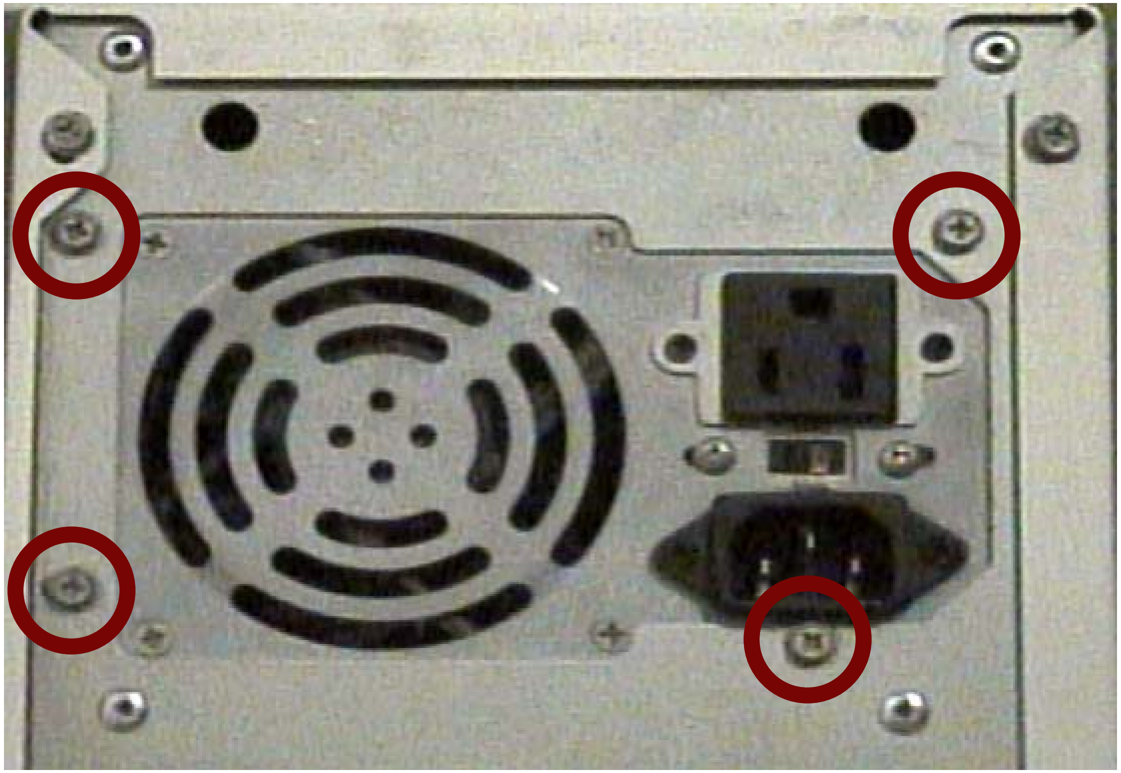
Cuidado

- Parafusos da fonte de alimentação



Cuidado

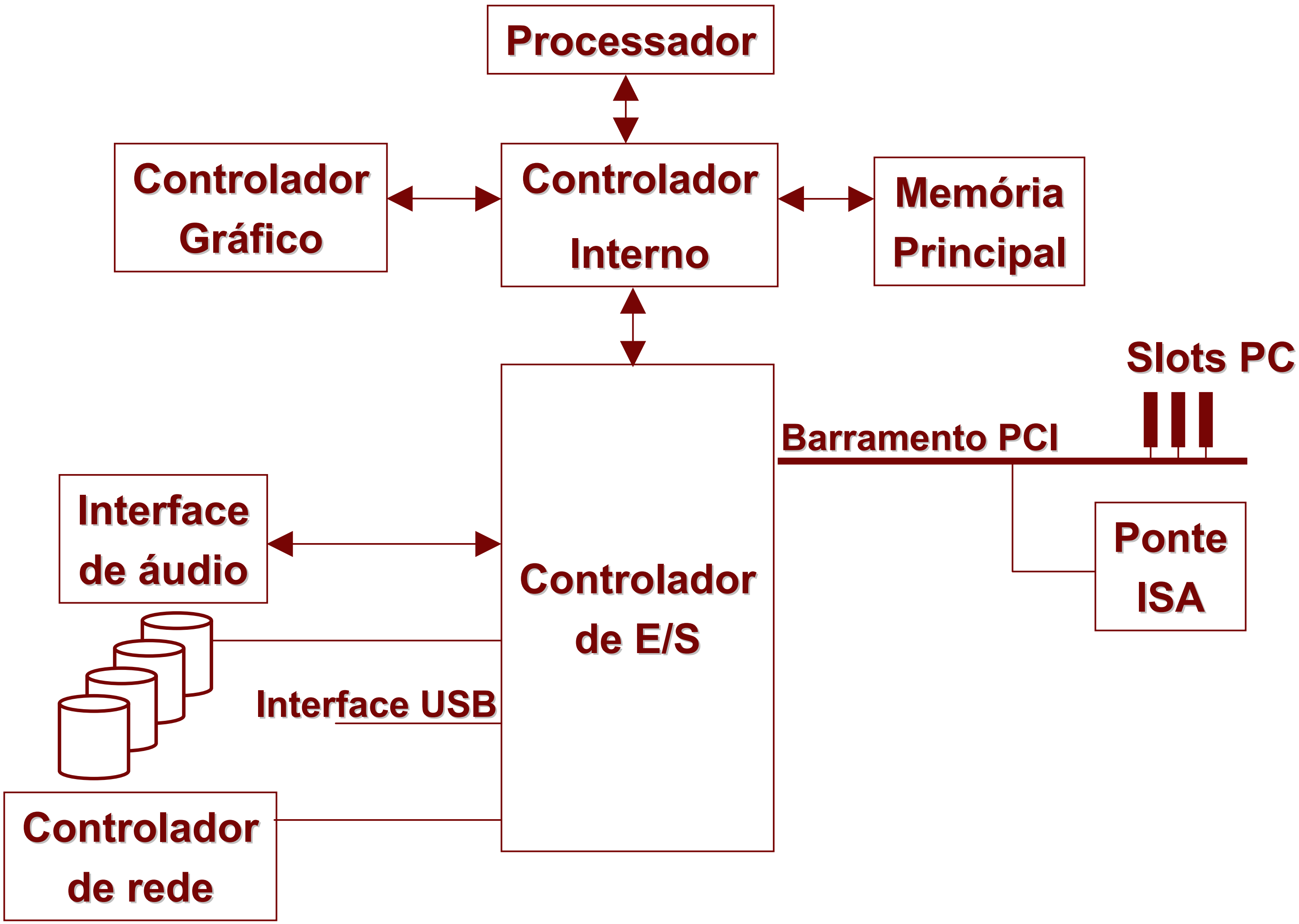
- Parafusos da fonte de alimentação



Dentro do Gabinete

- Fonte de alimentação
- Placa mãe
- Unidades de disco rígido
- Unidade de disco flexível (disquete)
- Cabos de conexão
  - Alimentação
  - Sinal

Fluxo de Informação

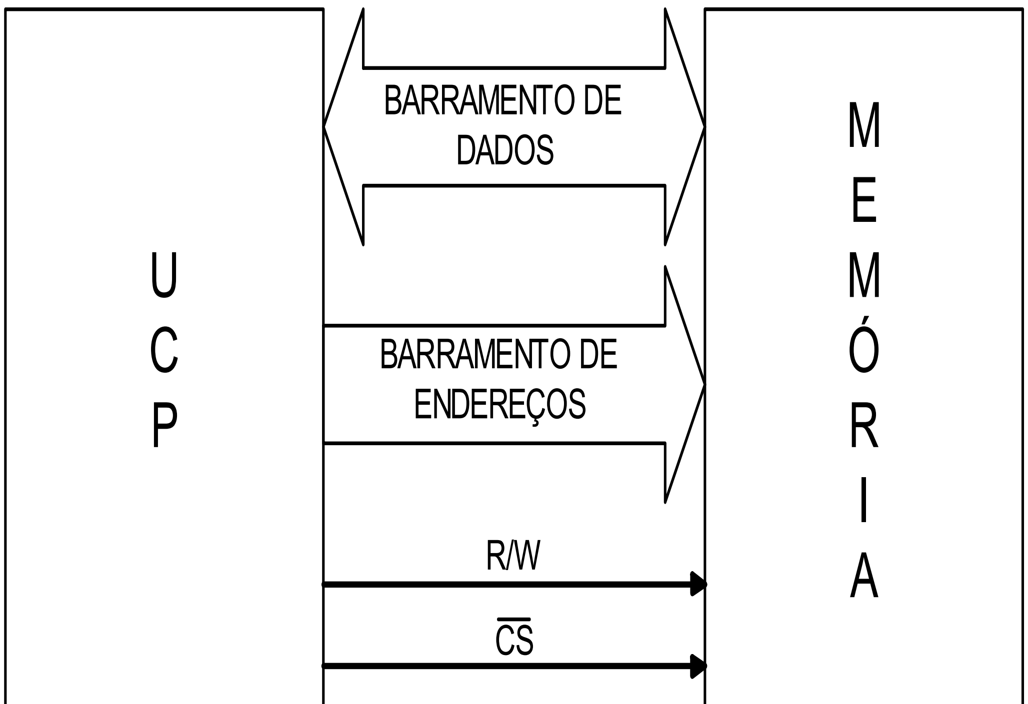


Barramento

- Barramento de Dados
  - ❖Permitem o transito de informações
- Barramento de Endereços
  - ❖Especificam a origem e/ou destino dos dados
- Barramento de Controle
  - ❖Especificam o tipo, tamanho, direção do dado

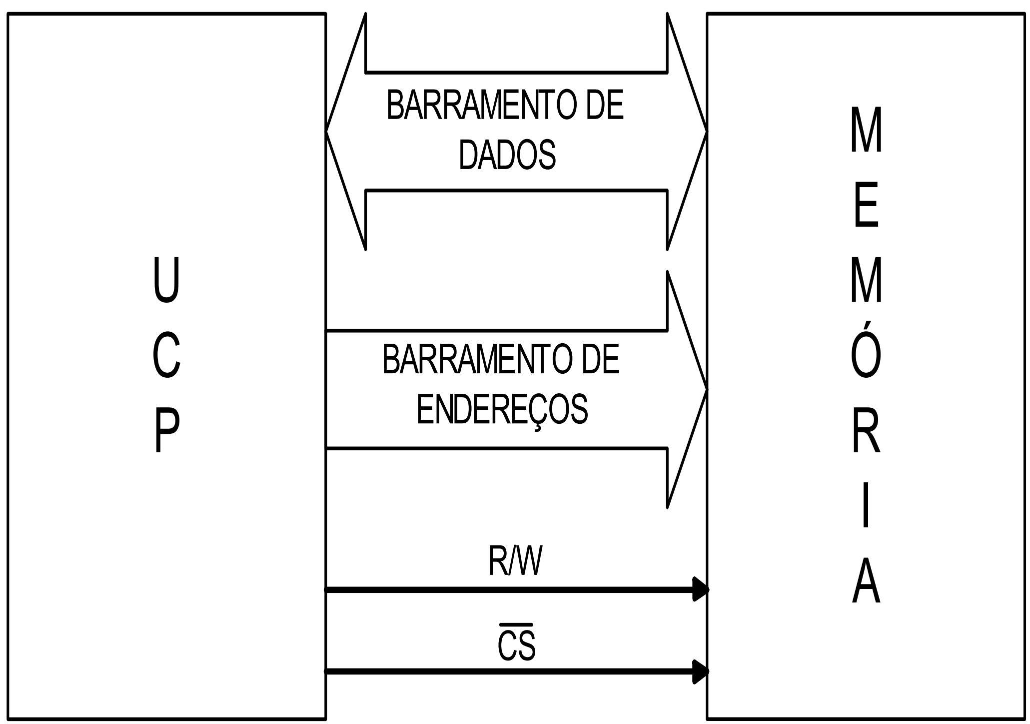
Ligação entre Processador e Memória

- Barramento de Dados (largura da palavra)
  - ❖ Número de bits que podem ser escritos/lidos simultaneamente



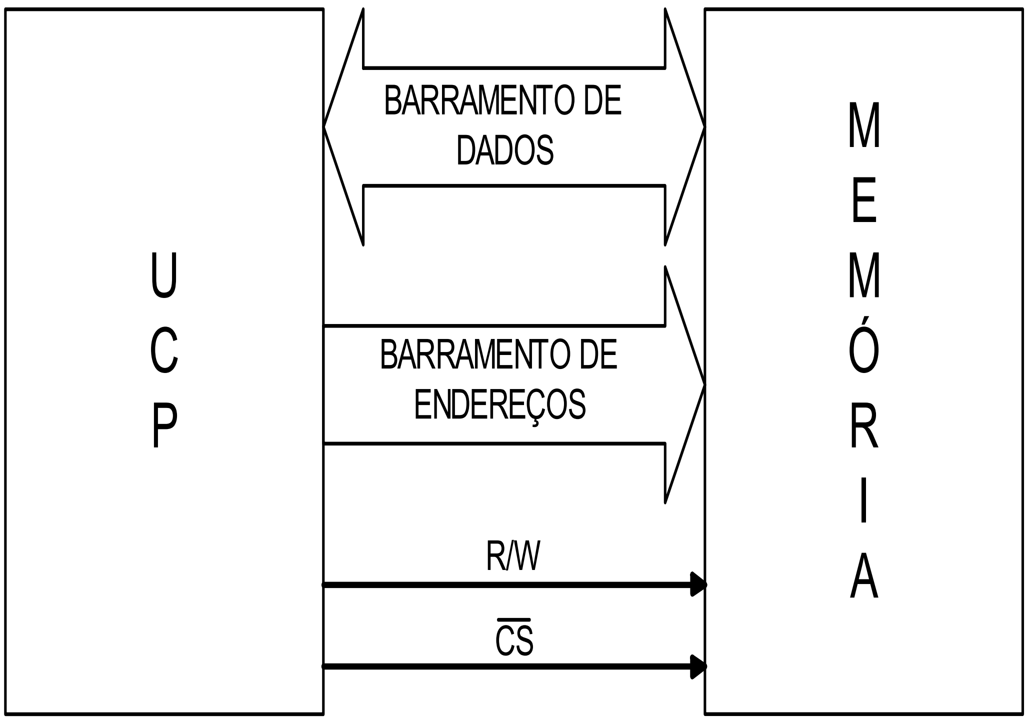
Ligação entre Processador e Memória

- Barramento de Endereço (capacidade de endereçamento)
  - ❖ Quantidade máxima de memória que pode ser endereçada



Ligação entre Processador e Memória

- Sinais de Controle
  - ❖ CS: seleciona o dispositivo de memória
  - ❖ R/W: informa o tipo de operação (Read, leitura – Write, escrita)



cederj

Fonte de Alimentação

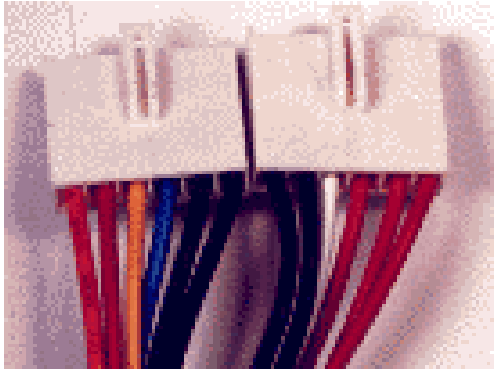
- Converte tensões e correntes que chegam ao local em níveis apropriados para o funcionamento do computador
- Fornece
  - +12V
  - +5V
  - 0V
  - -5V
  - -12V

Fonte de Alimentação

- Possue cabo para alimentar:
  - Placa mãe
  - Acionadores de discos rígidos e CD/DVD
  - Acionadores de disco flexível
  - Interfaces USB, seriais e paralelas
- Diversas potências disponíveis
- Tipo deve ser selecionado de acordo com a placa mãe e o gabinete
  - ATX
  - AT

Cabo de Alimentação de Placa Mãe

- Fonte padrão AT



Cabo de Alimentação de Periférico

- Amarelo → +12V
- Vermelho → +5V
- Preto → Terra (0V)

Controlador IDE

- Inicialmente era uma placa individual
- Passou a agrupar o controlador de disquete
- Posteriormente integrou-se com as portas seriais e paralelas (chamada de super-ide ou simplesmente ide)
- Atualmente está na placa mãe (on-board)
- Normalmente a placa mãe possui dois controladores
  - ❖ Primario
  - ❖ Secundário

Controlador IDE

- Possibilidade de conectar até dois dispositivos IDE
- Seleção do dispositivo através de jumper
- Primeiro dispositivo deve ser configurado como MASTER (mestre)
- Segundo dispositivo deve ser configurado como SLAVE (escravo)

Cabos IDE

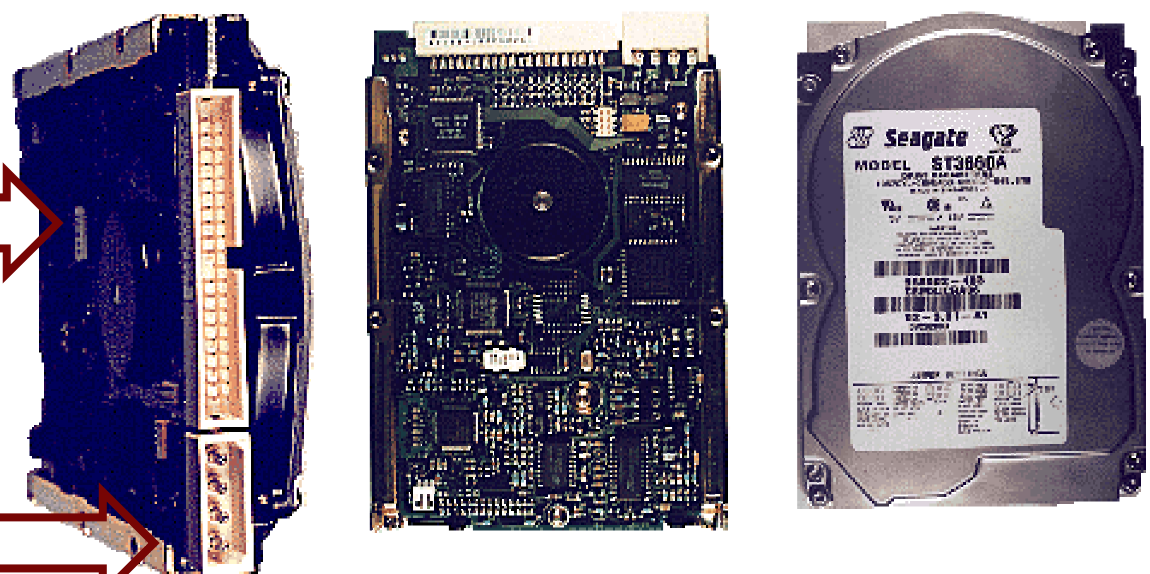
- Conecta o HD a interface IDE
- Transporte de dados e controle

Disco ou HD

- Armazenamento permanente
- Meio magnético

Cabo IDE

Cabo de Alimentação

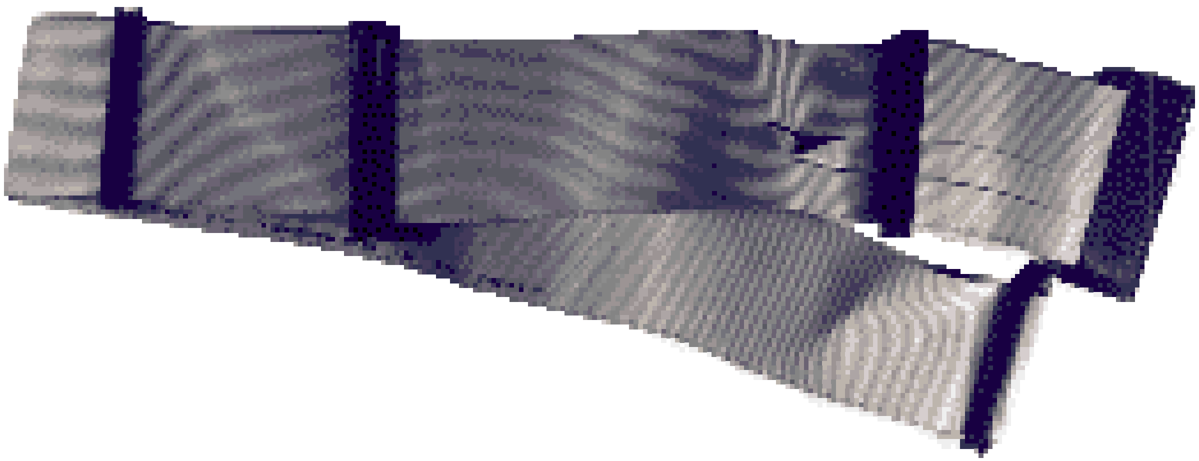


Unidade de Disquete

- Armazenamento magnético
- Mídia removível
- Conectada ao controlador de disquete através de um cabo com uma das pontas contendo uma inversão
- Possibilidade de conectar até dois dispositivos
- Identificação do dispositivo em relação à posição da inversão
  - Driva A: → depois da inversão
  - Drive B: → antes da inversão

Cabo do Acionador de Disco Flexível (Disquete)

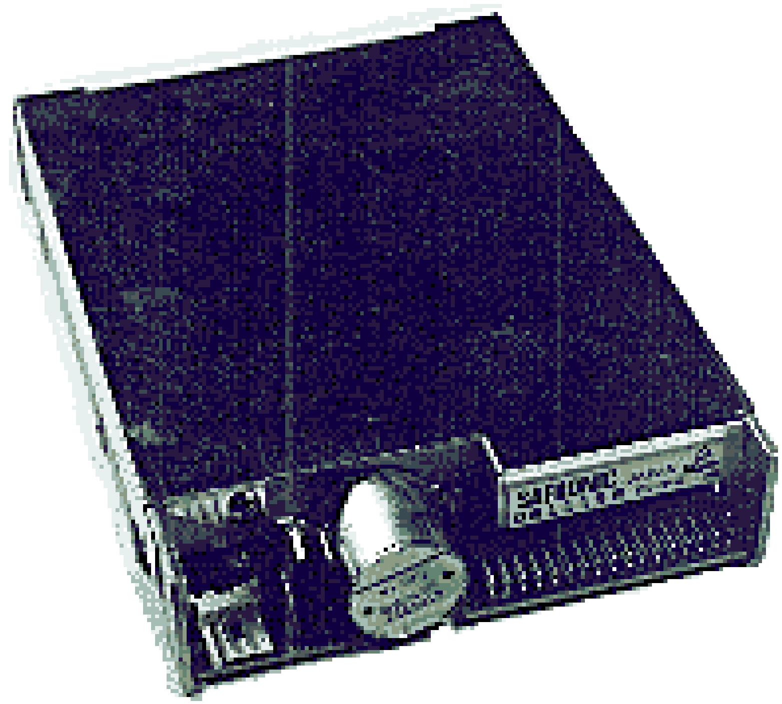
- Transporta controle e dados



Cabo de Alimentação de Unidade de Disco Flexível

- +12V (amarelo)
- 0V (preto)
- 0V (preto)
- +5V (vermelho)

Unidade de Disco Flexível



Tipos de Memória

- RAM (Random Access Memory)
- ROM (Read Only Memory)
- CMOS (Complementar Metal Oxide Silicon)

Exemplo de Uso de Memórias

- Boot de um computador tipo IBM-PC
  - O programa gravado na ROM é executado
  - O programa da ROM lê as informações da memória CMOS para identificar a configuração do sistema
  - Um programa é carregado do disco para a memória RAM para ser executado
  - Este programa carrega o Sistema Operacional

Memória RAM SIMM 30 pinos

- 8 bits
- Paridade Opcional
- Capacidade de 1 ou 4 Mbytes
- Velocidade até 70 ns
- (obsoleta há muito tempo)

Memória RAM SIMM 72 pinos

- 32 bits
- Paridade Opcional (muito rara)
- Capacidade de 4/8/16/32 Mbytes
- Velocidade de até 60 ns
- Opcionalmente pode ser EDO (somente para Pentium's)
- Quase obsoleta (difícil de encontrar)

Memória RAM DIMM 168 pinos

- Atualmente no mercado
- 64 bits
- Paridade opcional
- Capacidade começando em 16 Mbytes
- Velocidade de 50 ns
- Opcionalmente pode ser EDO
- Tempo de acesso durante rajada de até 7 ns

Composição de Memória (Pentium II/III/IV)

- Barramento de 64 bits de dados
- Placa Mãe com 1 até 4 slots de 168 pinos (64 bits)
- Módulos de 168 pinos adicionados independentemente
  - 16 Mbytes = 1 módulo de 16 Mbytes
  - 128 Mbytes = 1 módulo de 128 Mbytes ou 2 módulos de 64 Mbytes ou 1 módulo de 64 Mbytes e 2 módulos de 32 Mbytes

Outras Interfaces

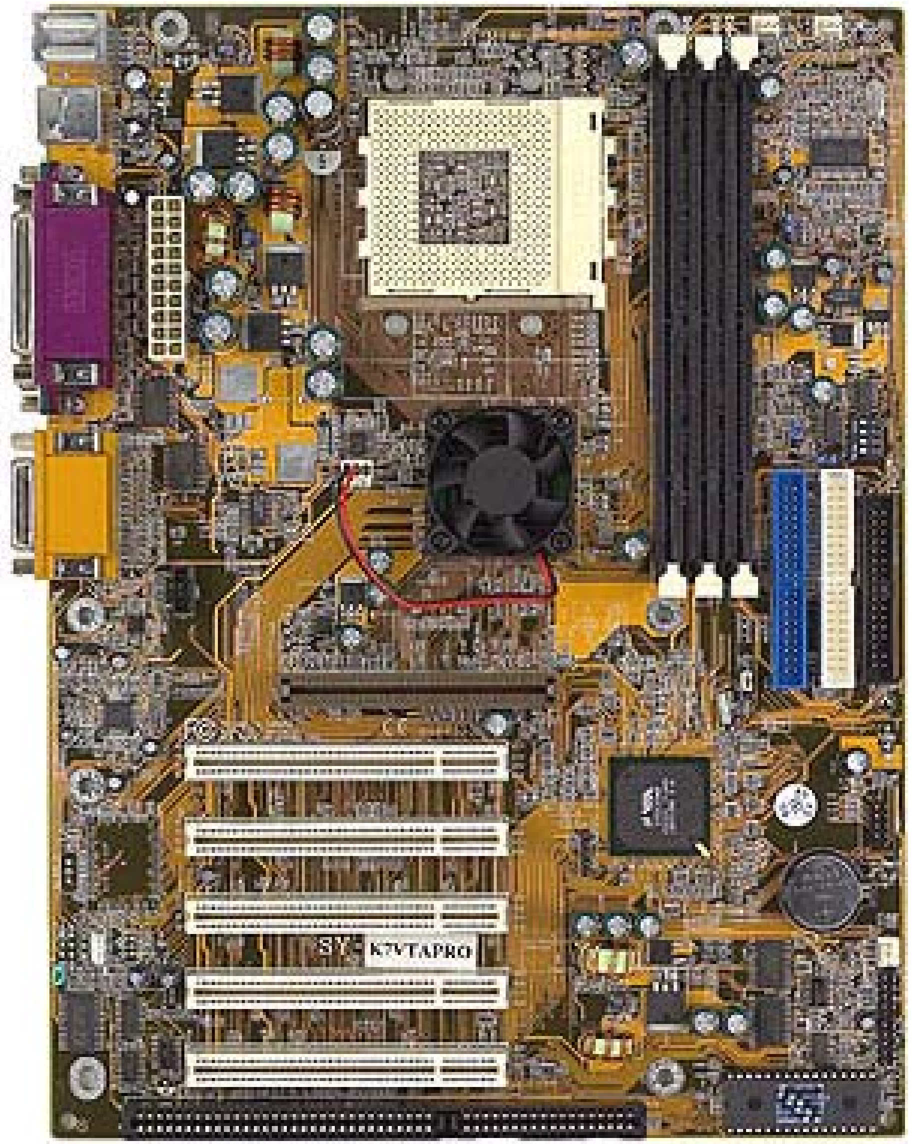
- Placa de Vídeo
- Placa de Som
- Placa de Rede
- Placa de Modem

- Atualmente podem estar integradas com a placa mãe (on-board)

Placa Mãe

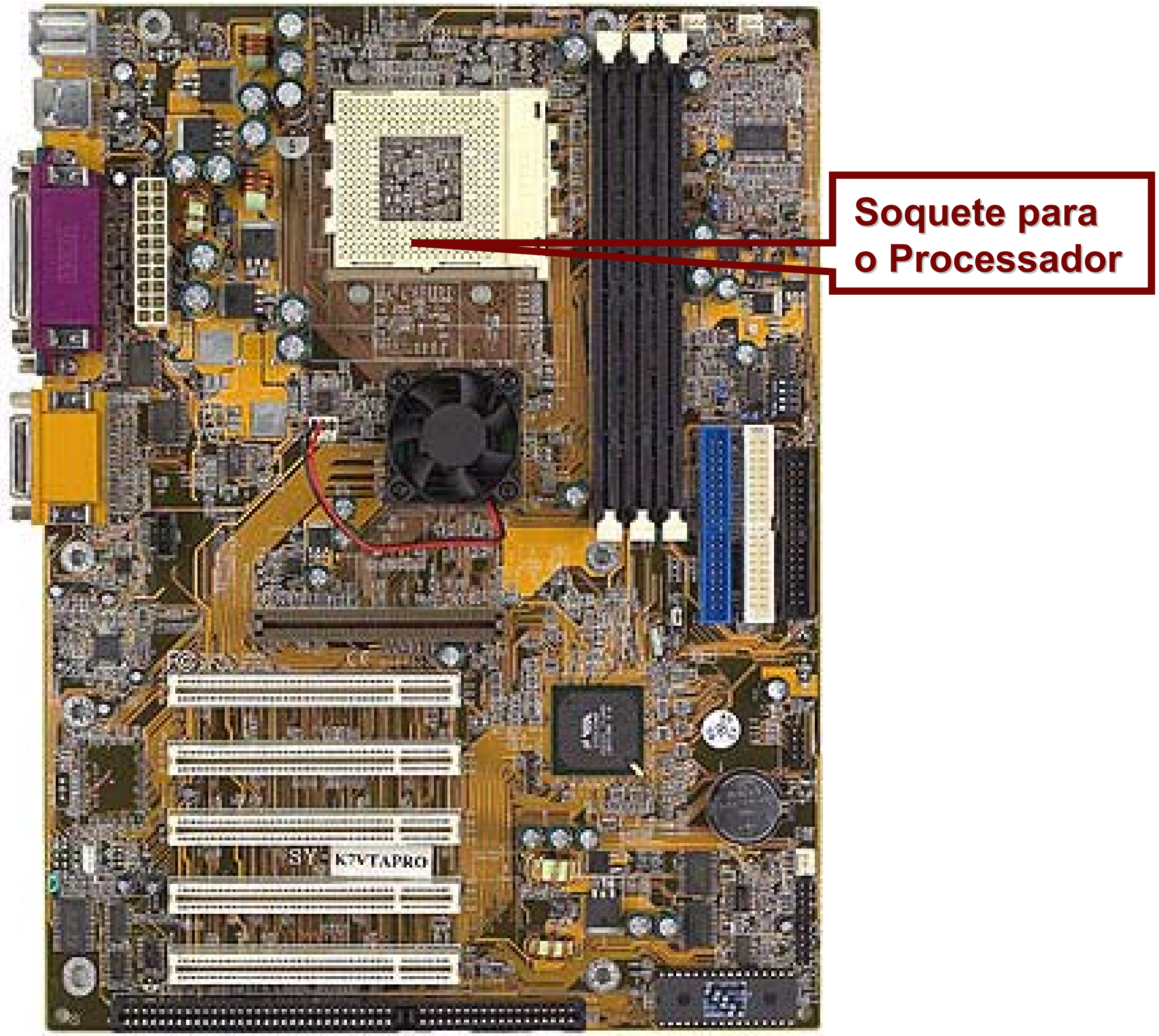
- Hospedam:
  - CPU
  - Conector ou slot de memória
  - Conectores ou slots de expansão
    - ISA (quase obsoleto)
    - VLB (obsoleto)
    - PCI
    - AGP
- Possuem barramentos (caminhos) entre os diversos dispositivos eletrônicos

Placa Mãe

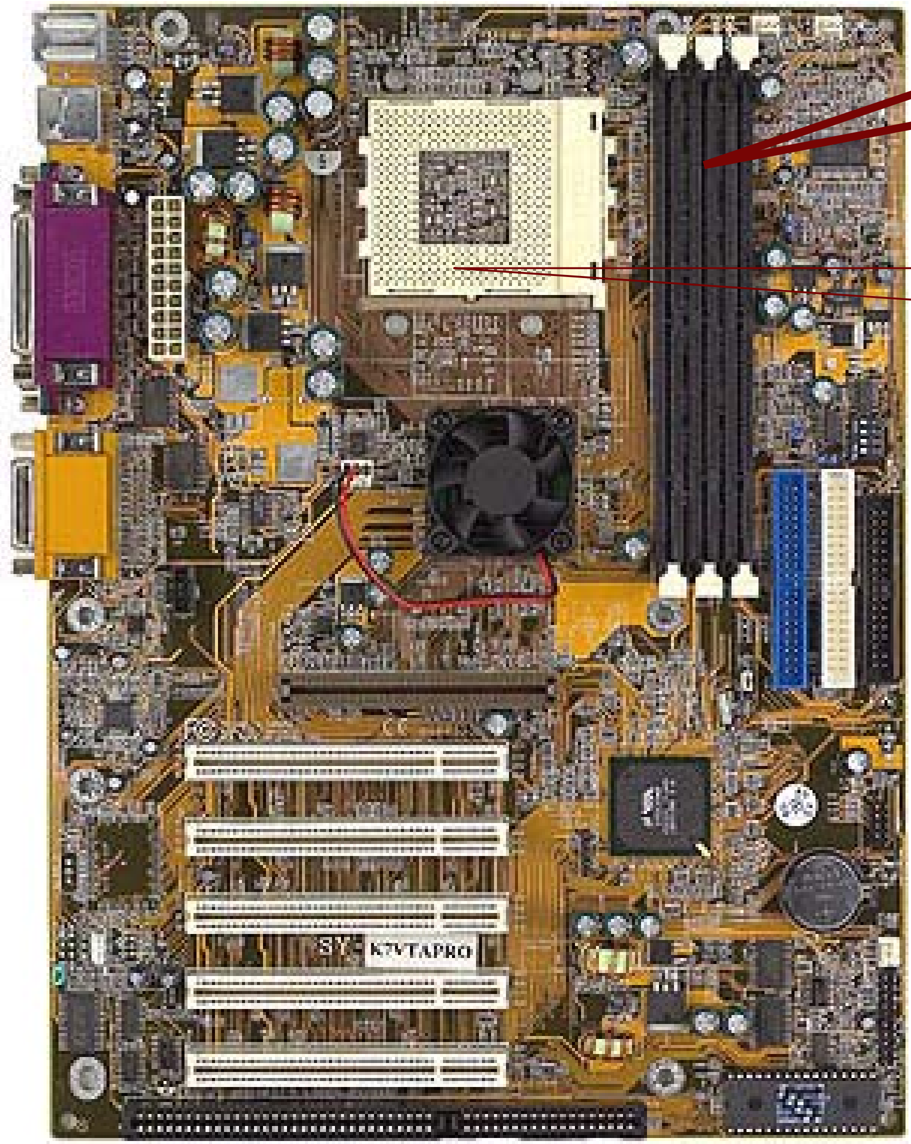


cederj

Placa Mãe



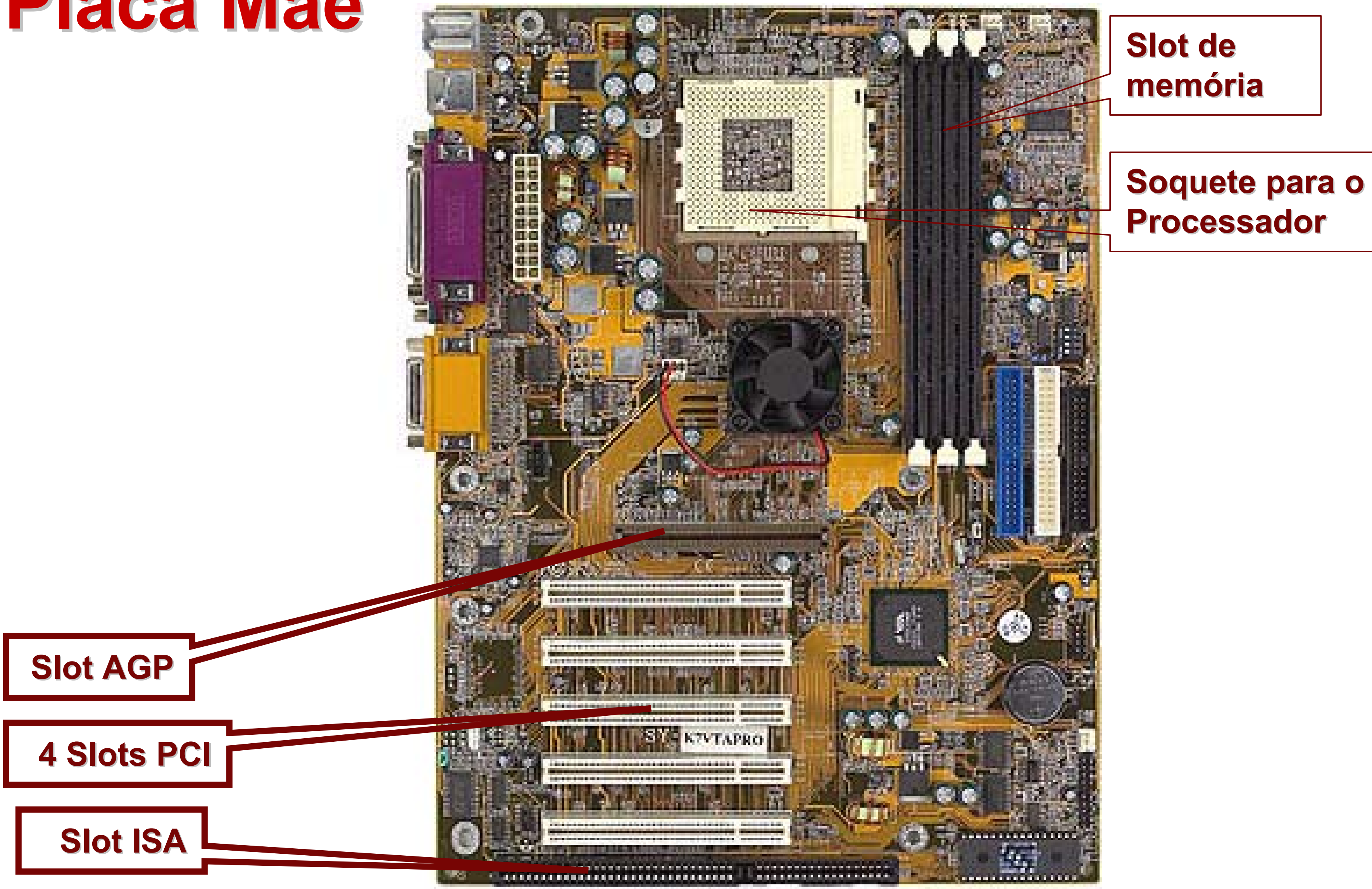
Placa Mãe



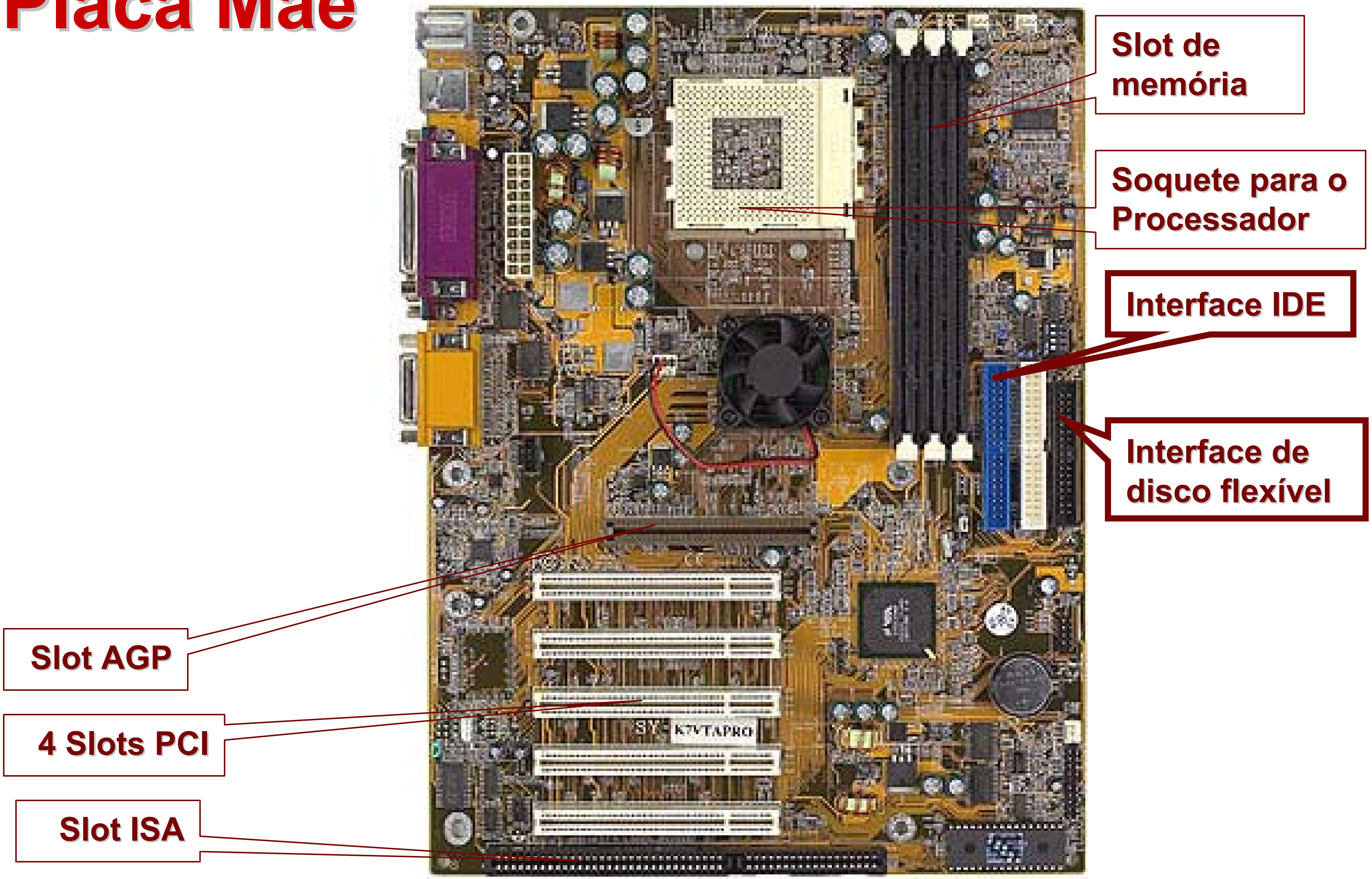
Slot de memória

Soquete para o Processador

Placa Mãe

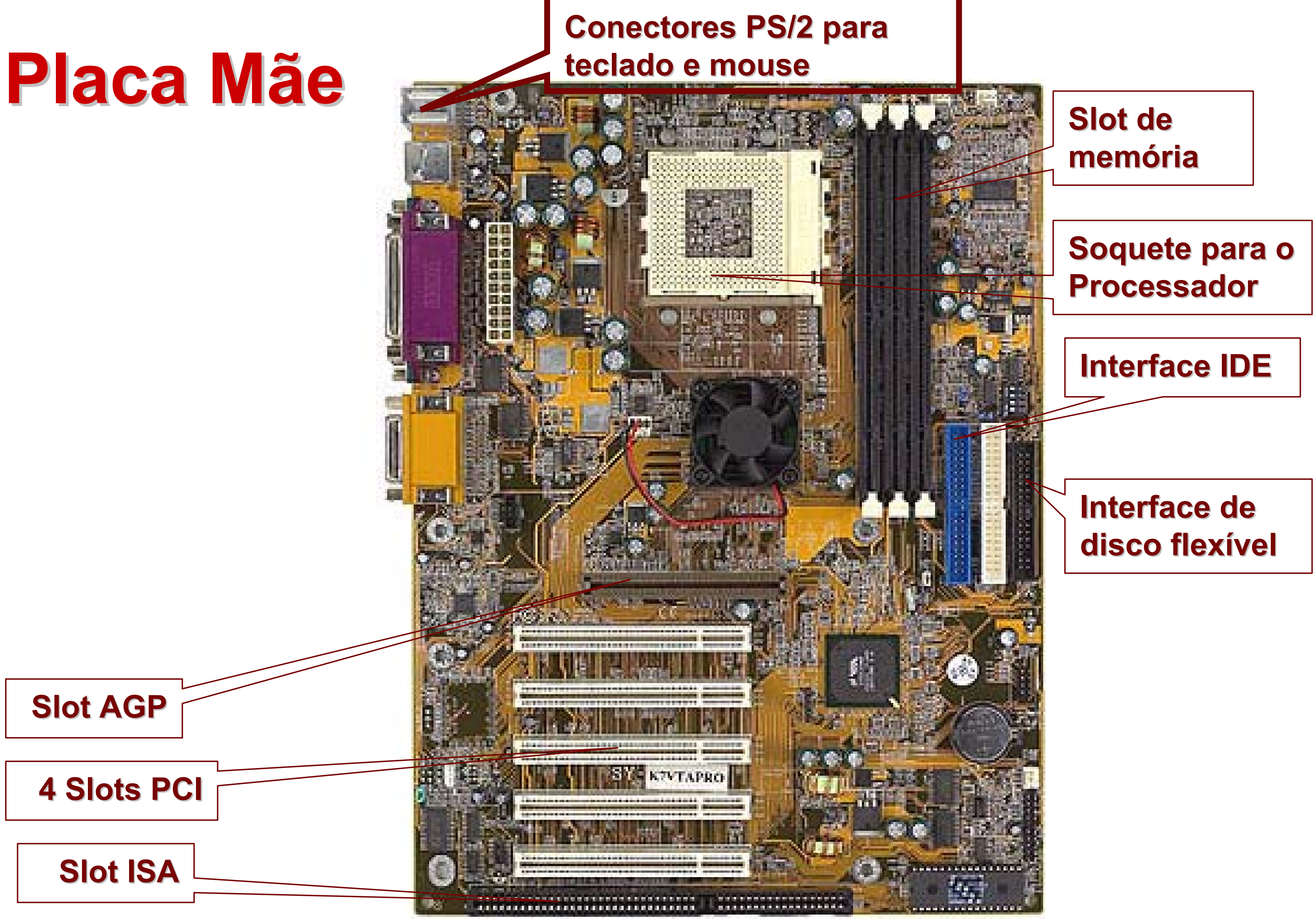

Slot de memória
Soquete para o Processador
Slot AGP
4 Slots PCI
Slot ISA
K7VTAPRO

Placa Mãe

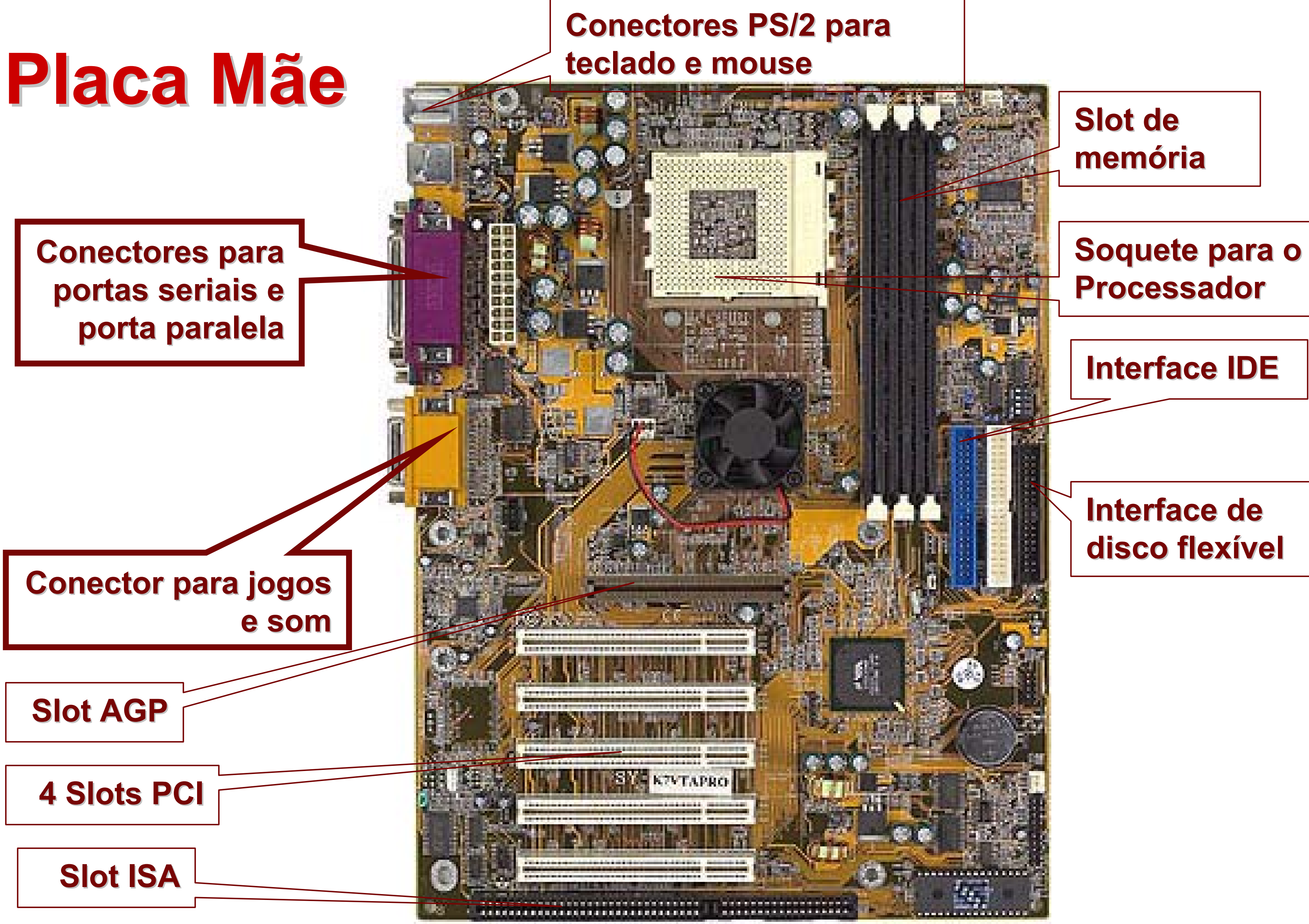


cederj

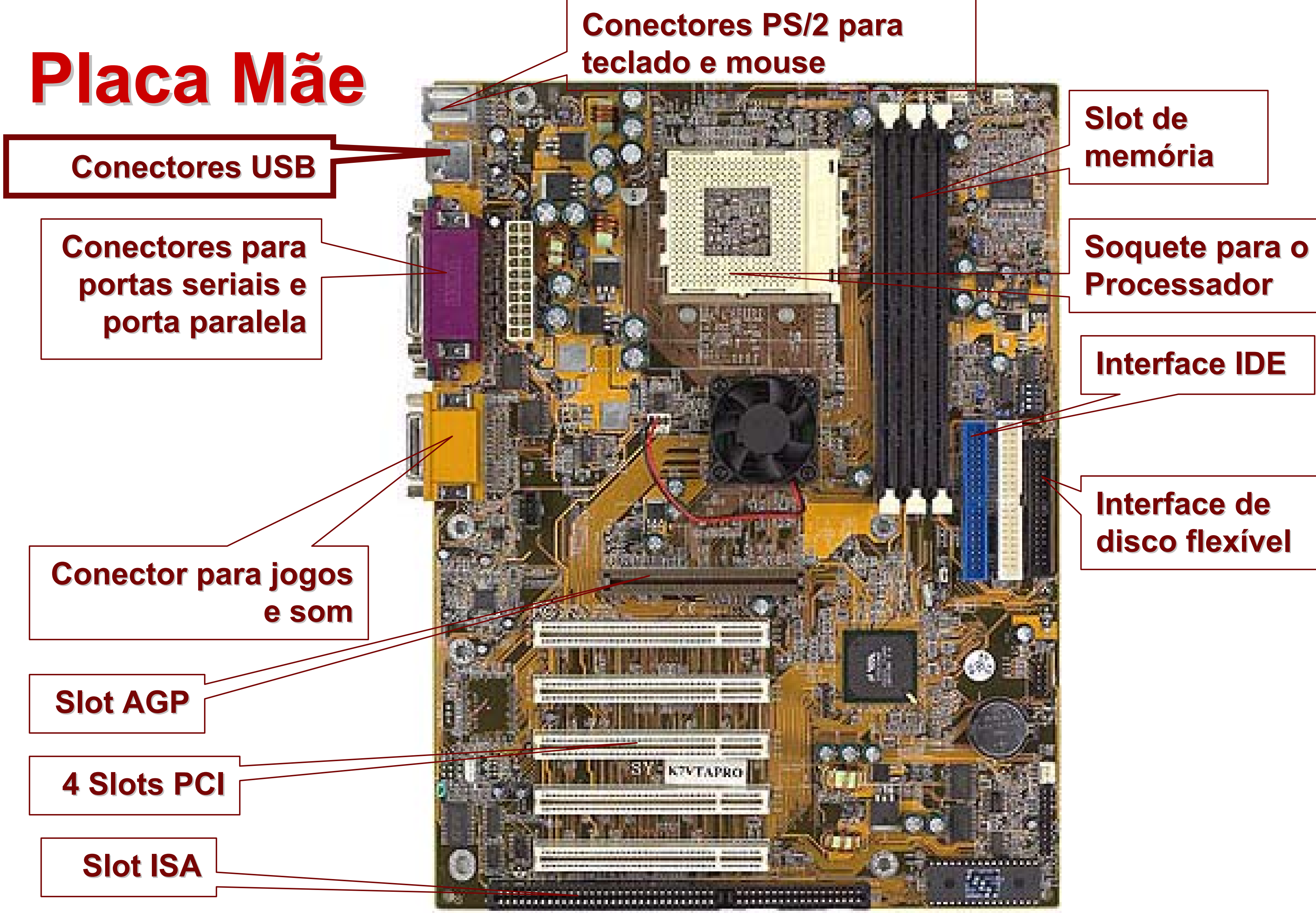
Placa Mãe


Conectores PS/2 para teclado e mouse
Slot de memória
Soquete para o Processador
Interface IDE
Interface de disco flexível
Slot AGP
4 Slots PCI
Slot ISA
K7VTAPRO

Placa Mãe

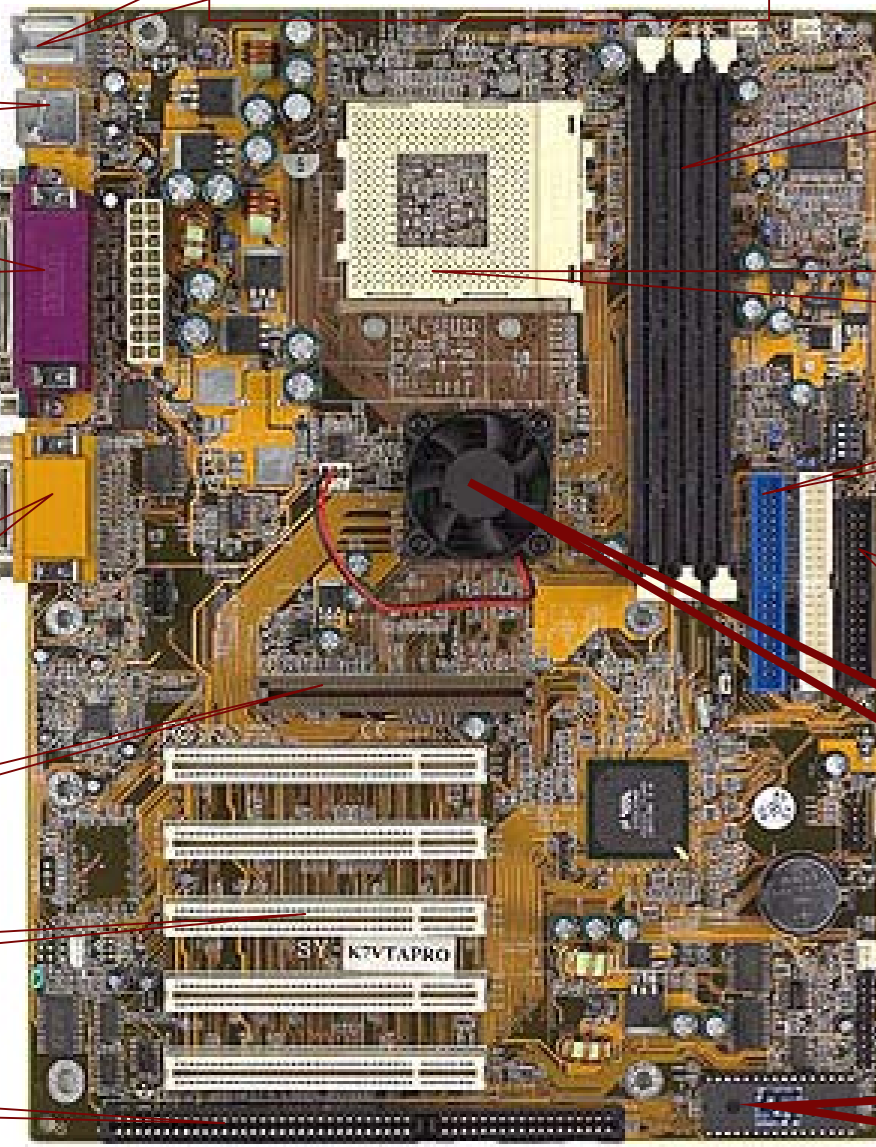

Conectores PS/2 para teclado e mouse
Slot de memória
Soquete para o Processador
Interface IDE
Interface de disco flexível
Conectores para portas seriais e porta paralela
Conector para jogos e som
Slot AGP
4 Slots PCI
Slot ISA

Placa Mãe

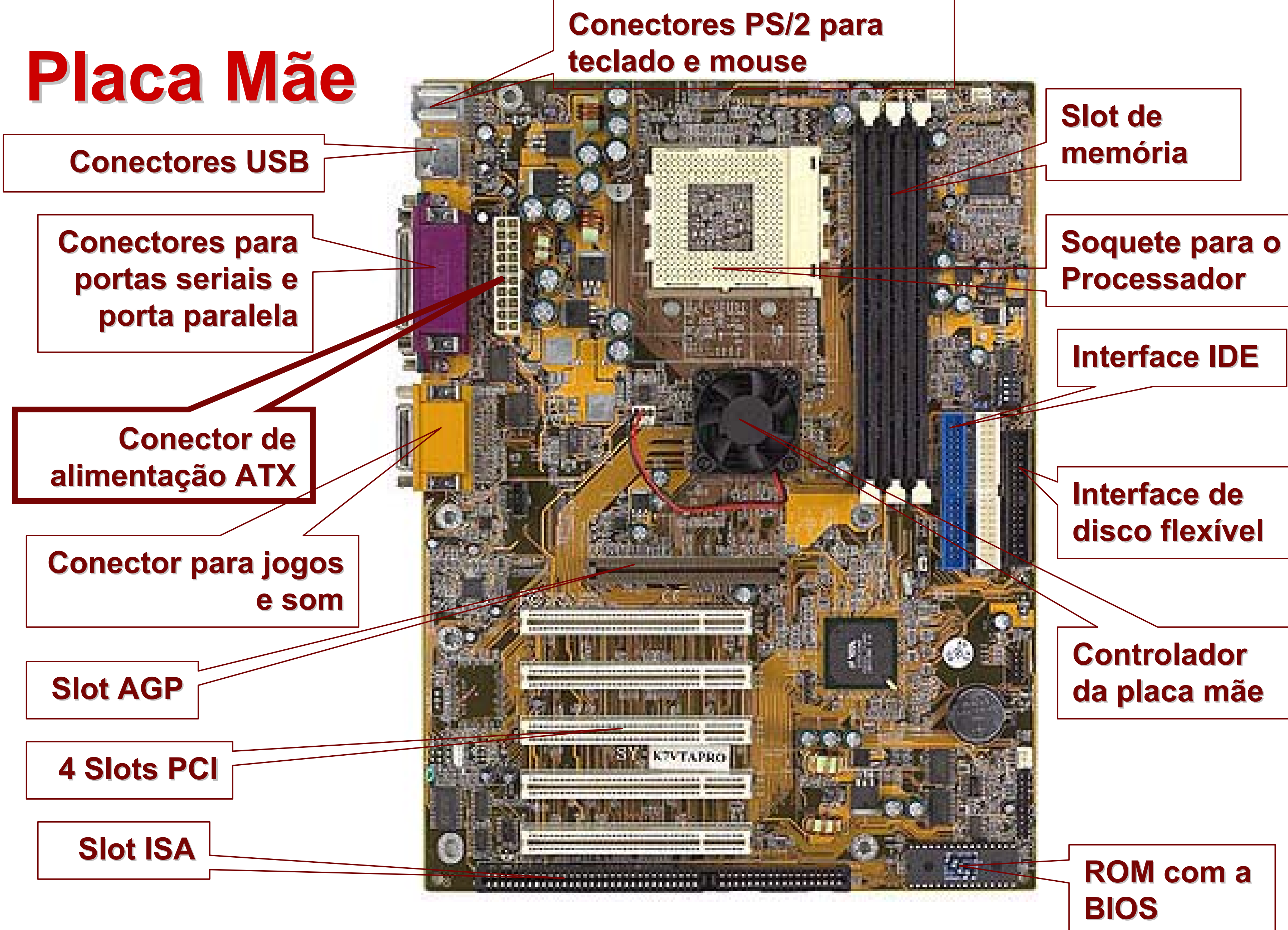


Conectores PS/2 para teclado e mouse
Conectores USB
Slot de memória
Conectores para portas seriais e porta paralela
Soquete para o Processador
Interface IDE
Interface de disco flexível
Conector para jogos e som
Slot AGP
4 Slots PCI
Slot ISA
K7VTAPRO

Placa Mãe

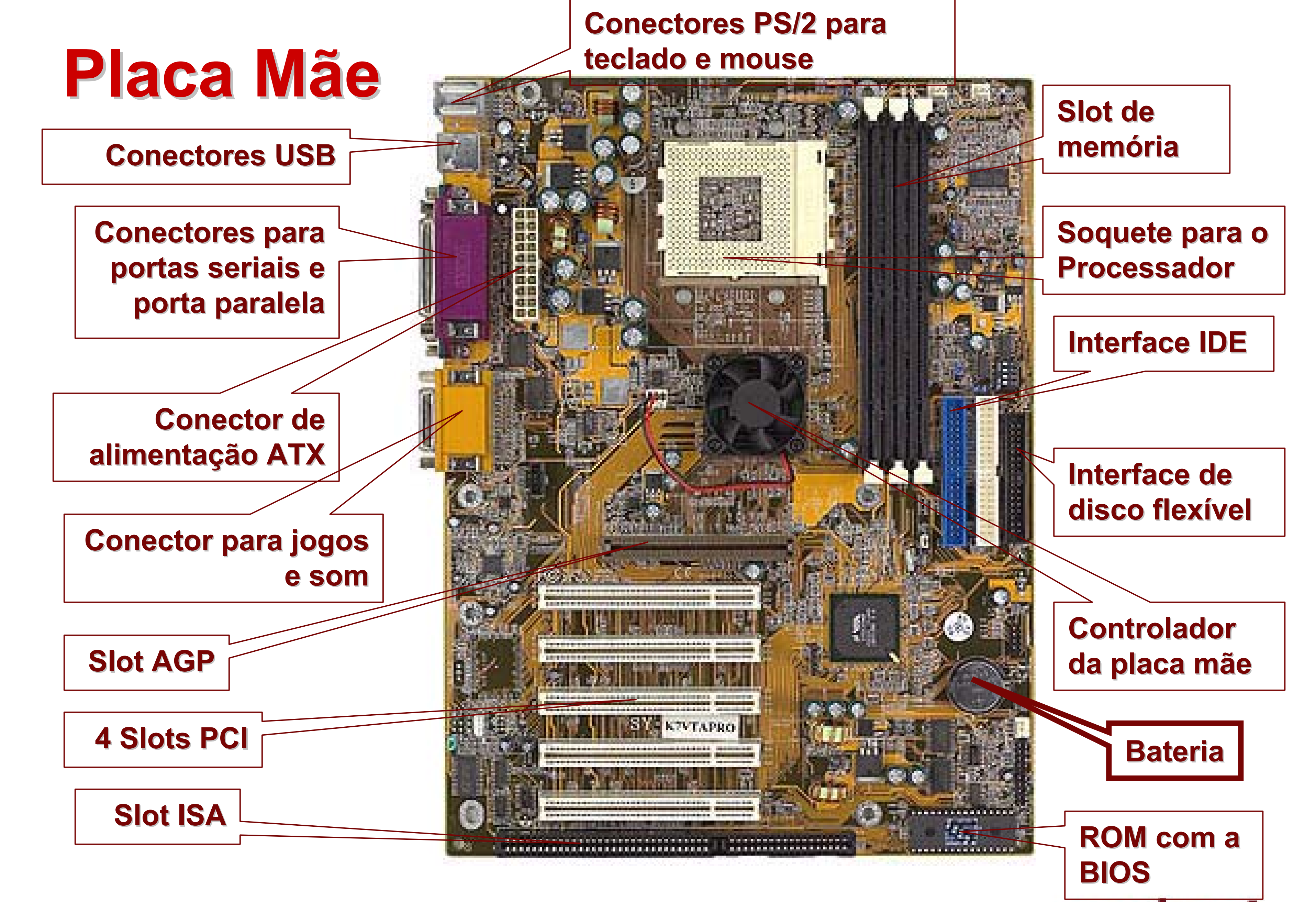

Conectores PS/2 para teclado e mouse
Conectores USB
Conectores para portas seriais e porta paralela
Conector para jogos e som
Slot AGP
4 Slots PCI
Slot ISA
Slot de memória
Soquete para o Processador
Interface IDE
Interface de disco flexível
Controlador da placa mãe
ROM com a BIOS

Placa Mãe



cederj

Placa Mãe


Conectores PS/2 para teclado e mouse
Conectores USB
Slot de memória
Conectores para portas seriais e porta paralela
Soquete para o Processador
Interface IDE
Conector de alimentação ATX
Interface de disco flexível
Conector para jogos e som
Controlador da placa mãe
Slot AGP
Bateria
4 Slots PCI
Slot ISA
ROM com a BIOS

Próximas Aulas

- Aula 2 - Exercícios de fixação
- Aula 3 - Especificando e montado um computador

Introdução à Informática

Alexandre Meslin
(meslin@nce.ufrj.br)

Aula Anterior

- Componentes de um computador
- Exercícios de fixação

Esta Aula

- Especificando um Computador
- Montando um Computador

Placa Mãe e Processador

- Devem ser especificados em conjunto
- Calcular relação custo x benefício
- Procurar alternativas no mercado
- Informar-se sobre bons fabricantes
  - Os melhores fabricantes geralmente possuem site na Internet
  - Verificar se o site possui informações de apoio a problemas

Placa Mãe e Processador

- Devem ser diretamente compatíveis
- Verificar no manual qual o processador necessário e/ou suportado pela placa mãe
- Escolher um conjunto de dissipador/ventoinha adequado ao processador

Gabinete

- O gabinete deve ser suficiente para armazenar a placa mãe e todos os dispositivos pretendidos
- Cuidado principalmente com o espaço entre a fonte de alimentação e o suporte da placa mãe
  - Permitir instalação do processador com dissipador de calor e ventoinha
  - Permitir que o ar circule livremente

Gabinete

- Torre

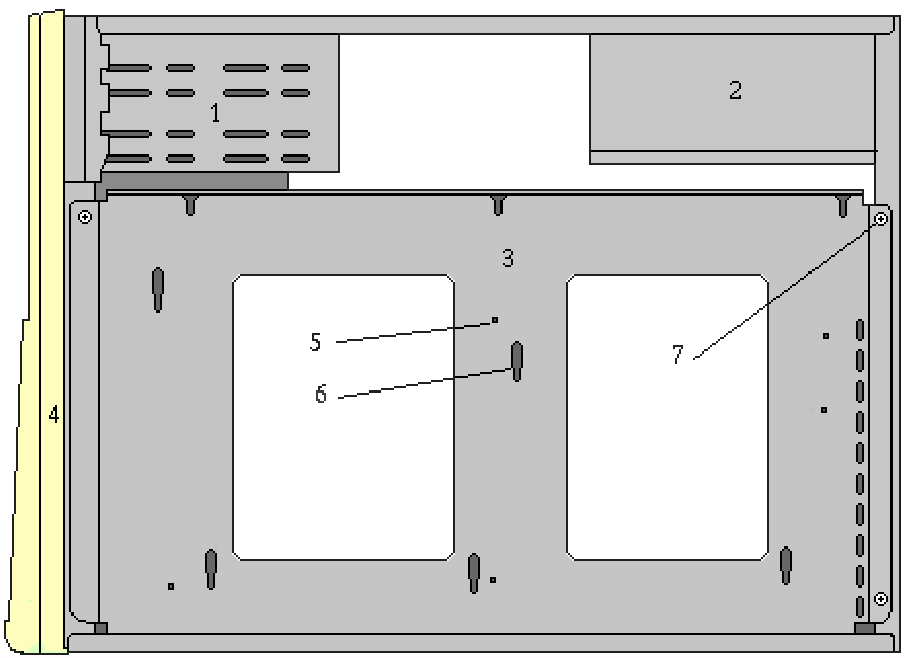
- Mesa (desktop)

Instalação da Placa Mãe

- Leia atentamente o manual de instruções
- Siga todos os passos
  - ❖ Configura a placa mãe
  - ❖ Instale a memória
  - ❖ Fixe a placa mãe no gabinete

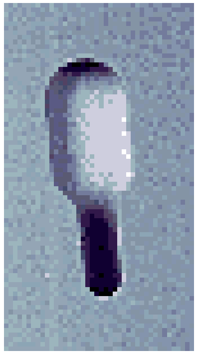
Instalação da Placa Mãe

1. Baias para discos
2. Fonte de alimentação
3. Base para montagem da placa mãe
4. Painel frontal
5. Orifício de fixação com rosca
6. Orifício de fixação com espaçador
7. Parafuso de fixação da base



Instalação da Placa Mãe

- Espaçador plástico
- Rasgo da caixa para o espaçador plástico
- Espaçador com rosca (prender com um parafuso) - metálico



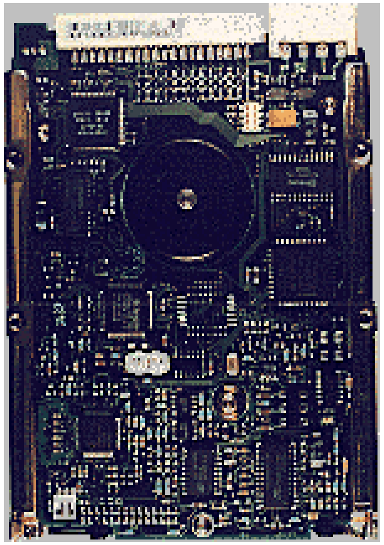
cederj

Instalação da Placa Mãe

- Conecte os fios:
  - ❖ Chaves
    - reset
    - turbo (se for o caso)
  - ❖ Leds
    - power
    - display (se for o caso)
    - HD
    - Turbo (se for o caso)
  - ❖ Auto Falante

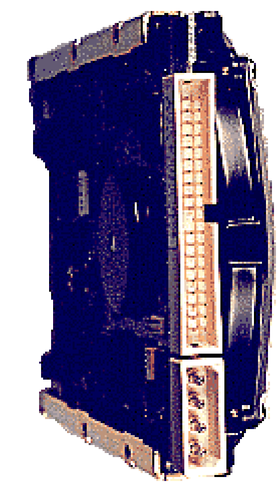
Escolha do Hard Disk (HD)

- Fabricante
- Interface
  - IDE
  - SCSI
- Capacidade
- Tamanho Físico
  - 3½”
  - 5¼”
- Desempenho
  - Tempo de Acesso
  - Taxa de Transferência
  - Velocidade de Rotação



HD IDE - Instalação

- Conector da Fonte
  - ❖ 4 pinos (+12V, +5V, 2 Terras)
  - ❖ Observar os chanfros
- Conector (cabo) IDE
  - ❖ Cabo com 3 conectores de 40 pinos
  - ❖ Pino 1 marcado pelo fio de cor diferente e pintura na placa do HD
  - ❖ Disco C configurado como *master* e D como *slave*



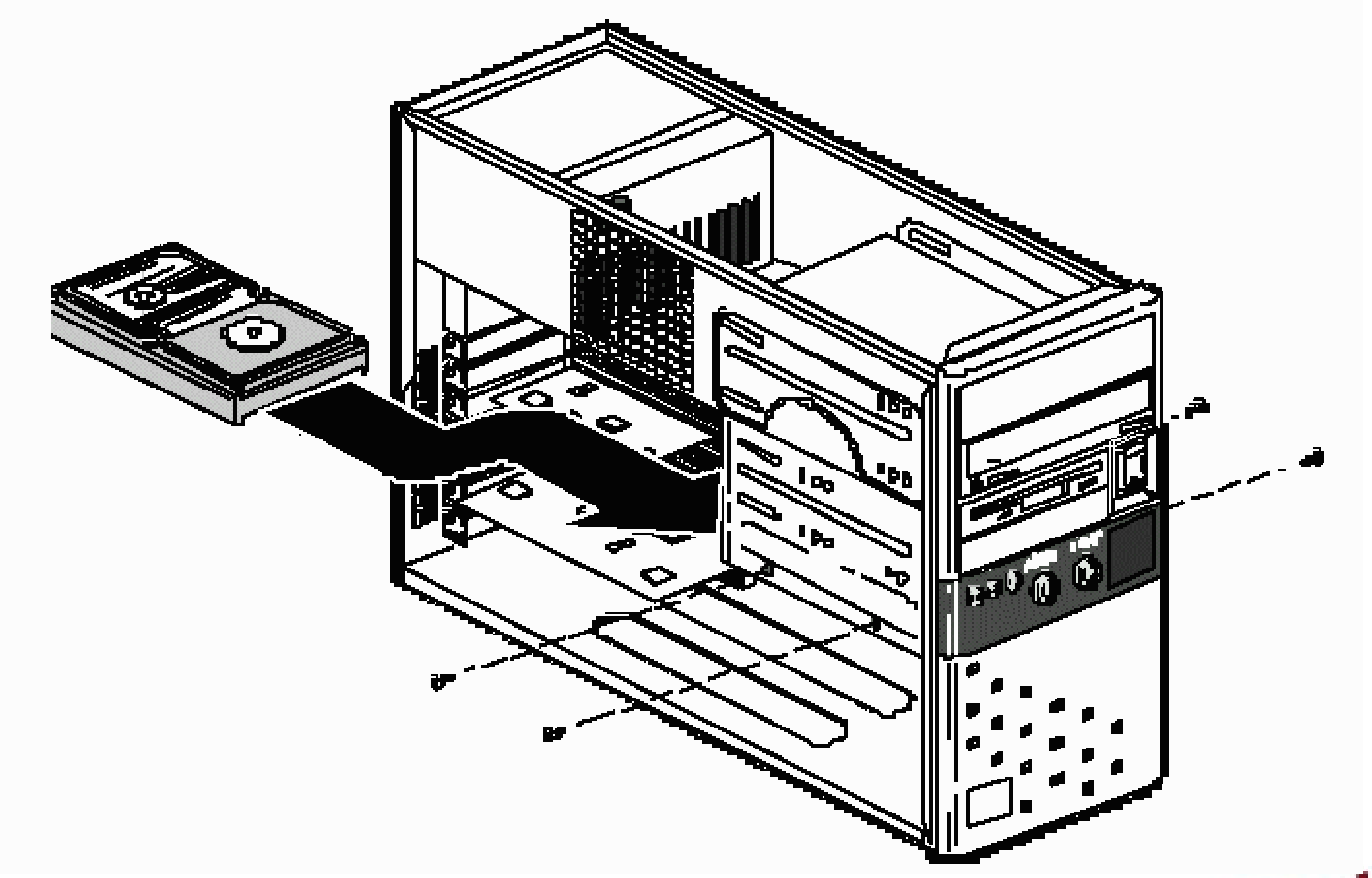
HD IDE - Instalação

- Verifique o pino 1 na serigrafia da placa do HD perto do conector IDE
- Verifique se os 4 parafusos de fixação se adaptam perfeitamente nos furos do HD e se não encostam na placa quanto totalmente apertados

HD IDE - Instalação

- Configure o HD para MASTER ou SLAVE
- Conecte o cabo IDE (40 vias) no HD e na interface IDE
- Conecte o cabo de força (4 fios)
- Prenda o HD através dos 4 parafusos
- Coloque o HD na sua baia

HD IDE – Instalação



Outros Dispositivos IDE

- Instalação e configuração semelhante ao HD

- CD-ROM
- Gravador de CD
- DVD
- Zip Drive

Acionador de Disco Flexível

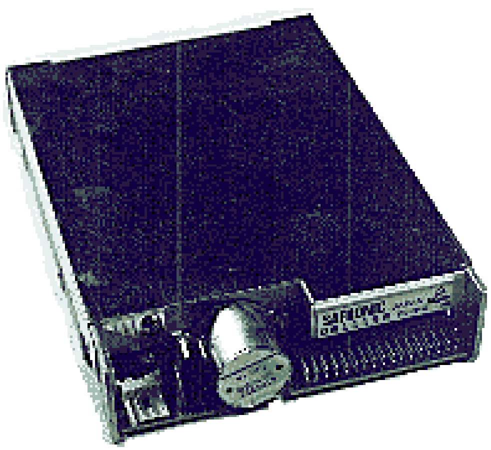
- Tamanho - 3½"
- Capacidade 1,44Mbytes

FDD - Instalação

- Conector da Fonte
  - ❖4 pinos (+12, +5, 2 Terras)
  - ❖Observar os chanfros
- Conector (cabo) para FDD
  - ❖cabo com 3, 4 ou 5 conectores de 34 pinos
  - ❖pino 1 marcado pelo fim de cor diferente no cabo e pintura na placa do FD
  - ❖Disco A colocado na ponta do cabo (depois da inversão)

FDD - Instalação

- Conectores
- Cabo de FDD
- Cabo de Alimentação



FDD - Instalação

- Verifique o pino 1 na serigrafia da placa do FDD perto do conector de 34 vias
- Verifique se os 4 parafusos de fixação se adaptam perfeitamente nos furos do FDD e se não encostam na placa quanto totalmente apertados
- Remova a tampa cega da baia desejada
- Coloque o FDD na sua baia

FDD - Instalação

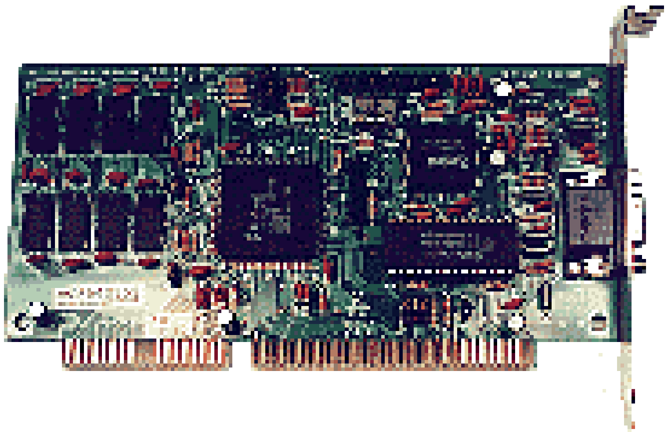
- Prenda o FDD através dos 4 parafusos
- Conecte o cabo de 34 vias - o disco da ponta será o drive A e o do meio o B - no FDD e na interface do FDD
- Conecte o cabo de força (4 fios)

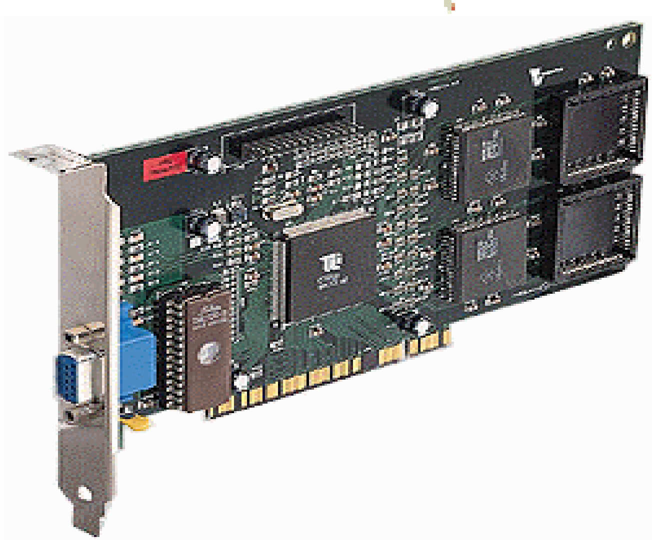
Placa de Vídeo

- Conecta-se no barramento da placa mãe
- Escolha uma placa de vídeo compatível com o monitor e com o barramento
- Possuem conector do barramento do PC e um conector de vídeo (15 ou 9 pinos fêmea)

Placa de Vídeo - Reconhecimento

- Conector fêmea do tipo DB15 (três fileiras) ou conector fêmea do tipo DB9 (duas fileiras)
- Conector no padrão da placa mãe
- Conjunto de memória





Placa de Vídeo – Características

- Padrão (VGA ou SVGA)
- Resolução máxima
- Número de cores
- Tamanho da Memória
- Processamento
- Conector (AGP ou PCI)

Placa de Vídeo – Instalação

- Retire a aba trazeira do gabinete
- Alinhe os contados da placa em relação ao slot
- Posicione a placa no slot suavemente
- Verifique se o conector da placa entrou totalmente
- Conecte o cabo de vídeo

Placa de Som

- Conecta-se ao barramento da placa mãe
- Atualmente são todas PCI
- Converte informação digital em analógica (sonora)
- Podem ter de 2 até 6 canais
- Possibilidade de entrada para
    - CD/DVD
    - Auxiliar
    - Microfone
    - Etc

Placa de Som – Características Importantes

- Número de bits usados na conversão
  - ❖ Digital/analógico
  - ❖ Analógico/digital
- Relação sinal ruído
- Decodificador MP3

cederj

Placa de Som – Instalação

- Retire a aba trazeira do gabinete
- Alinhe os contados da placa em relação ao slot
- Posicione a placa no slot suavemente
- Verifique se o conector da placa entrou totalmente

Placa de Rede

- Conecta-se ao barramento da placa mãe
- Deve possuir conector apropriado ao tipo de rede a qual se destina
- Interfaces
  - PCI
  - RJ-45
- Taxa de transmissão
  - 10 Mbps
  - 10/100 Mbps

Placa de Rede – Instalação

- Retire a aba trazeira do gabinete
- Alinhe os contados da placa em relação ao slot
- Posicione a placa no slot suavemente
- Verifique se o conector da placa entrou totalmente

Interface de Fax/Modem

- Conecta o micro à linha telefônica
- Utilizavam interface ISA
- Placas atuais com interface PCI
- Existem modems externos RS232 e USB
- Interface 2 x 1
  - Fax
  - Modem
- Hardmodem x Softmodem

Interface de Fax/Modem – Instalação

- Retire a aba trazeira do gabinete
- Alinhe os contados da placa em relação ao slot
- Posicione a placa no slot suavemente
- Verifique se o conector da placa entrou totalmente
- Conecte o cabo da linha telefônica
- Algumas placas permitem conexão de um aparelho telefônico

Próxima Aula

- Exercícios de fixação

Introdução à Informática

Alexandre Meslin
(meslin@nce.ufrj.br)

cederj

Organização da Memória

- Conceito de hierarquia de memória
- Memória principal e memórias secundárias
- Projeto lógico da memória principal
- Memórias cache
- Memória virtual

Conceito de Hierarquia de Memória

- Memória no computador
  - **Registradores**
  - **Memória principal**
  - **Memória cache**
  - Memória ROM
  - Discos magnéticos
  - Discos ópticos
  - Fitas magnéticas

Registradores

- É a memória mais rápida
- Montada dentro da CPU
- Local onde as operações aritméticas são calculadas
- Conjuntos de apenas algumas dezenas

Memória Principal

- Construída utilizando memória dinâmica
- Memória de leitura e escrita
- DRAM
- Mais lenta e mais barata que a memória cache
- Computadores atuais possuem pelo menos 64 megabytes
- Tempo de acesso unitário de 60 ns e em modo rajada de 7 ns
- O seu conteúdo é perdido quando a energia é desligada

Problemas

- Configuração de hardware
  - Processador de 1 GHz
  - Unidade aritmética de 32 bits (4 bytes)
  - Tempo de acesso à memória:
    - Unitário: 60 ns
    - Rajada: 7 ns (PC133)
- Processador necessita de dados a cada 1 ns
  - Período é o inverso da freqüência
  - 1 ns = 1/1 GHz

Valores

- Memória de 8 bits
  - ❖ 4 acessos necessários
  - ❖ 1o acesso em 60 ns
  - ❖ 2o, 3o e 4o acesso a cada 7 ns
  - ❖ Tempo total = 60 + 3*7 = 81 ns
  - ❖ Memória 80 vezes mais lenta que o processador

cederj

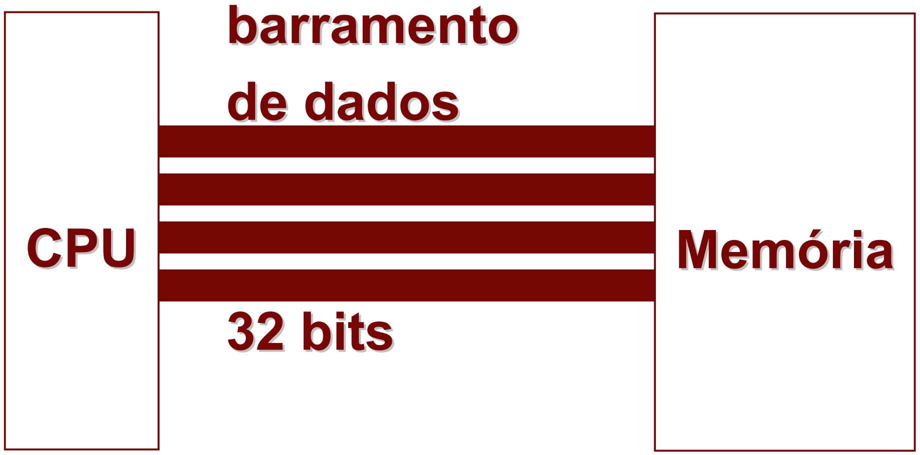
Valores

- Memória de 16 bits
  - ❖ 2 acessos necessários
  - ❖ 1o acesso em 60 ns
  - ❖ 2o acesso em 7 ns
  - ❖ Tempo total = 60 + 7 = 67 ns
  - ❖ Memória 67 vezes mais lenta que o processador

cederj

Valores

- Memória de 32 bits
  - ❖ 1 único acesso
  - ❖ acesso em 60 ns
  - ❖ Tempo total = 60 ns
  - ❖ Memória 60 vezes mais lenta que o processador



Memória Cache

- Construída utilizando memória estática
- Memória de leitura e escrita
- SRAM
- Extremamente rápida
- Pouca capacidade
- Opera em velocidade perto da velocidade do processador
  - Igual ou metade
- Memória de acesso aleatório

Memória Cache

- Nível 1 → construída junto ao processador
- Nível 2 → fora do processador (na placa mãe)
- A maior parte dos PC’s contém:
  - Alguns quilobytes de nível 1
  - Poucos megabytes de nível 2
- O tempo de acesso é menor do que 6ns
- Memória muito cara
- O seu conteúdo é perdido quando a energia é desligada

Termos Comuns

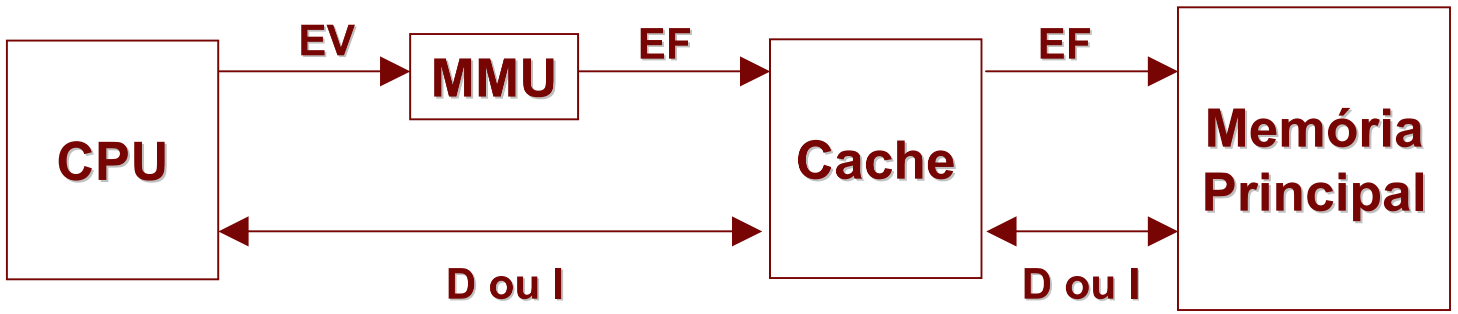
- Referências pelo processador a dados armazenados em cache são chamados de ACERTO (HIT)
  - Taxa de acerto normalmente maior que 95%
- Referências pelo processador a dados não armazenados em cache são chamados de FALHA (MISS)
- Cache normalmente busca uma linha
  - Localidade espacial
- Descritor (tag)
  - Dados, endereços, validade, etc.

Funcionamento

- Tentativa de aproximar o tempo de acesso à memória do tempo de acesso da CPU
- Dependente de diversos fatores
  - ❖ Arquitetura do computador
  - ❖ Comportamento dos programas
  - ❖ Tamanho e organização da cache
- Transparente para o programa/programador

Funcionamento

- Pedido de memória verificado antes na cache
  - ❖ Se estiver presente, a cache fornece/recebe informação da CPU (acerto ou hit)
  - ❖ Caso contrário, o pedido é enviado para a memória principal (falha ou miss)



Princípios da Cache

- Localidade espacial
  - Grande probabilidade de acessos à locais vizinhos
- Localidade temporal
  - Grande probabilidade de se acessar regiões que foram recentemente acessadas
- Seqüencialidade
  - Se uma referência é feita ao endereço X, existe grande probabilidade de haver referência ao endereço X+1

Memória Cache

Organização do Cache

- Mapeamento direto
  - Direct mapped
- Conjunto associativo
  - Set associative
- Totalmente associativo
  - Fully associative

Curiosidades

- Supondo memória cache com:
  - 256 kbytes
  - 32 bytes/bloco
  - 8 linhas
  - Barramento de endereço de 32 bits
  - Capacidade de endereçar até 4 Gbytes
  - Mapeamento direto

Cache com Mapeamento Direto

- Curiosidades
  - Os bytes 0, 1, 2 até o byte 31 estão na primeira linha da cache
  - Os bytes 32, 33, 34 até o byte 63 estão na segunda linha da cache
  - Os bytes 64, 65, 66 até o byte 127 estão na terceira linha da cache

Cache com Mapeamento Direto

- Convertendo os números para binário, observa-se:
  - Os 5 bits menos significativos dos elementos que pertencem à mesma linha são diferentes
  - Os outros bits são todos iguais.

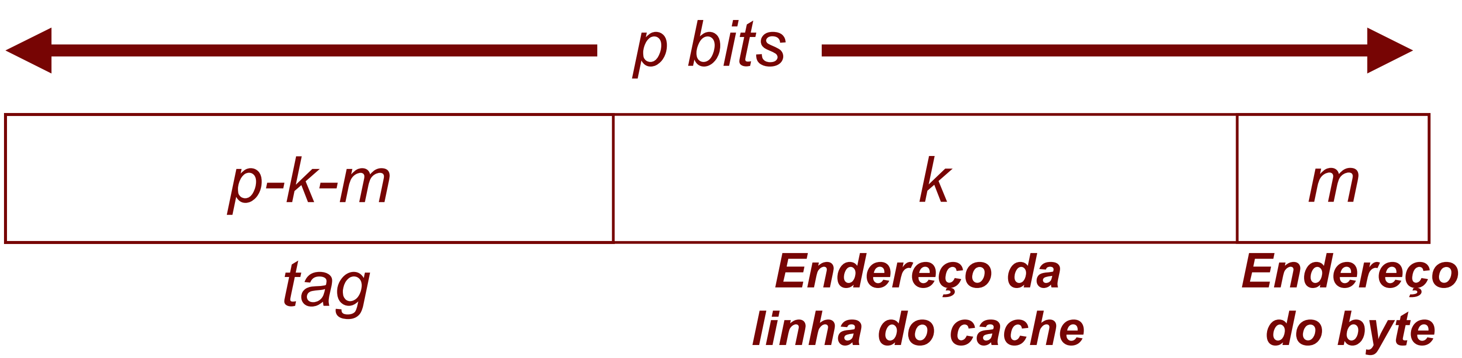
Cache com Mapeamento Direto

- Como o cache tem 256 kbytes endereços, convertendo para binário, isto representa um número de 18 bits

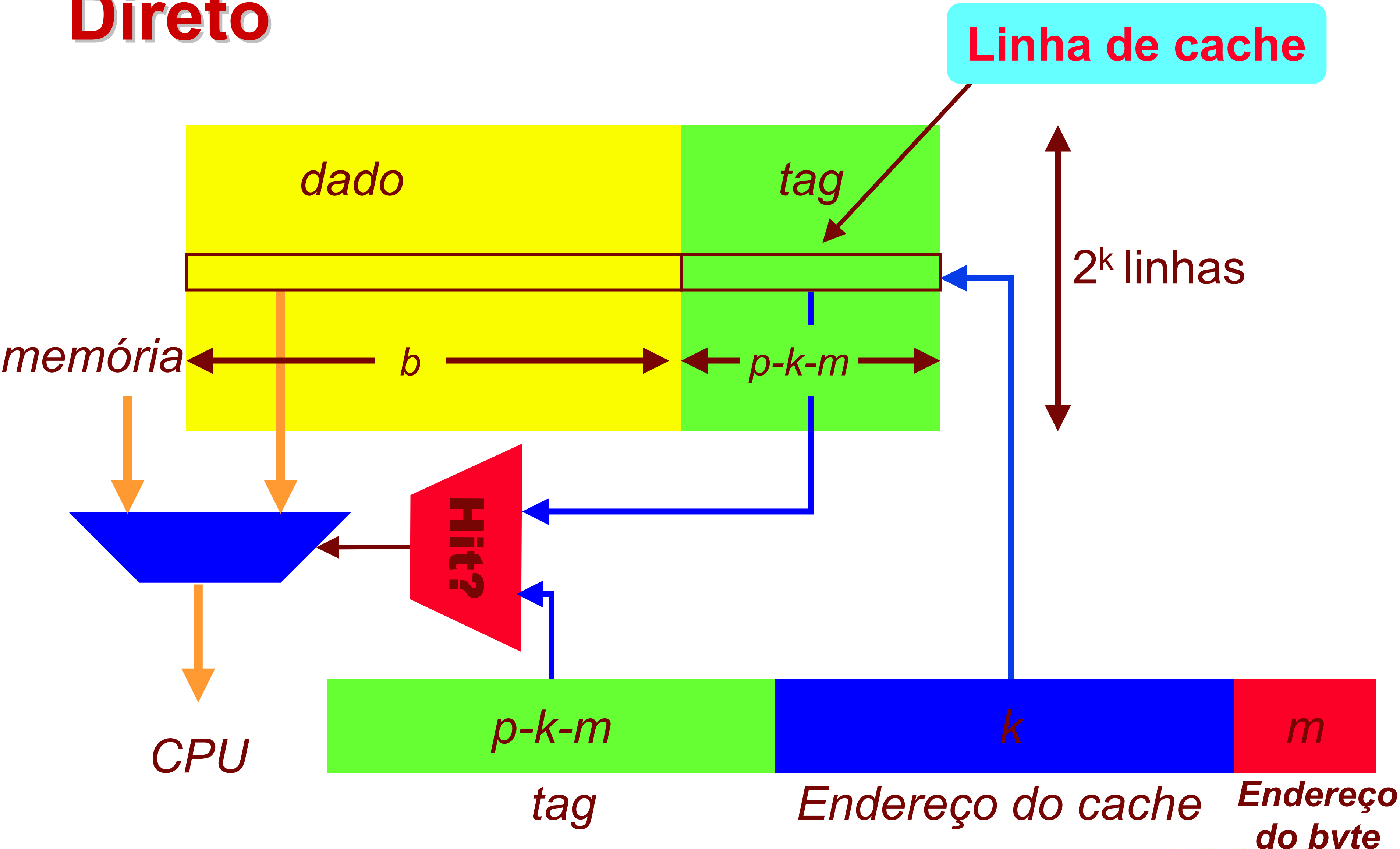
- $2^{18} = 262144 = 256$ kbytes

Cache com Mapeamento Direto

- Assumindo que:
  - Memória cache tem 2^k linhas
  - Memória cache tem bloco com 2^m bytes
  - p bits de barramento de endereço
- Bits mais baixos utilizados para selecionar a linha da cache
- Bits mais altos usados para comparar o endereço da linha



Cache com Mapeamento Direto



cederj

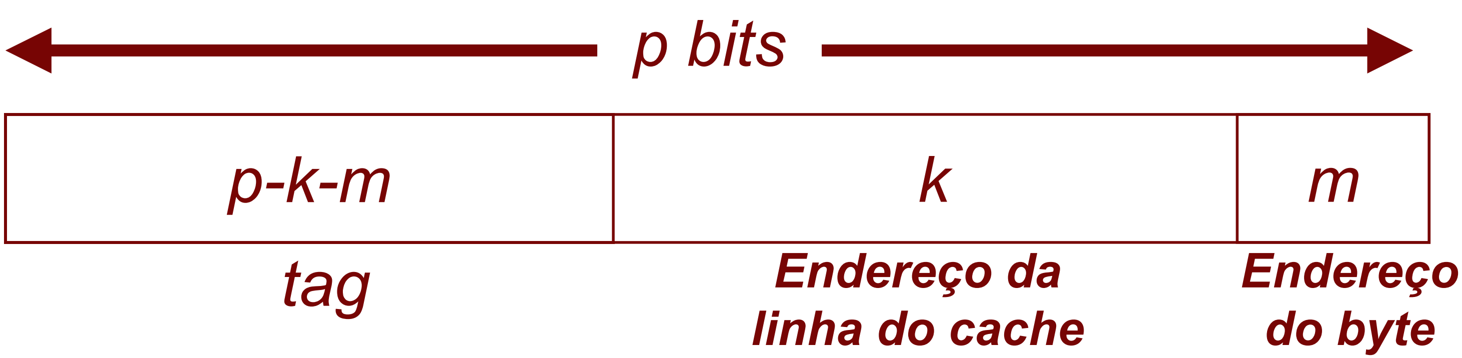
Exemplo

- Supondo memória cache com:
  - ❖256 kbytes
  - ❖32 bytes/bloco
  - ❖8 k linhas
  - ❖Barramento de endereço de 32 bits
  - ❖Mapeamento direto
- Calcular o bloco que será utilizado pelo endereço:

87a6c1b4 (hexadecimal)

Exemplo

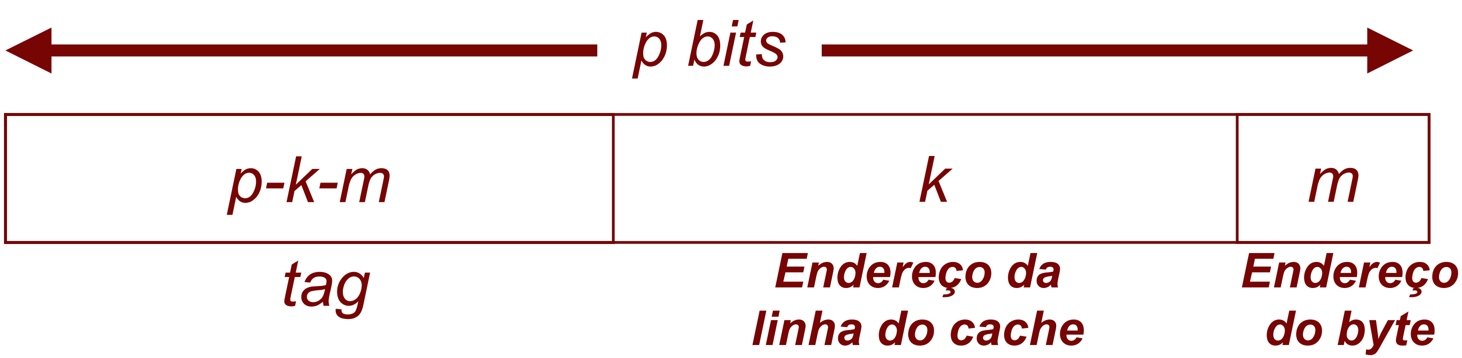
- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 7 a 6 c 1 b 4
    - 1000 0111 1010 0110 1100 0001 1001 0100



cederj

Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 7 a 6 c 1 b 4
    - 1000 0111 1010 0110 1100 0001 10010100



Exemplo

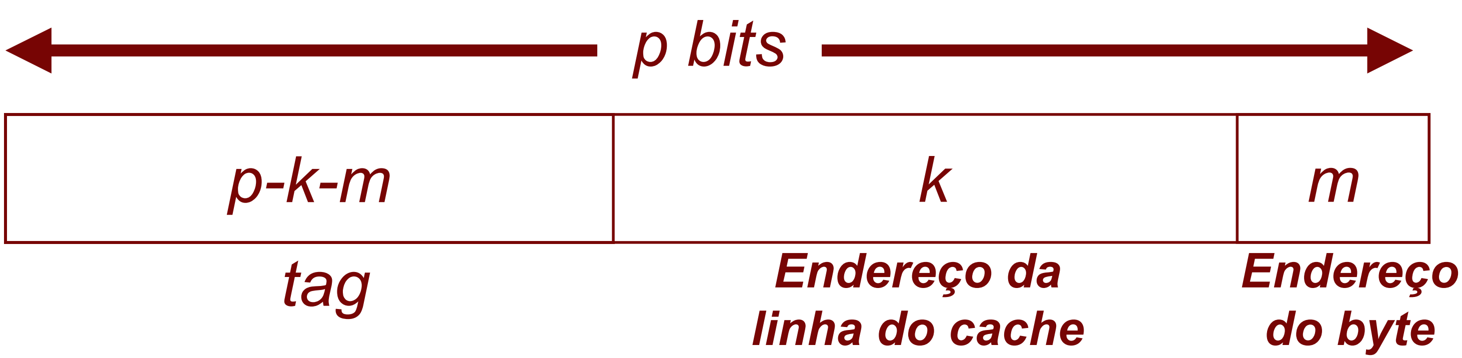
- p = 32 bits
- m = 5 bits (2^5 = 32)
- k = 13 bits (2^{13}=8 k)
- 8 7 a 6 c 1 b 4
  - ❖ 1000 0111 1010 0110 1100 000110010100

← *p bits* →

p-k-m	*k*	*m*
tag	***Endereço da linha do cache***	***Endereço do byte***

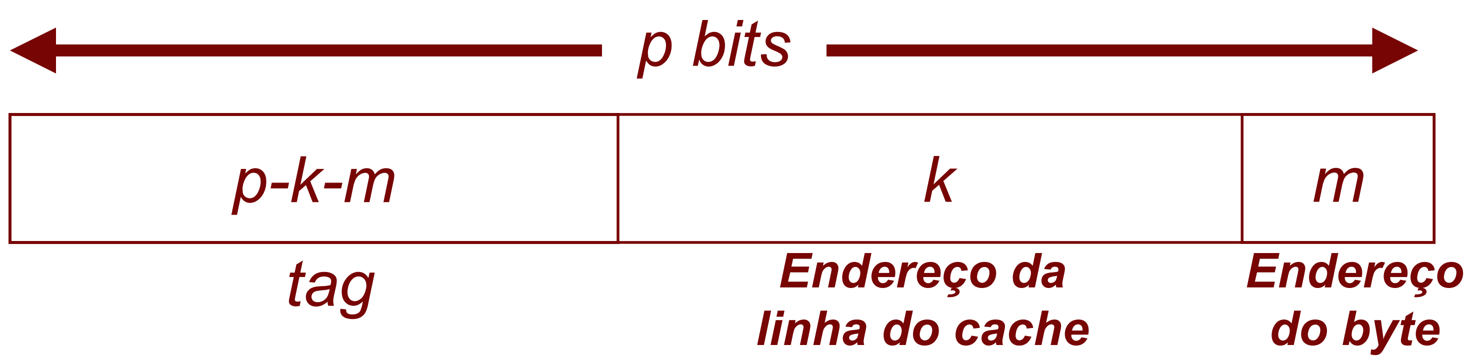
Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 7 a 6 c 1 b 4
    - ❖ 1000 0111 1010 0110 1100000110010100



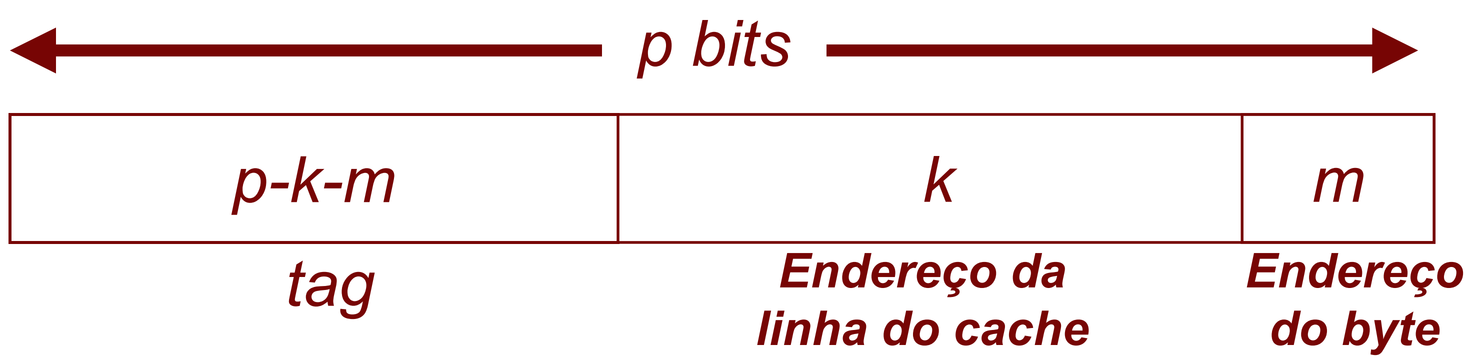
Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 7 a 6 c 1 b 4
  - 1000 0111 1010 01101100000110010100



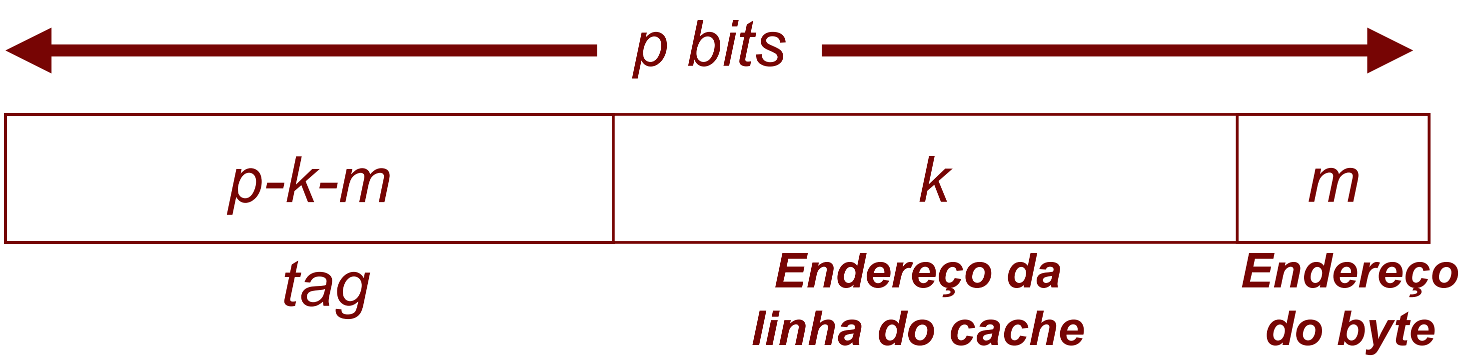
Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 7 a 6 c 1 b 4
  - ❖ 1000 0111 101001101100000110010100



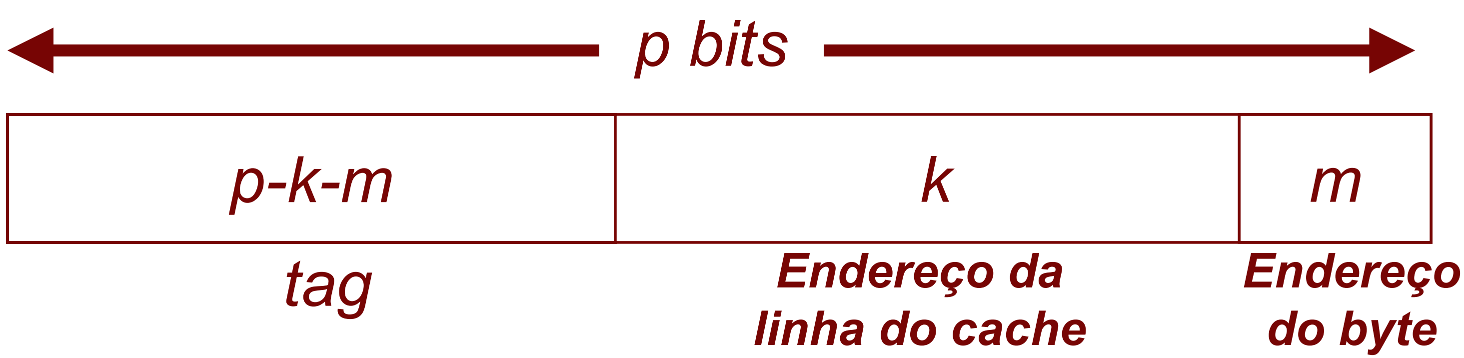
Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- `    8   7   a  6  c   1  b  4`
  - ❖ `1000 0111101001101100000110010100`



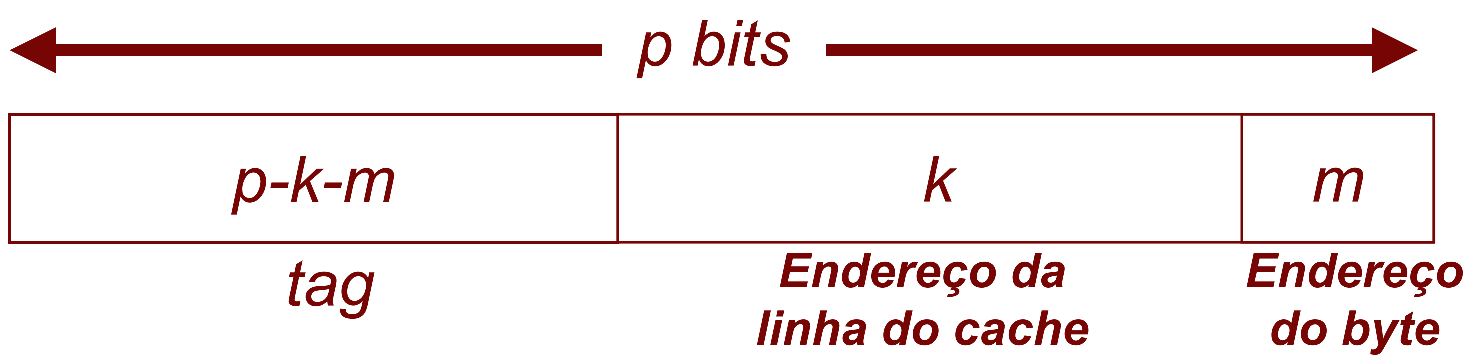
Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 7 a 6 c 1 b 4
  - 10000111101001101100000110010100



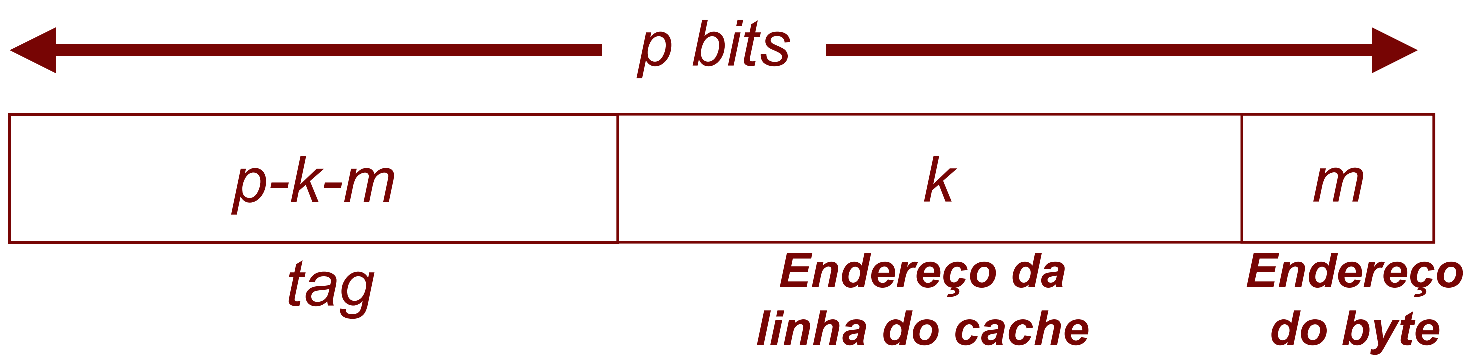
Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 87a6c1b4
  - 10000111101001101100000110010100



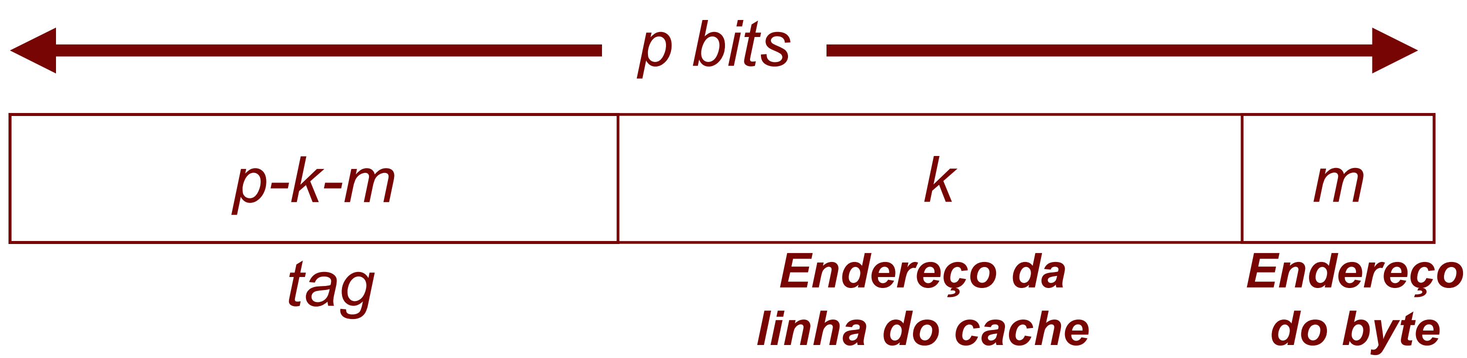
Exemplo

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 87a6c1b4
  - ❖ 100001111010011011000001100 10100



Exemplo

- p = 32 bits
- m = 5 bits (2^5 = 32)
- k = 13 bits (2^{13}=8 k)
- 87a6c1b4
  - ❖ `1000011110100 1011000001100 10100`
  - ❖ `k = 1011000001100`



Exemplo

- Bloco endereçado por 87a6c1b4 (hexadecimal)
- Linha de cache 1011000001100 (binário)
  - 160C (hexadecimal)
  - 5644 (decimal)

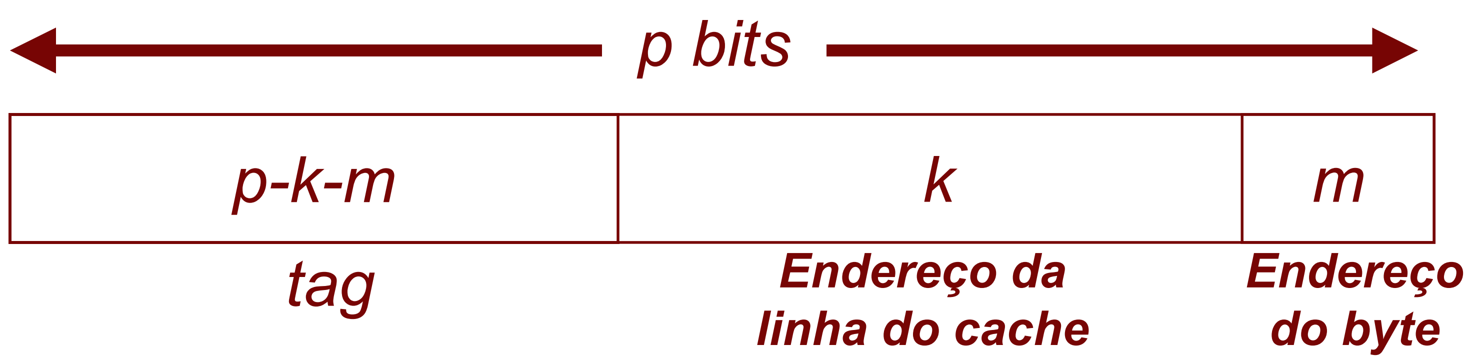
Problema

- Refazer o problema para o endereço
  8<u>8</u>a6c1b4 (hexadecimal)

cederj

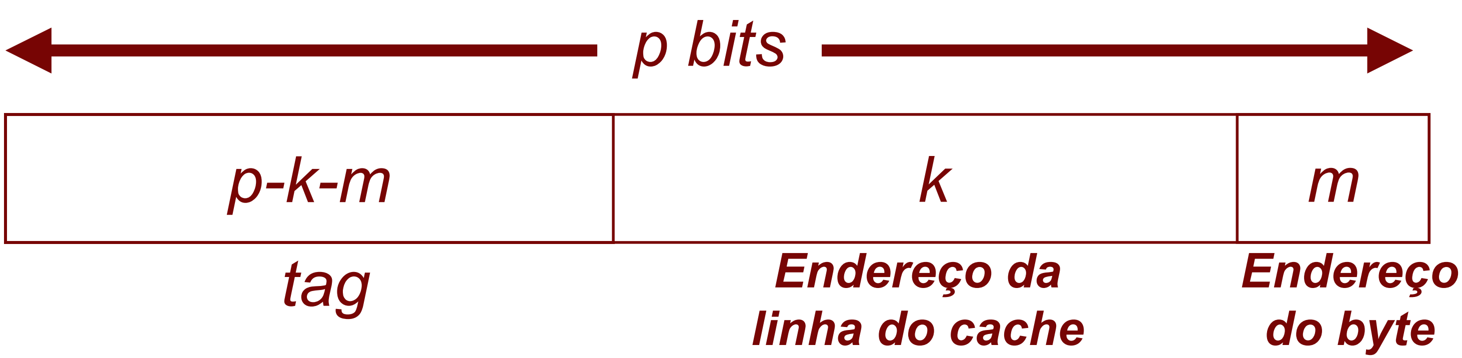
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 8 a 6 c 1 b 4
    - ❖ 1000 1000 1010 0110 1100 0001 1001 0100



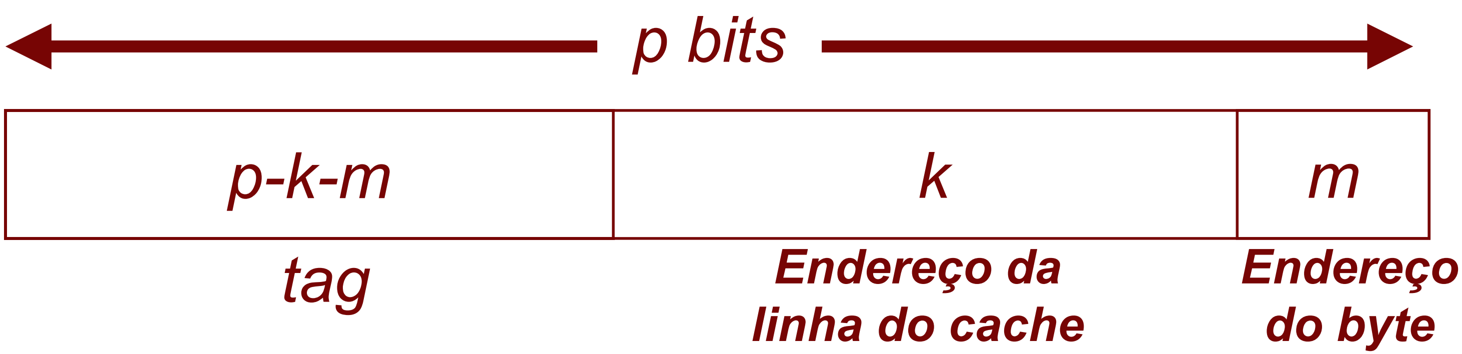
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 8 a 6 c 1 b 4
  - ❖ 1000 1000 1010 0110 1100 0001 10010100



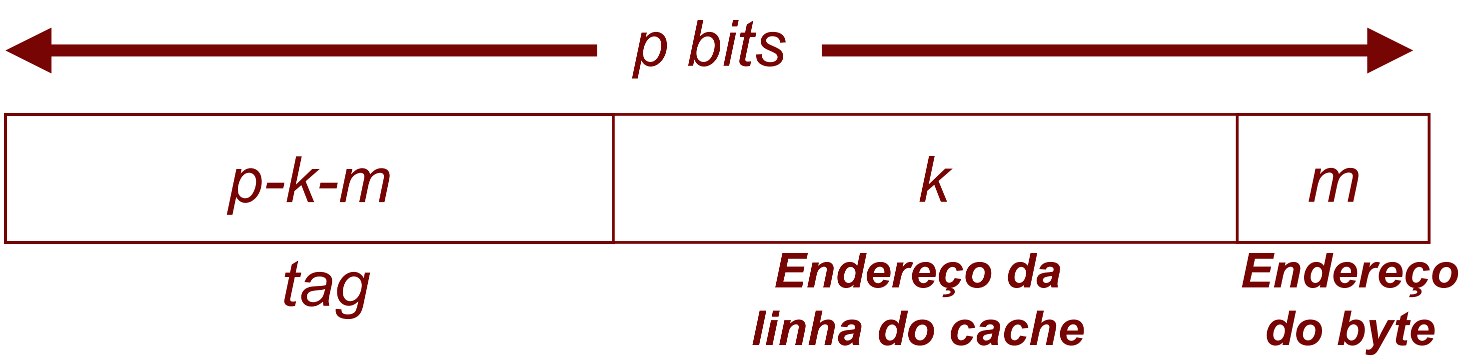
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 8 a 6 c 1 b 4
    - ❖ 1000 1000 1010 0110 1100 0001100010100



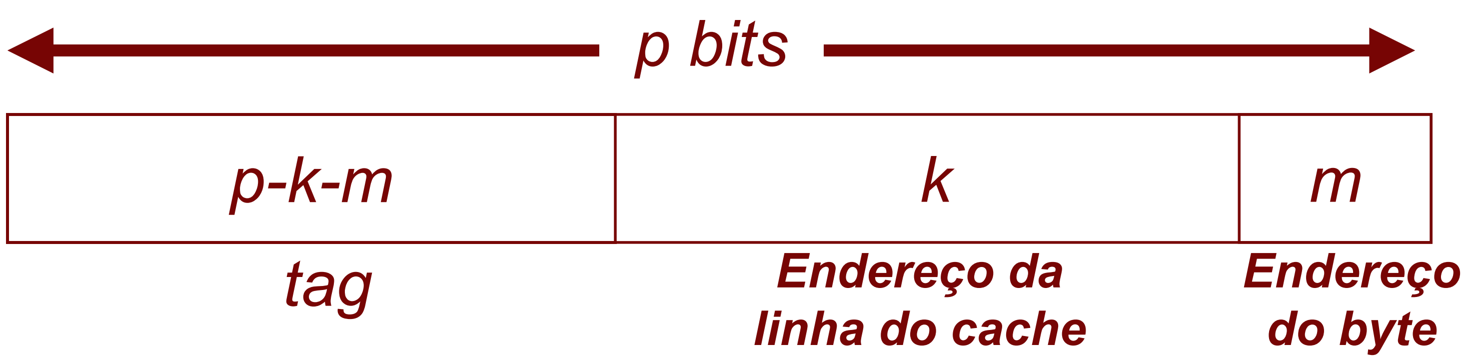
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 8 a 6 c 1 b 4
  - ❖ 1000 1000 1010 0110 11000000110010100



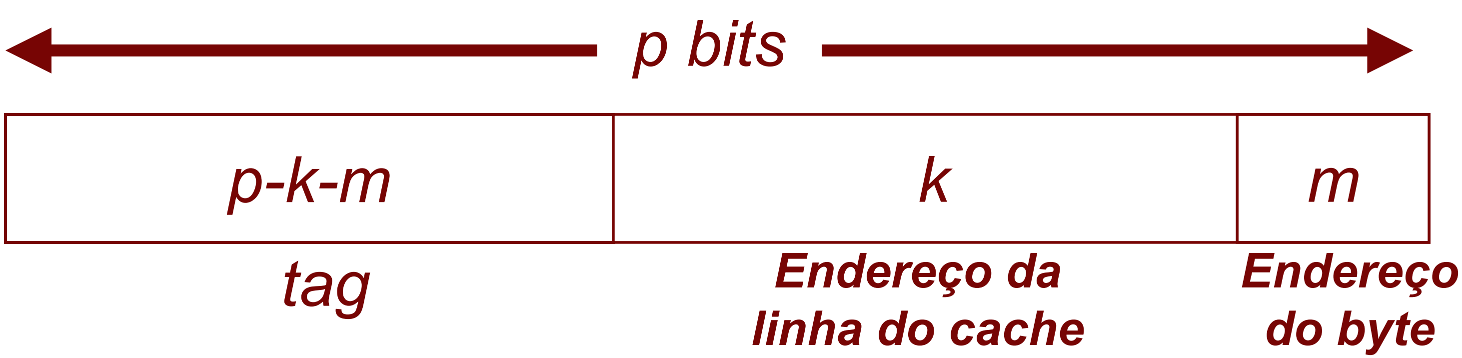
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 8 a 6 c 1 b 4
  - ❖ 1000 1000 1010 01101100000110010100



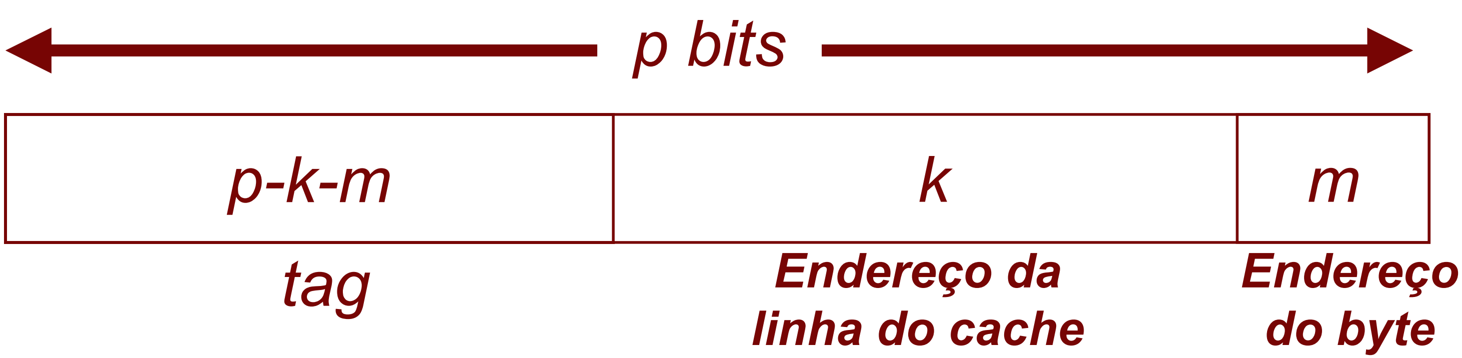
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- `8 8 a 6 c 1 b 4`
  - ❖ `1000 1000 1010011011000000110010100`



Problema

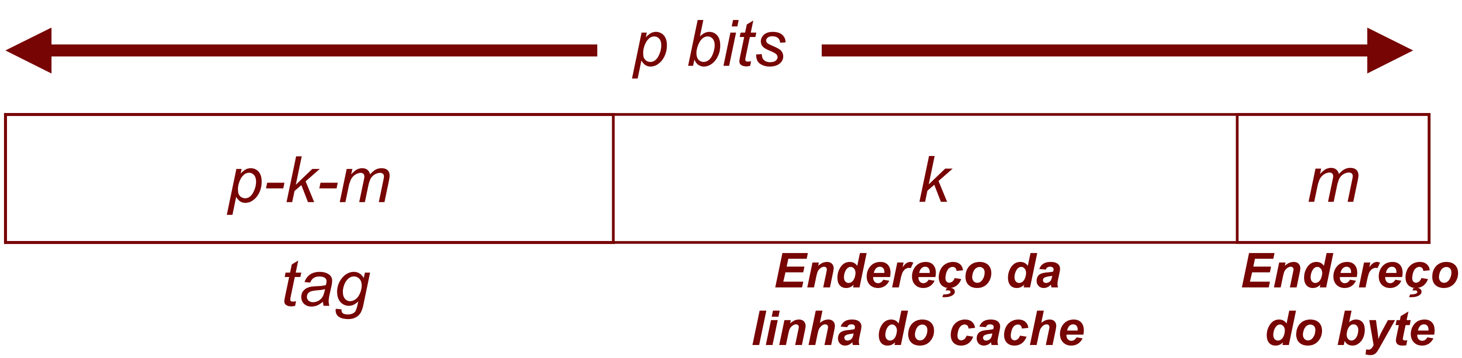
- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- `8 8 a 6 c 1 b 4`
  - ❖ `1000 10001010011011000000110010100`



cederj

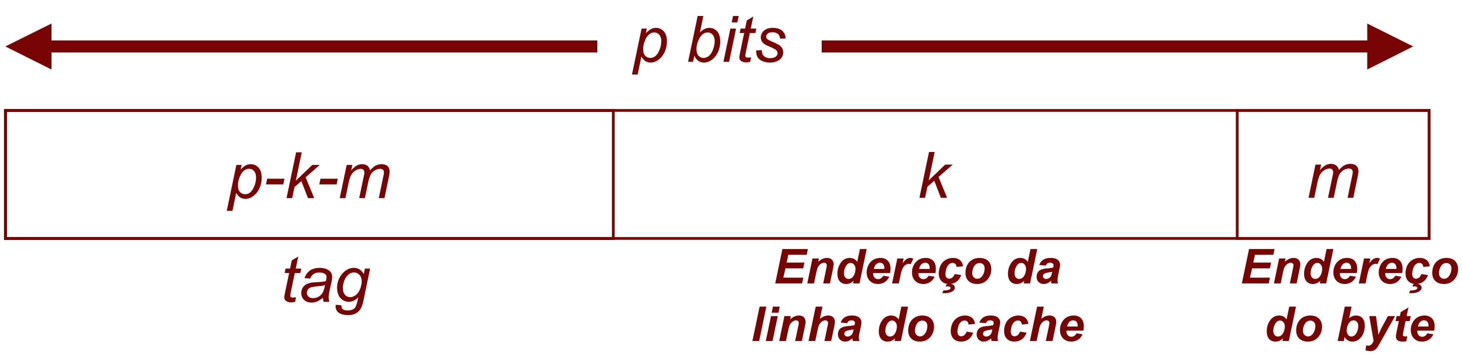
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 8 8 a 6 c 1 b 4
  - ❖ 10001000101001101100000110010100



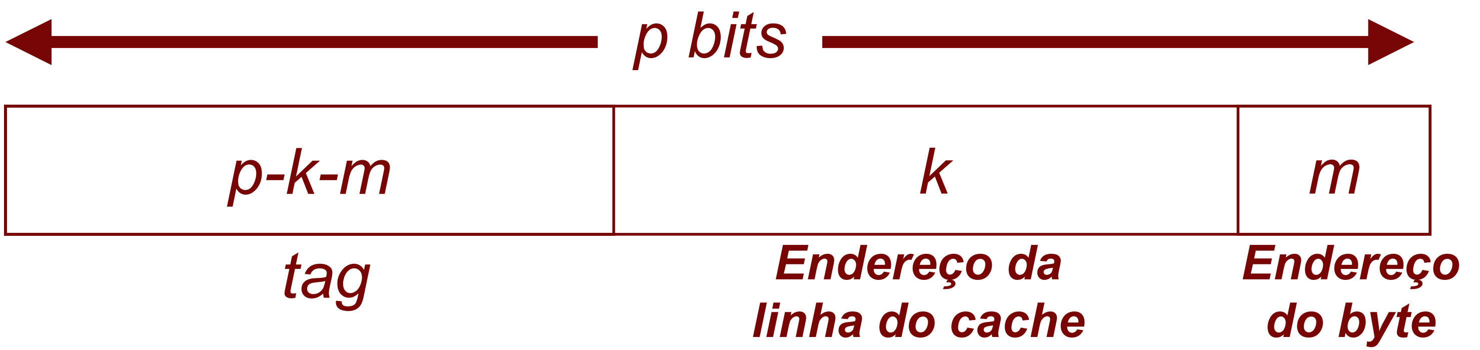
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 88a6c1b4
  - 10001000101001101100000110010100



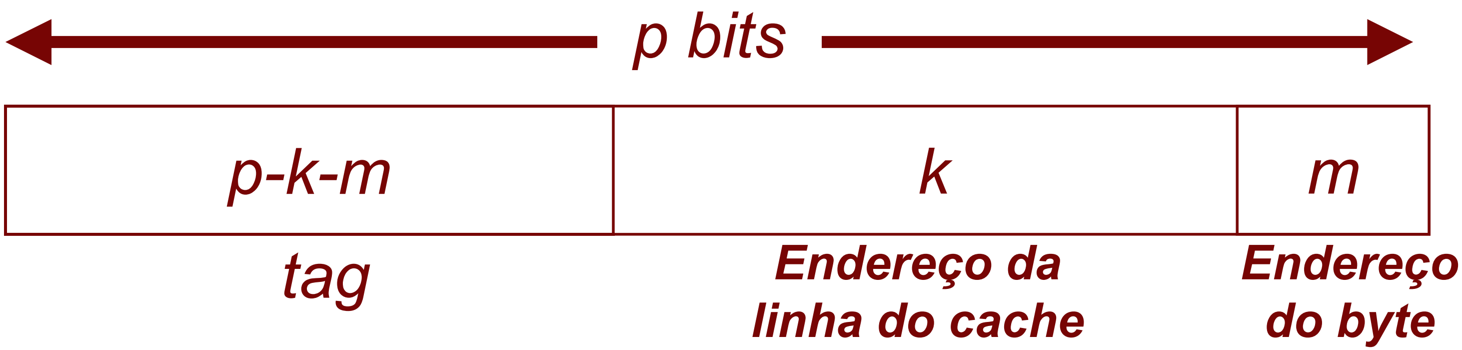
Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 88a6c1b4
  - ❖ 1000100010100110110000011 00 10100



Problema

- p = 32 bits
- m = 5 bits ($2^5 = 32$)
- k = 13 bits (2^{13}=8 k)
- 88a6c1b4
  - ❖ `10001000101001 1011000001100 10100`
  - ❖ `k = 1011000001100`



cederj

Problema

- Bloco endereçado por 88a6c1b4 (hexadecimal)
- Linha de cache 1011000001100 (binário)
  - 160C (hexadecimal)
  - 5644 (decimal)

Problema

- Os endereços 87a6c1b4 e 88a6c1b4 necessitam da mesma linha de cache

- Conclusão:
  - Todo par de endereços cuja diferença for múltipla do tamanho da cache vai utilizar a mesma linha gerando conflito
  - Sempre que houver conflito, a linha mais antiga será descartada da cache

Próxima Aula

- Continuação de hierarquia de memória
- Tentativa de solucionar o problema de conflito no cache

Introdução à Informática

Alexandre Meslin
(meslin@nce.ufrj.br)

cederj

Organização da Memória

- Conceito de hierarquia de memória
- Memória principal e memórias secundárias
- Projeto lógico da memória principal
- Memórias cache
- Memória virtual

Conceito de Hierarquia de Memória

- Memória no computador
  - Registradores
  - Memória principal
  - **Memória cache**
  - Memória ROM
  - Discos magnéticos
  - Discos ópticos
  - Fitas magnéticas

Memória Cache

- Na aula anterior, foi visto:
  - Memória cache com mapeamento direto
  - Conflito de endereços na cache

Cache com Conjunto Associativo

- Semelhante ao mapeamento direto
- Bits mais baixos utilizados para selecionar linhas com conjuntos de 2, 4, 8 ou mais blocos
- Parte alta do endereço utilizada para garantir que realmente houve um acerto

Cache com Conjunto Associativo

- Assumindo que:
  - ❖ Memória cache tem 2^k linhas
  - ❖ Bloco com 2^m bytes
  - ❖ p bits de barramento de endereço
  - ❖ 2^a conjuntos associativos
- Bits mais baixos utilizados para selecionar a linha da cache
- Bits mais altos usados para comparar o endereço da linha

← *p bits* →

p-k+a-m	*k-a*	*m*
tag	***Endereço da linha do cache***	***Endereço do byte***

Cache Associativo

- Cache com *n* conjuntos associativos
  - Cada linha de cache pode ter *n* blocos
  - *n* pode ser pequeno (2, 4, 8)
  - Melhor desempenho
  - Custo moderado

Cache Associativo

- Como cada linha possui o mesmo endereço básico →
- Para selecionar a linha, basta calcular o seu endereço, com no exemplo de mapeamento direto
- Dentro de uma linha, vários blocos
- Para selecionar o bloco, deve se eleita uma política de substituição

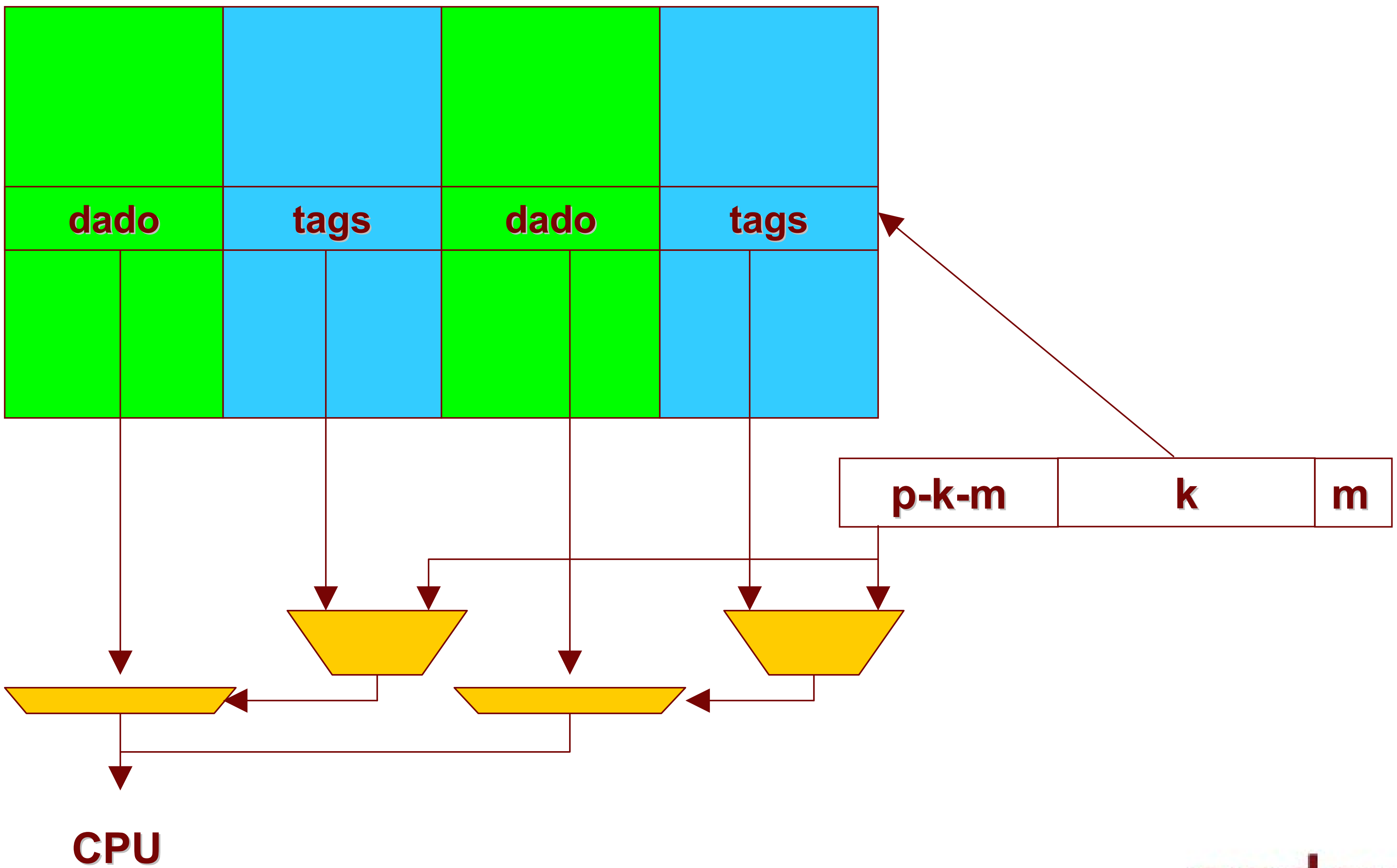
Política de Substituição de Bloco

- FIFO (First-in, First-out): primeiro a chegar, primeiro a sair
  - Se houver blocos vazios, qualquer um pode ser escolhido
  - Se todos estiverem ocupados, o mais antigo deve sair

Política de Substituição de Bloco

- LRU (least recently used): menos recentemente utilizado
  - ❖ Se houver blocos vazios, qualquer um pode ser escolhido
  - ❖ Se todos estiverem ocupados, o bloco que está a mais tempo sem ser utilizado será substituido

Cache com Conjunto Associativo



Exemplo

- Supondo memória cache com:
  - 256 bytes
  - 32 bytes / bloco
  - 8 linhas
  - Mapeamento direto
  - Barramento de endereços de 32 bits
  - Palavra de dados de 32 bits

Exemplo

- Comportamento de um programa:
  - ❖ Supor um programa que inicie no endereço 0
  - ❖ Muito provavelmente a seqüência de endereços de instruções será 0, 4, 8, 12, 16, 20, …
  - ❖ Inicialmente a cache está toda vazia (bit de validade em zero)

Exemplo

- Estado inicial da cache

Dado	Endereço	Validade
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 0 (**falha**)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 4 (acerto)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 8 (acerto)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 12 (acerto)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 16 (acerto)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 20 (acerto)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 24 (acerto)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

cederj

Exemplo

- Estado da cache depois do acesso 28 (acerto)

Dado	Endereço	Validade
Inst 0 → 28	0	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 32 (**falha**)

Dado	Endereço	Validade
Inst 0 → 32	0	1
Inst 32 → 60	32	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Exemplo

- Estado da cache depois do acesso 36 (acerto).

Dado	Endereço	Validade
Inst 0 → 32	0	1
Inst 32 → 60	32	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Outro Exemplo

- Este mesmo sistema, agora somando duas matrizes
- Cada matriz ocupa 256 bytes de memória
- Matriz 8x8 inteiros de 4 bytes
- Observe que a distância entre as matrizes pode ser múltipla do tamanho do cache
- Alto risco de haver conflito

Trecho do Programa-Exemplo

```
Para i ← 0 até 7 faça
   Para j ← 1 até 7 faça
        a(i,j) ← b(i,j)+c(i,j)
   fim para
fim para
```

- Este programa irá gerar (além de outros) acessos a endereços com 256 bytes de diferença para acessar as matrizes b e c

Estado da Cache

- No início

Dado	Endereço	Validade
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Estado da Cache

- Leitura do dado b(0, 0) (**falha**)

Dado	Endereço	Validade
b(0,0)	End b(0,0)	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Estado da Cache

- Leitura do dado c(0, 0) (**falha**)

Dado	Endereço	Validade
c(0,0)	End c(0,0)	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Estado da Cache

- Leitura do dado b(0, 1) (**falha**)

Dado	Endereço	Validade
b(0,1)	End b(0,1)	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Estado da Cache

- Leitura do dado c(0, 1) (**falha**)

Dado	Endereço	Validade
c(0,1)	End c(0,1)	1
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0
?	?	0

Cache com Conjunto Associativo

- Mesmo sistema, mesmo programa, agora com cache com conjunto associativo
- Supondo memória cache com:
  - 256 bytes
  - 32 bytes / bloco
  - 4 linhas
  - 2 ou mais conjuntos associativos
  - Barramento de endereços de 32 bits
  - Palavra de dados de 32 bits

Estado da Cache

- No início

Dado	End.	Val.	Dado	End.	Val.
?	?	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0

Estado da Cache

- Leitura do dado b(0,0) (**falha**)

Dado	End.	Val.	Dado	End.	Val.
b(0,0) b(0,7)	End. b(0,0)	1	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0

Estado da Cache

- Leitura do dado c(0,0) (**falha**)

Dado	End.	Val.	Dado	End.	Val.
b(0,0) b(0,7)	End. b(0,0)	1	c(0,0) c(0,7)	End. c(0,0)	1
?	?	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0

Estado da Cache

- Leitura do dado b(0,1) (acerto)

Dado	End.	Val.	Dado	End.	Val.
b(0,0) b(0,7)	End. b(0,0)	1	c(0,0) c(0,7)	End. c(0,0)	1
?	?	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0

Estado da Cache

- Leitura do dado c(0,1) (acerto)

Dado	End.	Val.	Dado	End.	Val.
b(0,0) b(0,7)	End. b(0,0)	1	c(0,0) c(0,7)	End. c(0,0)	1
?	?	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0

Estado da Cache

- Mais tarde…

Dado	End.	Val.	Dado	End.	Val.
b(0,0) b(0,7)	End. b(0,0)	1	c(0,0) c(0,7)	End. c(0,0)	1
?	?	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0

Estado da Cache

- Leitura do dado b(1,0) (acerto)

Dado	End.	Val.	Dado	End.	Val.
b(0,0) b(0,7)	End. b(0,0)	1	c(0,0) c(0,7)	End. c(0,0)	1
b(1,0) b(1,7)	End. B(1,0)	0	?	?	0
?	?	0	?	?	0
?	?	0	?	?	0

Estado da Cache

- Leitura do dado c(1,0) (acerto).

Dado	End.	Val.	Dado	End.	Val.
b(0,0) b(0,7)	End. b(0,0)	1	c(0,0) c(0,7)	End. c(0,0)	1
b(1,0) b(1,7)	End. B(1,0)	0	c(1,0) c(1,7)	End. c(1,0)	0
?	?	0	?	?	0
?	?	0	?	?	0

Cache Totalmente Associativo

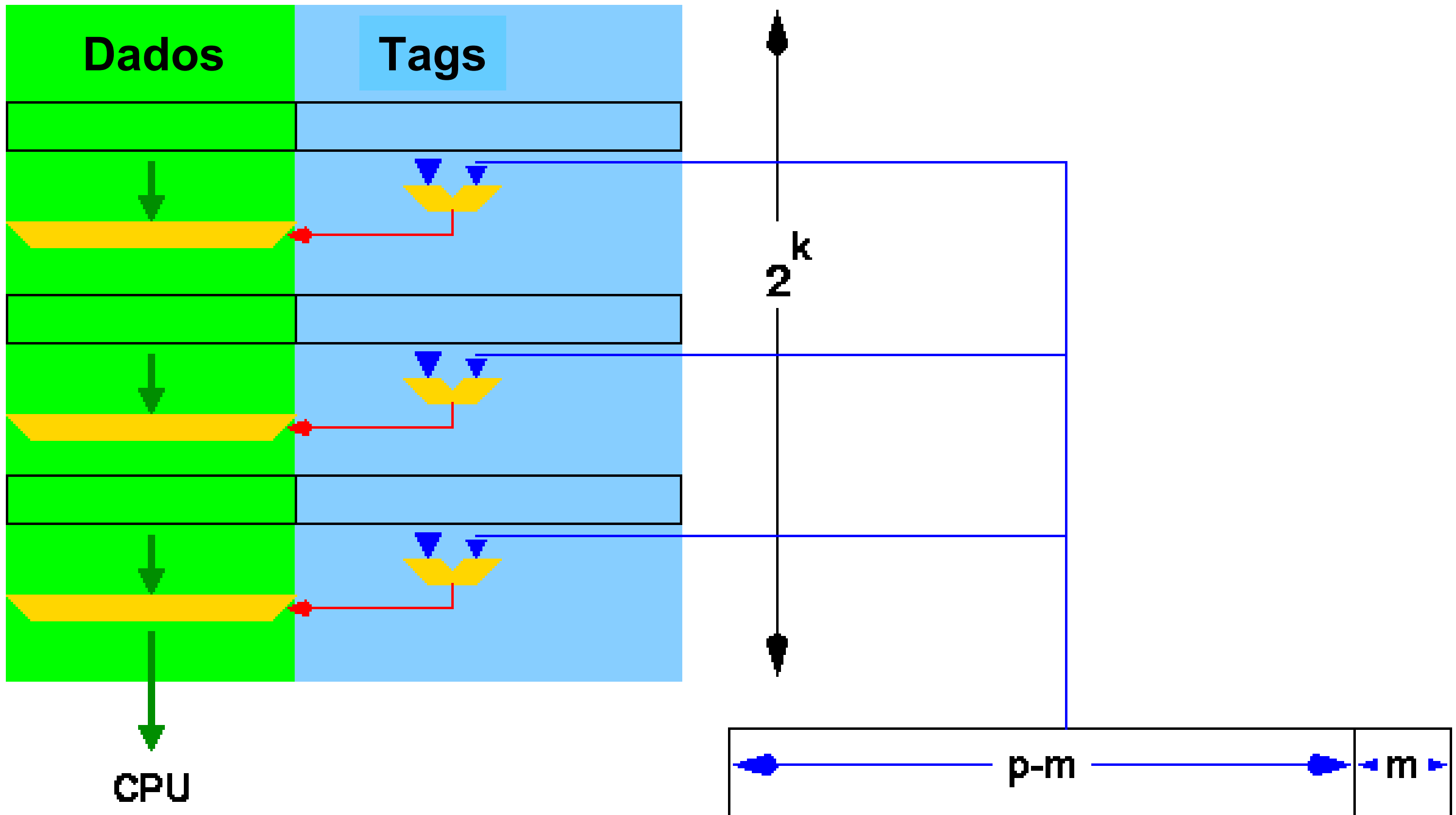
- Pedido da CPU procurado em todo o cache
  - Se encontrar: acerto na cache (cache hit)
  - Caso contrário: falha na cache (cache miss)
- Todas as tags são comparadas ao mesmo tempo
- Qualquer endereço de memória pode estar sendo representado em qualquer linha da cache
- Evita substituição desnecessária

Cache Totalmente Associativo

- Política de substituição
  - LRU
  - FIFO
- Necessita de muitos bits adicionais de estado (na tag)
- Custo de hardware muito alto
  - Muitos comparadores
  - Tags muito grandes (muitos bits)

cederj

Cache Totalmente Associativo



Política de Escrita – Write Through

- Write Through
  - Todo acesso de escrita com acerto é feito simultaneamente na cache e na memória principal
  - Se houver um acesso de escrita com falha, somente a memória principal será modificada
  - Informação na memória principal está sempre atualizada
  - Leitura pode ser feita em um único ciclo
  - Escrita demora o tempo de acesso à memória principal

Política de Escrita – Write Back

- Write Back
  - Todo acesso de escrita com acerto é feito somente na cache
  - O que ocorre em caso de falha dependerá da política de alocação (ver mais tarde)
  - Informação na memória principal pode estar desatualizada
  - Leitura e escrita podem ser feitas em um único ciclo
  - Se o bloco for substituído e a sua informação estiver sido modificada, esta deverá ser enviada para a memória principal

Outros Campos da Tag

- Bit de “dirty” (sujo)
- Utilizado para controlar o estado da memória principal
  - Coerente → memória atualizada
  - Incoerente → memória desatualizada
- Caso a memória esteja incoerente, este bit indicará que o bloco na cache está sujo
- Em caso de substituição deste bloco, os dados deverão ser escritos na memória

Política de Alocação

- O que acontece em caso de falha de escrita em um cache write back?

1. A informação pode ser lida da memória principal para a cache e então modificada
2. A escrita pode ser feita diretamente na memória principal

Write Back x Write Through

- Qual o melhor?
- Write back
  - Escritas mais rápidas
- Write through
  - Bom para multiprocessamento (memória sempre coerente)
  - Necessita de menos hardware

Próxima Aula

- Continuação da hierarquia de memória

Introdução à Informática

Alexandre Meslin
(meslin@nce.ufrj.br)

cederj

Objetivos

- Dispositivos de armazenamento
- Sistemas de arquivos

Memória ROM

- Memória de apenas leitura
- Utilizada para armazenar programas e dados de modo permanente
- O seu conteúdo não é perdido quando a energia é desligada
- Atualmente os PC's possuem ROM's de até 1 megabyte.
- Tempo de acesso entre 20 ns e 80 ns

Memória ROM

- Diversos tipos de ROM atualmente em uso
  - PROM: ROM programável uma vez
  - EPROM: PROM apagável (Eraseble) e reprogramável
  - EEPROM: EPROM apagável eletricamente
- Computadores PC usam EEPROM's para armazenarem a BIOS
- Usuário pode atualizar através de programas especiais

Disco Óptico

- CD
- DVD
- Capacidade entre 150 Mbytes e 4,7 Gbytes
- Tempo de acesso entre 80 ms e 300 ms
- Utilizam laser para leitura (escrita)
- O seu conteúdo não é perdido quando a energia é desligada

Fitas Magnéticas

- Similar aos discos magnéticos
- Acesso puramente seqüencial
- Diversos tipos disponíveis
  - DAT
  - 4 mm
  - 8mm
- Muito mais lenta do que qualquer outra memória
- Muito mais barata
- Muito utilizada para cópia de segurança (backup)
- Capacidade de armazenamento entre 150 MB e 40 GB
- Pode ser lida e escrita várias vezes

Disco Magnético

- Disco flexível (disquete) (de 1.2MB até 1GB)
- Hard Disk (HD) (muitos gigabytes)
- Muito mais barata que memória cache ou DRAM
- Por possuir componentes mecânicos, é muito mais lento que a memória de circuito integrado
- Tempo de acesso entre 5 ms e 13 ms
- O seu conteúdo não é perdido quando a energia é desligada

Disco Magnético

- Utilizado como memória secundária
  - ❖ Memória virtual
  - ❖ Capacidade muito maior do que memória principal
  - ❖ Tempo de acesso muito maior
- Sistema de armazenamento definitivo
  - ❖ Arquivos
    - Programas
      - Aplicativos
    - Dados
      - Textos
      - Músicas
      - Imagens
  - ❖ Diretórios (Pastas)

Acesso ao Sistema de Arquivos

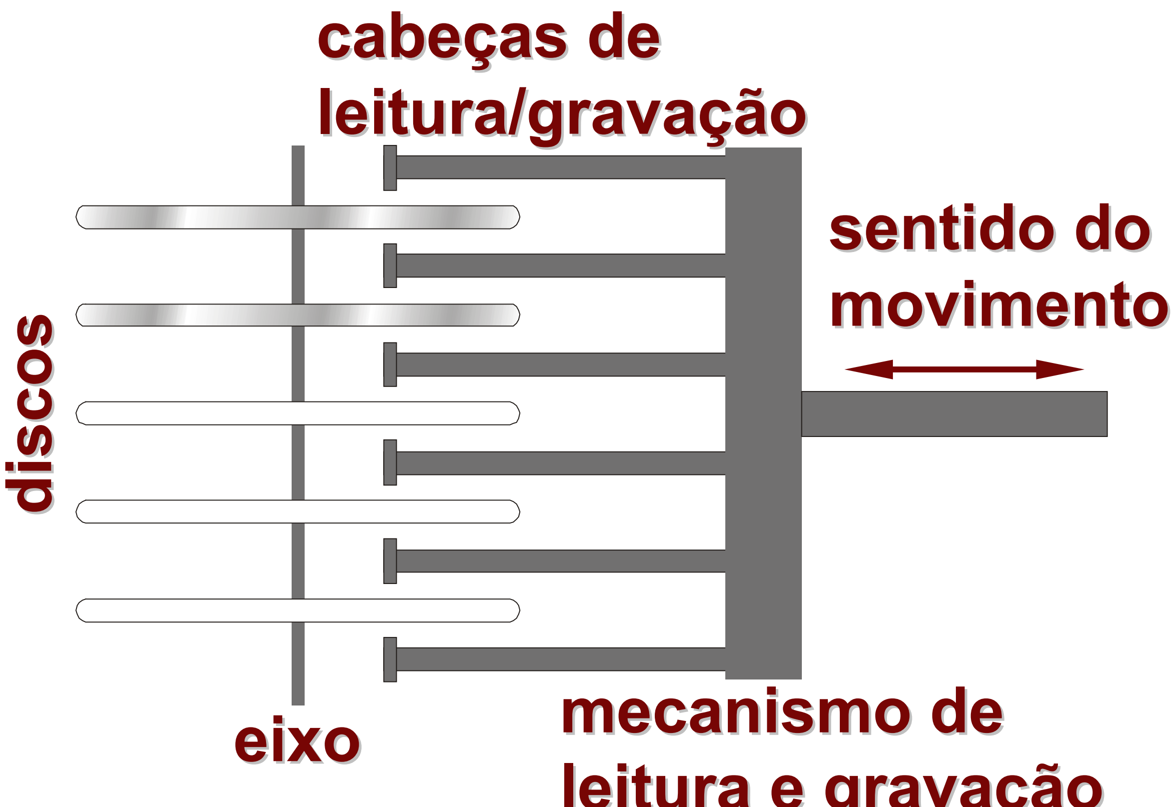
- Hierarquia
  - Aplicação utiliza funções genéricas de sua biblioteca
  - Funções da biblioteca chamam funções do sistema operacional (*systems call*)
  - Funções do sistema operacional utilizam funções de mais baixo nível
  - Estas funções utilizam acionadores (*drivers*) específicos do hardware
  - Acesso ao hardware de armazenamento

Disco

- Seqüência de bits
- Significado dependente
  - Sistema operacional
  - Aplicação
  - Hardware
- Formatação
  - Necessária para permitir ajuda do hardware para acesso ao meio

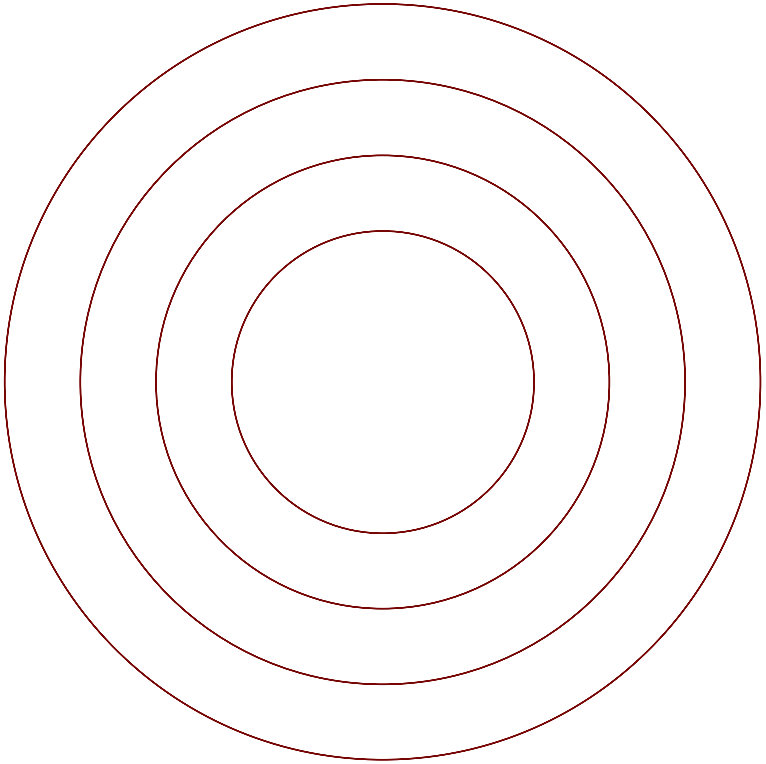
Organização Física do Disco

- Unidade composta por diversas superfícies



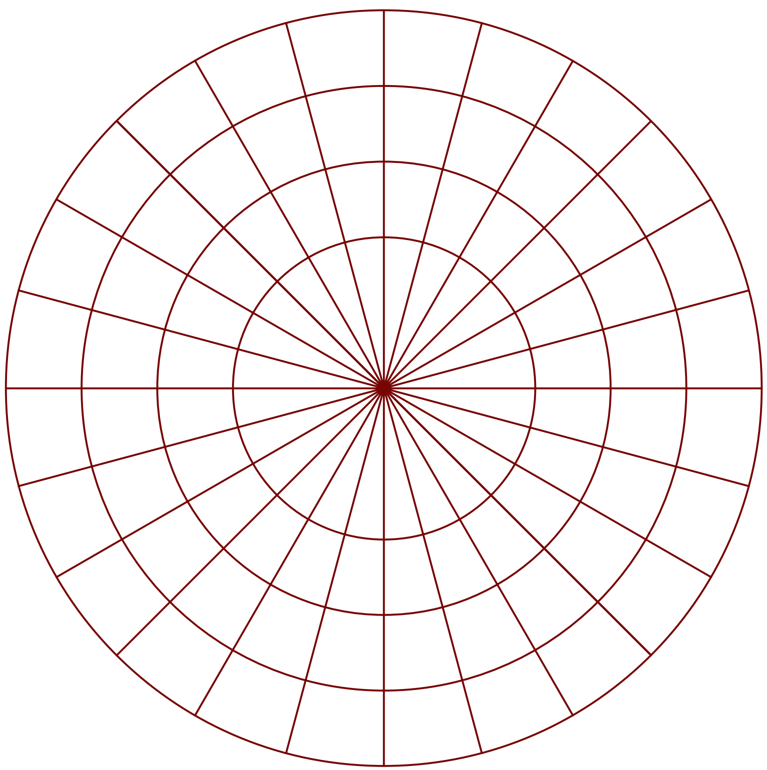
Organização Física do Disco

- Disco dividido em trilhas concêntricas



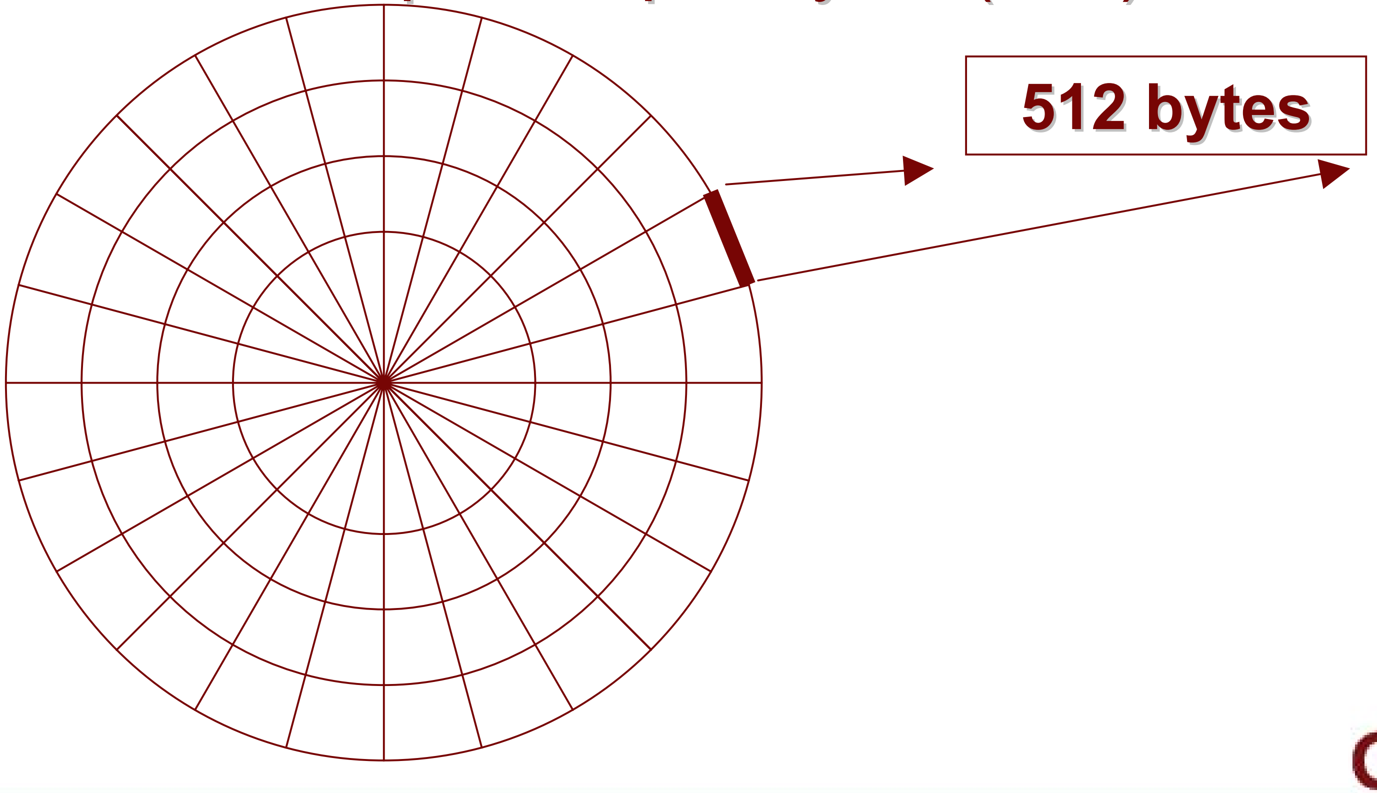
Organização Física do Disco

- Disco dividido em trilhas concêntricas
- Trilhas divididas em setores radiais



Organização Física do Disco

- Disco dividido em trilhas concêntricas
- Trilhas divididas em setores radiais
- Setores compostos por bytes (512)



Cálculo da Capacidade

- Capacidade do disco
  - ❖ Medida em bytes e seus multiplicadores (MB, GB)
  - ❖ Capacidade =

    Número de superfícies X

    Número de trilhas (cilindros) por superfícies X

    Número de setores por trilhas X

    Tamanho do setor (geralmente 512 bytes)

Organização Lógica do Disco

- Setores são numerados seqüencialmente pelas superfícies e trilhas
- Setores seqüenciais são agrupados em conjuntos de blocos
- Geram unidade mínima de alocação de espaço

Organização Lógica do Disco

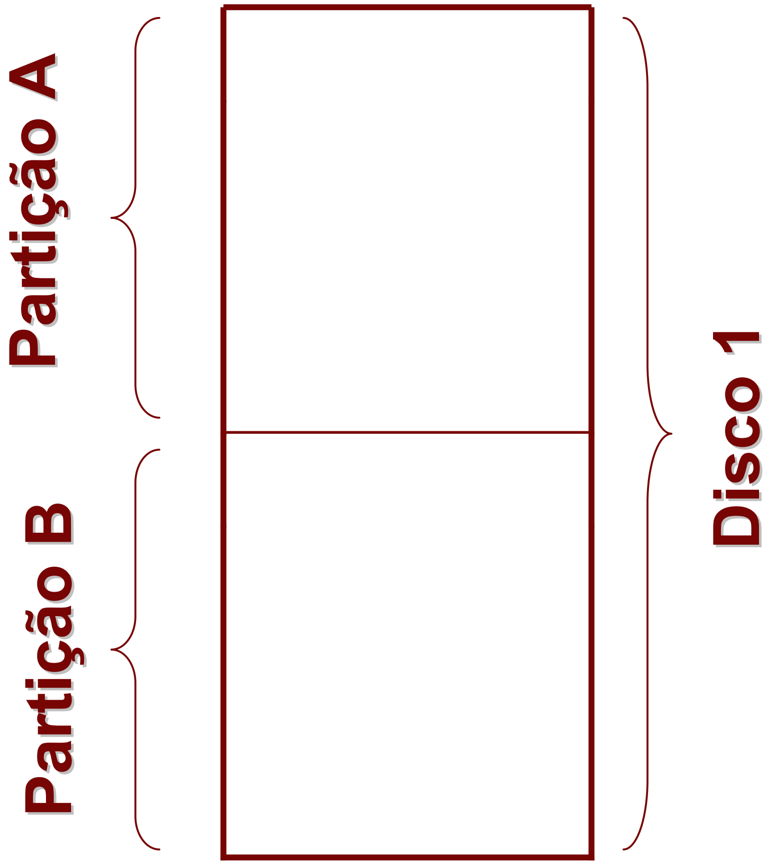
- Em um sistema de 16 bits (FAT16), podem existir, no máximo, 65.536 blocos

- Em um disco de 40 Gbytes, o tamanho do bloco será de 62 Mbytes (=40 Gbytes/65536)
  - Gera grande desperdício de espaço
  - Perda de 31 Mbytes, em média, por arquivo
  - Mesmo os arquivos pequenos ocupam 62 Mbytes
  - Quanto maior o disco, maior o desperdício

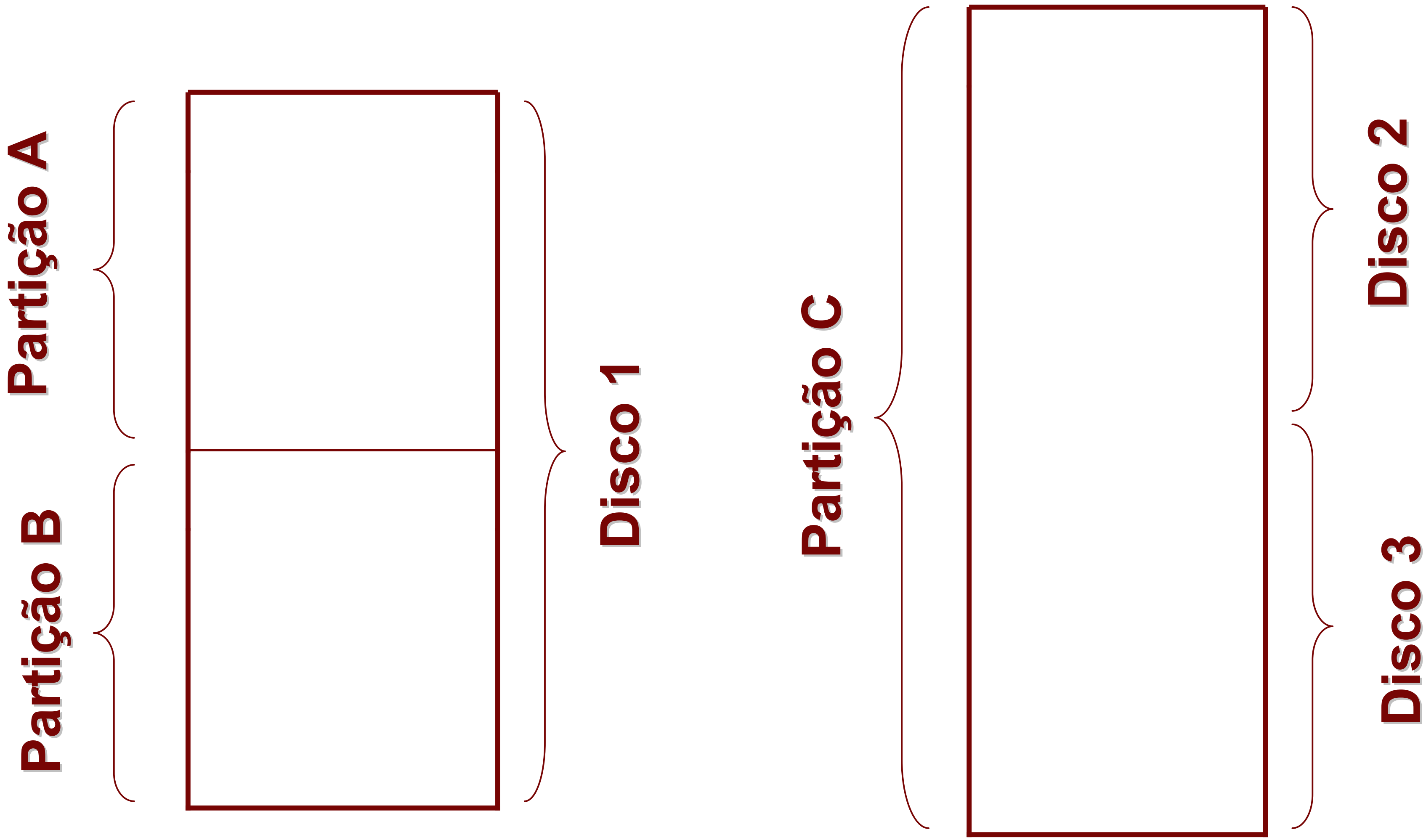
Organização Lógica do Disco

- Um sistema FAT32 permite a existência de 4.294.967.296 blocos (mais de 4 bilhões de blocos)
  - Em um disco de 250 Gbytes, blocos de 1 setor – pouco desperdício de espaço

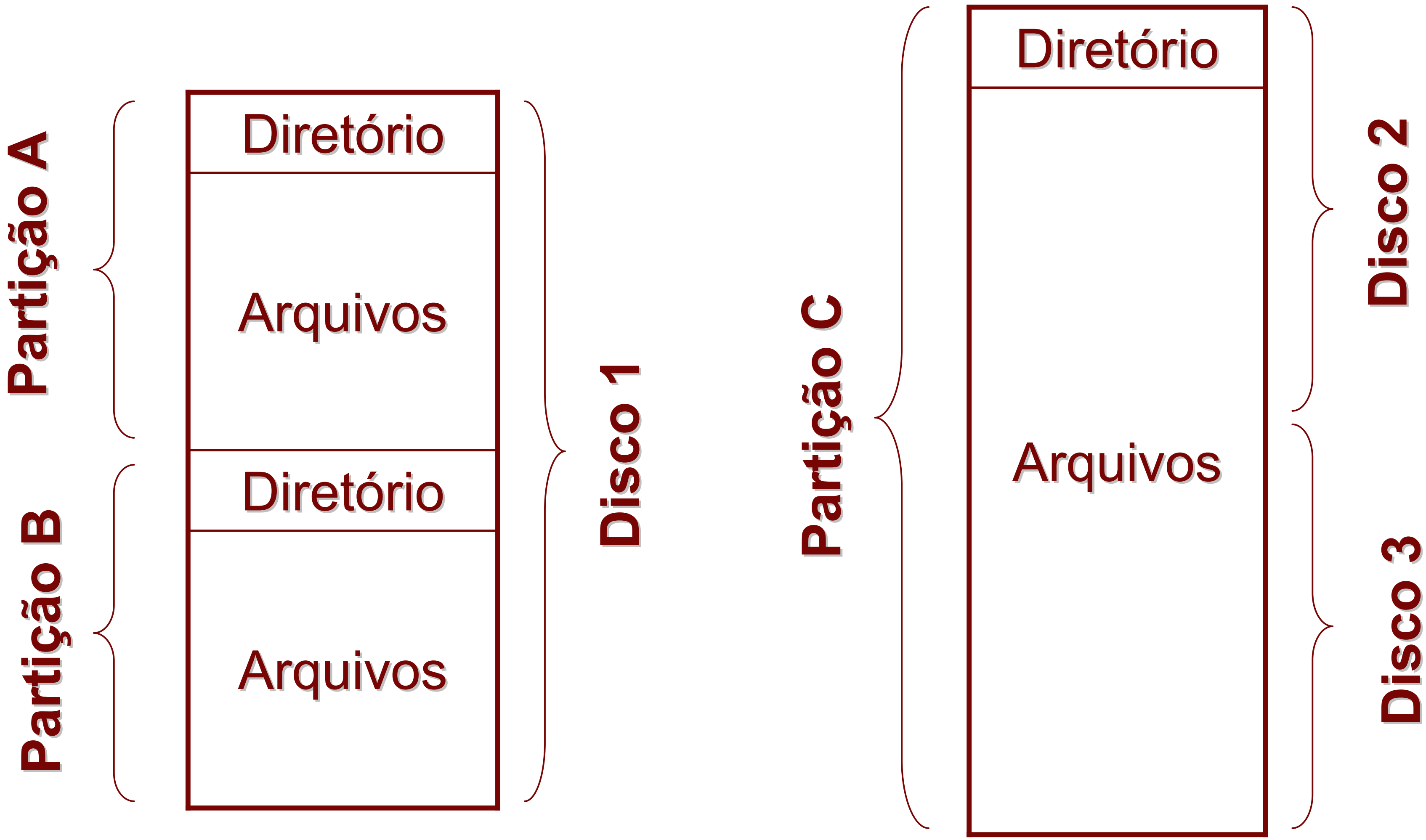
Organização Lógica do Disco



Organização Lógica do Disco



Organização Lógica do Disco

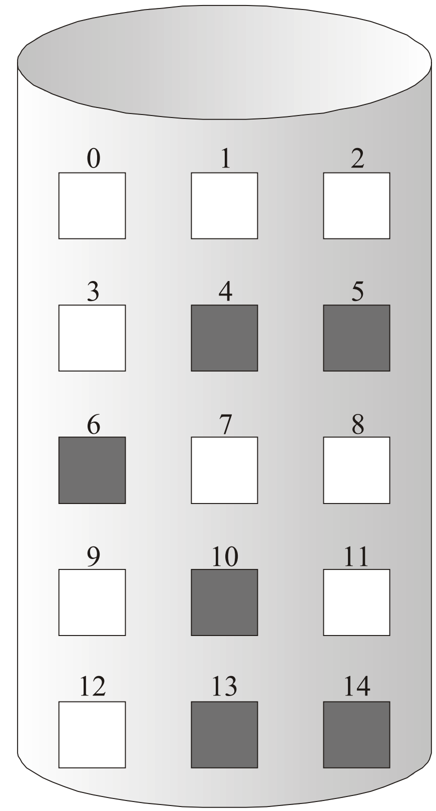


Métodos de Alocação de Blocos

- Como os blocos são alocados formando
  - Diretórios (pastas)
  - Arquivos
- Alocação contígua
- Alocação encadeada
- Alocação indexada

Alocação Contígua

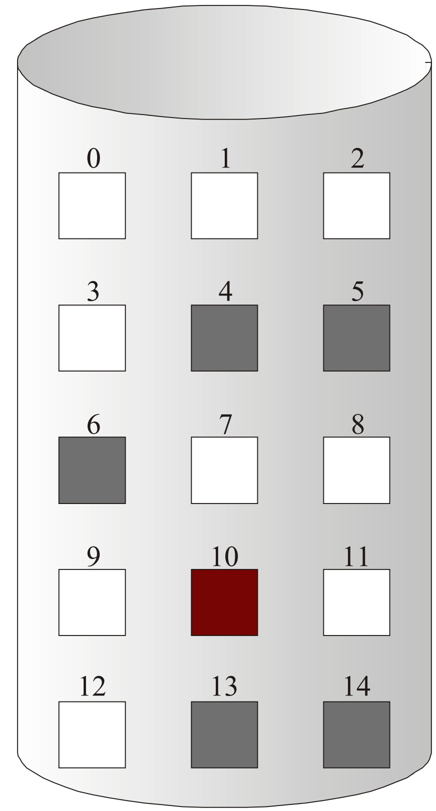
- Cada arquivo ocupa um conjunto de blocos contíguos no disco
- Método simples
  - ❖ Necessário apenas bloco inicial e número de blocos
- Permite acesso aleatório
- Grande desperdício de espaço
- Arquivos não podem crescer (facilmente)



Arquivo	Bloco	Tamanho
A. TXT	4	3
B. TXT	10	1
C. TXT	13	2

Alocação Contígua

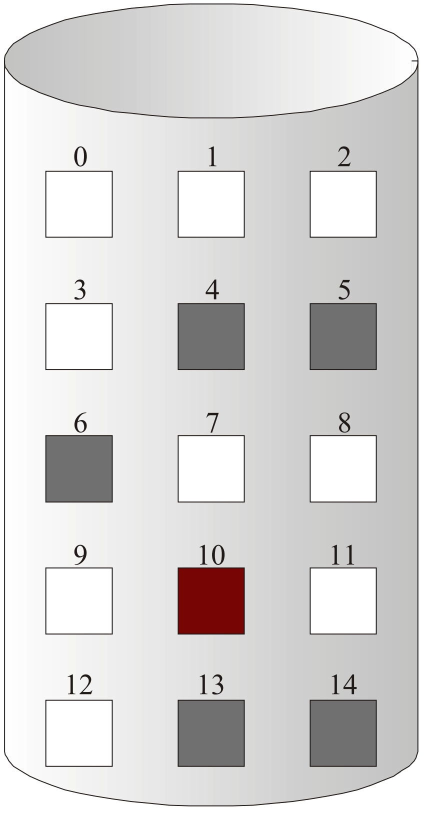
- Arquivos não podem crescer (facilmente)
- Observar arquivo B.TXT
  - ❖ Bloco inicial: 10
  - ❖ Quantidade de blocos: 1



Arquivo	Bloco	Tamanho
A. TXT	4	3
B. TXT	10	1
C. TXT	13	2

Alocação Contígua

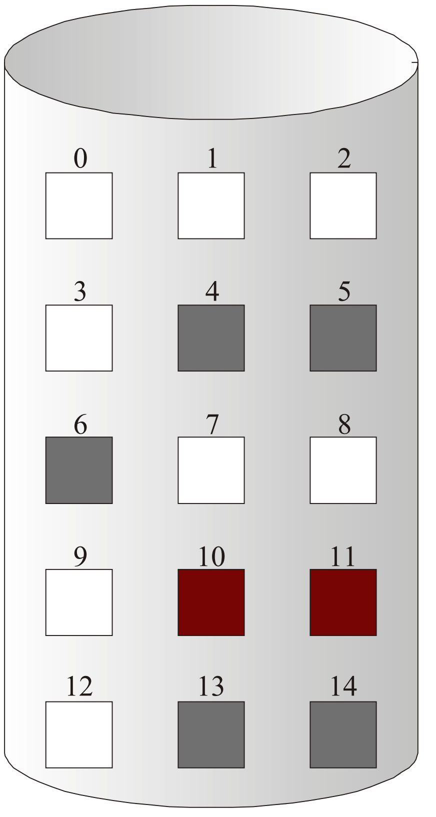
- Arquivos não podem crescer (facilmente)
- Observar arquivo B.TXT
  - ❖ Bloco inicial: 10
  - ❖ Quantidade de blocos: 1
  - ❖ **Incluir mais um bloco (11)**



Arquivo	Bloco	Tamanho
A. TXT	4	3
B. TXT	10	1
C. TXT	13	2

Alocação Contígua

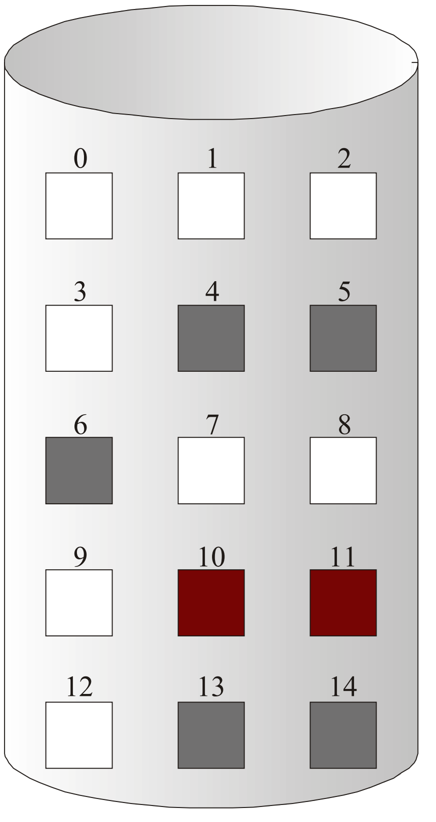
- Arquivos não podem crescer (facilmente)
- Observar arquivo B.TXT
  - ❖ Bloco inicial: 10
  - ❖ Quantidade de blocos: 1
  - ❖ **Incluir mais um bloco (11)**



Arquivo	Bloco	Tamanho
A. TXT	4	3
B. TXT	10	2
C. TXT	13	2

Alocação Contígua

- Arquivos não podem crescer (facilmente)
- Observar arquivo B.TXT
  - ❖ Bloco inicial: 10
  - ❖ Quantidade de blocos: 1
  - ❖ Incluir mais um bloco (11)
  - ❖ **Incluir mais um bloco (12)**

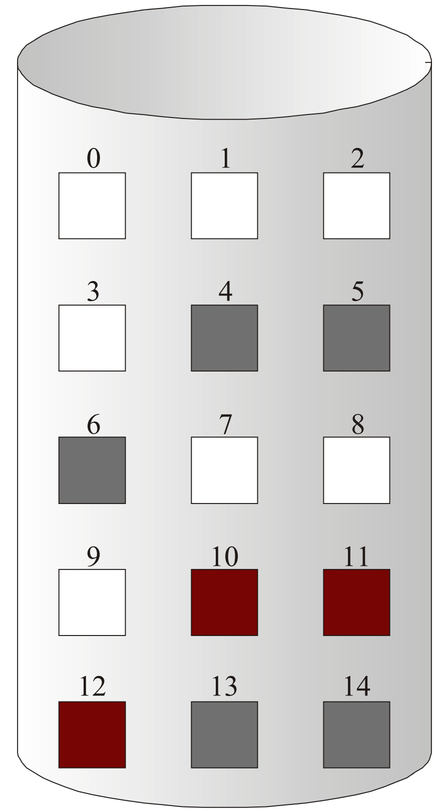


Arquivo	Bloco	Tamanho
A. TXT	4	3
B. TXT	10	3
C. TXT	13	2

cederj

Alocação Contígua

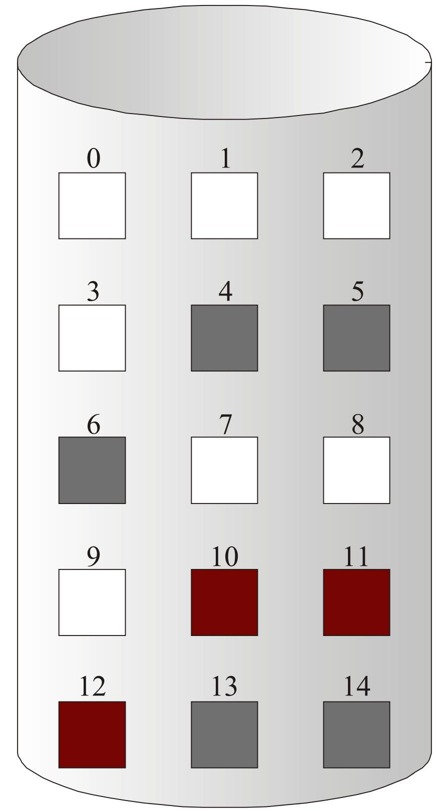
- Arquivos não podem crescer (facilmente)
- Observar arquivo B.TXT
  - ❖ Bloco inicial: 10
  - ❖ Quantidade de blocos: 1
  - ❖ Incluir mais um bloco (11)
  - ❖ **Incluir mais um bloco (12)**



Arquivo	Bloco	Tamanho
A. TXT	4	3
B. TXT	10	3
C. TXT	13	2

Alocação Contígua

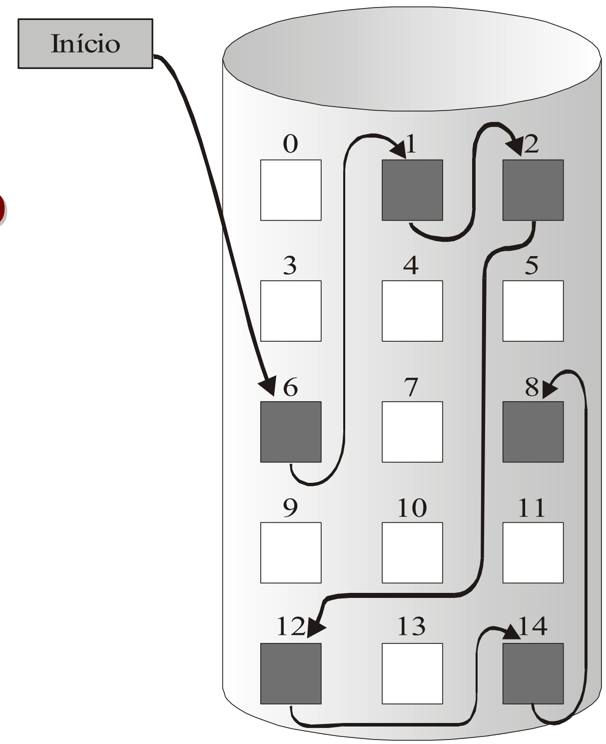
- Arquivos não podem crescer (facilmente)
- Observar arquivo B.TXT
  - ❖ Bloco inicial: 10
  - ❖ Quantidade de blocos: 1
  - ❖ Incluir mais um bloco (11)
  - ❖ Incluir mais um bloco (12)
  - ❖ **Incluir mais um bloco (???)**



Arquivo	Bloco	Tamanho
A. TXT	4	3
B. TXT	10	3
C. TXT	13	2

Alocação Encadeada

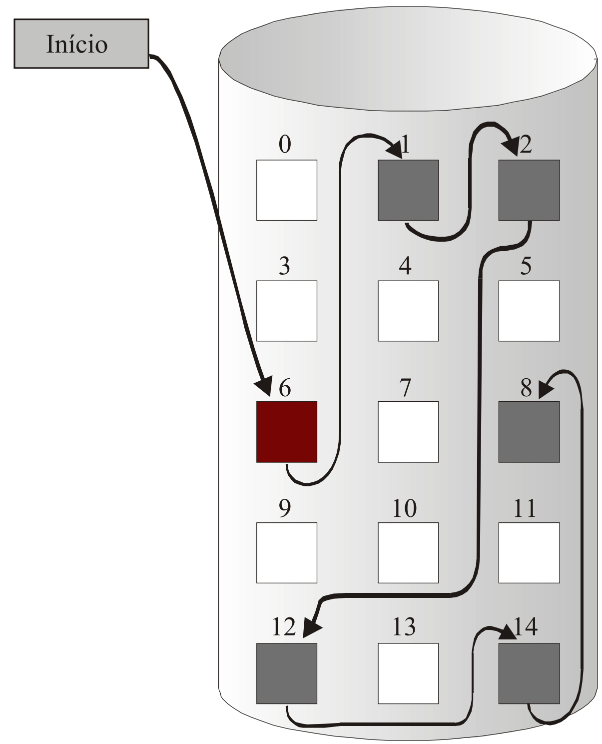
- FCB indica o início do arquivo
- Cada bloco possui um ponteiro para o próximo
- Arquivo pode crescer/diminuir facilmente
- Perda de um bloco ocasiona corrupção do restante do arquivo



Arquivo	Bloco
A.TXT	6
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

Alocação Encadeada

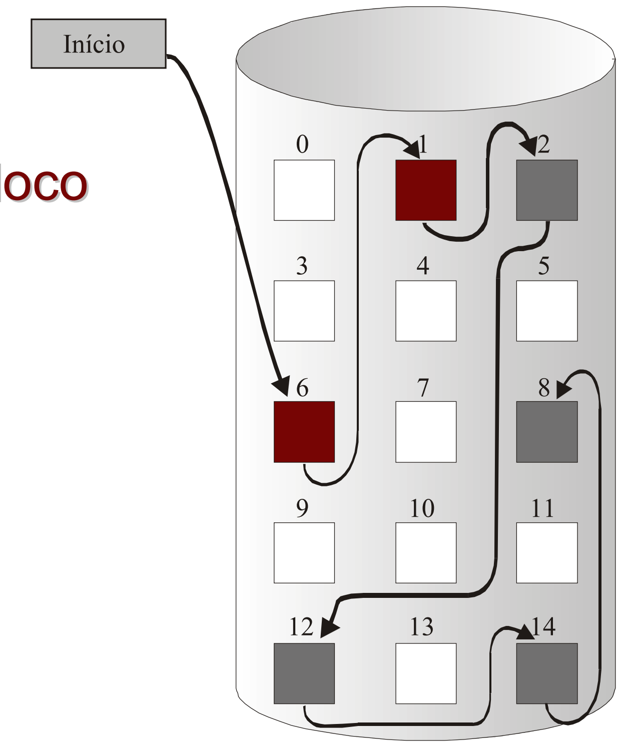
- Observar o arquivo A.TXT
  - ❖ Bloco inicial: 6



Arquivo	Bloco
A.TXT	6
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

Alocação Encadeada

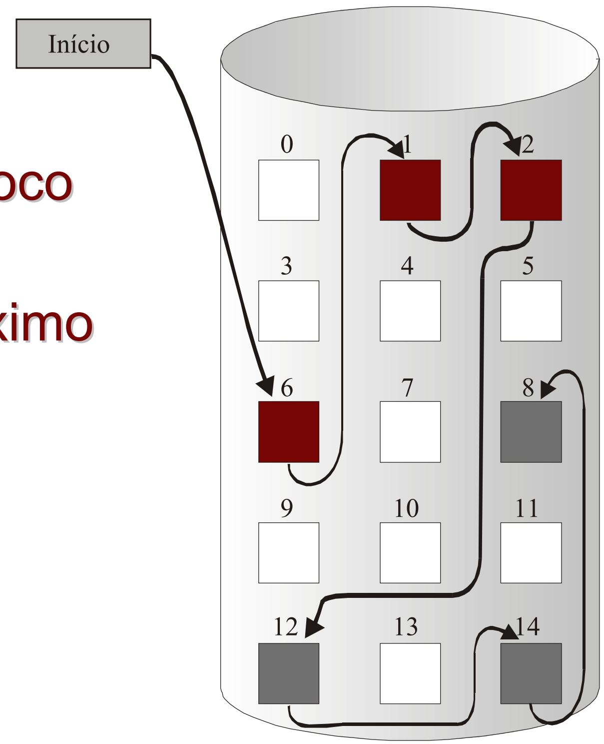
- Observar o arquivo A.TXT
  - ❖ Bloco inicial: 6
  - ❖ Final do bloco 6 possui ponteiro para próximo bloco (1)



Arquivo	Bloco
A.TXT	6
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

Alocação Encadeada

- Observar o arquivo A.TXT
  - ❖ Bloco inicial: 6
  - ❖ Final do bloco 6 possui ponteiro para próximo bloco (1)
  - ❖ Bloco 1 aponta para próximo bloco (2)



Arquivo	Bloco
A.TXT	6
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

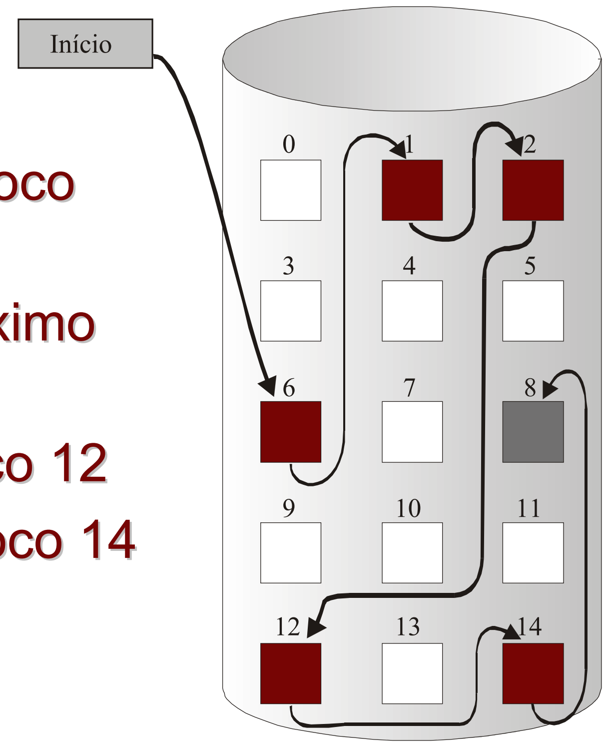
Alocação Encadeada

- Observar o arquivo A.TXT
  - ❖ Bloco inicial: 6
  - ❖ Final do bloco 6 possui ponteiro para próximo bloco (1)
  - ❖ Bloco 1 aponta para próximo bloco (2)
  - ❖ Bloco 2 aponta para bloco 12

Arquivo	Bloco
A.TXT	6
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

Alocação Encadeada

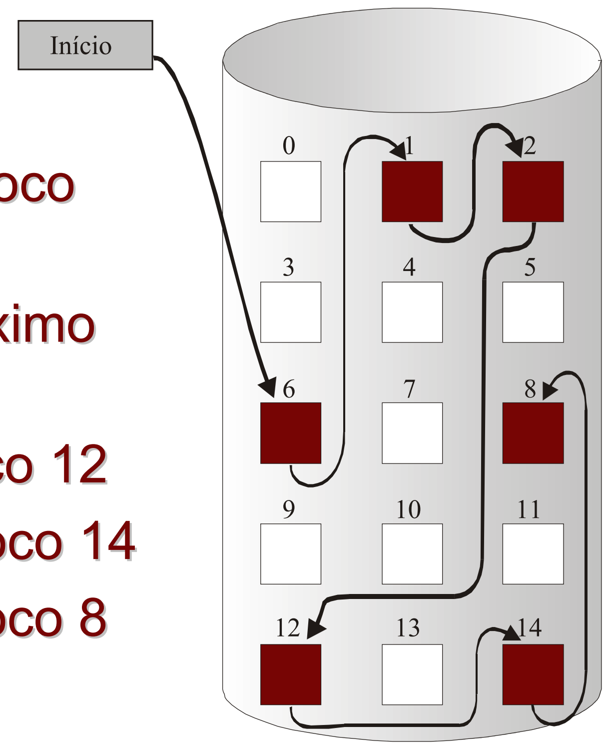
- Observar o arquivo A.TXT
  - ❖ Bloco inicial: 6
  - ❖ Final do bloco 6 possui ponteiro para próximo bloco (1)
  - ❖ Bloco 1 aponta para próximo bloco (2)
  - ❖ Bloco 2 aponta para bloco 12
  - ❖ Bloco 12 aponta para bloco 14



Arquivo	Bloco
A.TXT	6
...	...
...	...
...	...
...	...

Alocação Encadeada

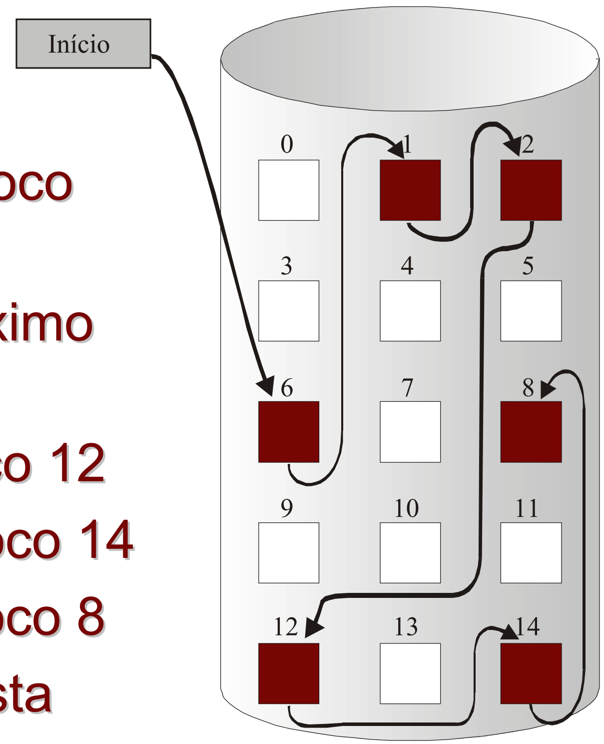
- Observar o arquivo A.TXT
  - ❖ Bloco inicial: 6
  - ❖ Final do bloco 6 possui ponteiro para próximo bloco (1)
  - ❖ Bloco 1 aponta para próximo bloco (2)
  - ❖ Bloco 2 aponta para bloco 12
  - ❖ Bloco 12 aponta para bloco 14
  - ❖ Bloco 14 aponta para bloco 8



Arquivo	Bloco
A.TXT	6
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

Alocação Encadeada

- Observar o arquivo A.TXT
  - ❖ Bloco inicial: 6
  - ❖ Final do bloco 6 possui ponteiro para próximo bloco (1)
  - ❖ Bloco 1 aponta para próximo bloco (2)
  - ❖ Bloco 2 aponta para bloco 12
  - ❖ Bloco 12 aponta para bloco 14
  - ❖ Bloco 14 aponta para bloco 8
  - ❖ Bloco 8 sinaliza fim de lista

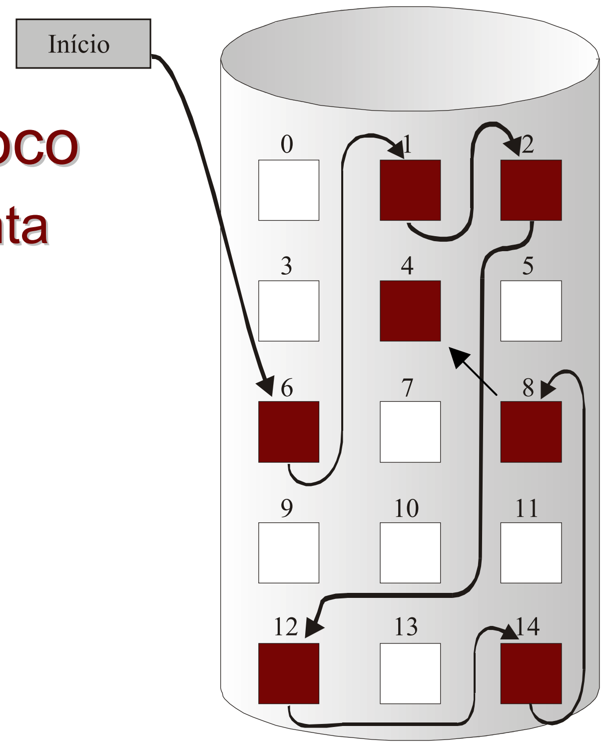


Arquivo	Bloco
A.TXT	6
...	...
...	...
...	...
...	...

Alocação Encadeada

- Observar o arquivo A.TXT

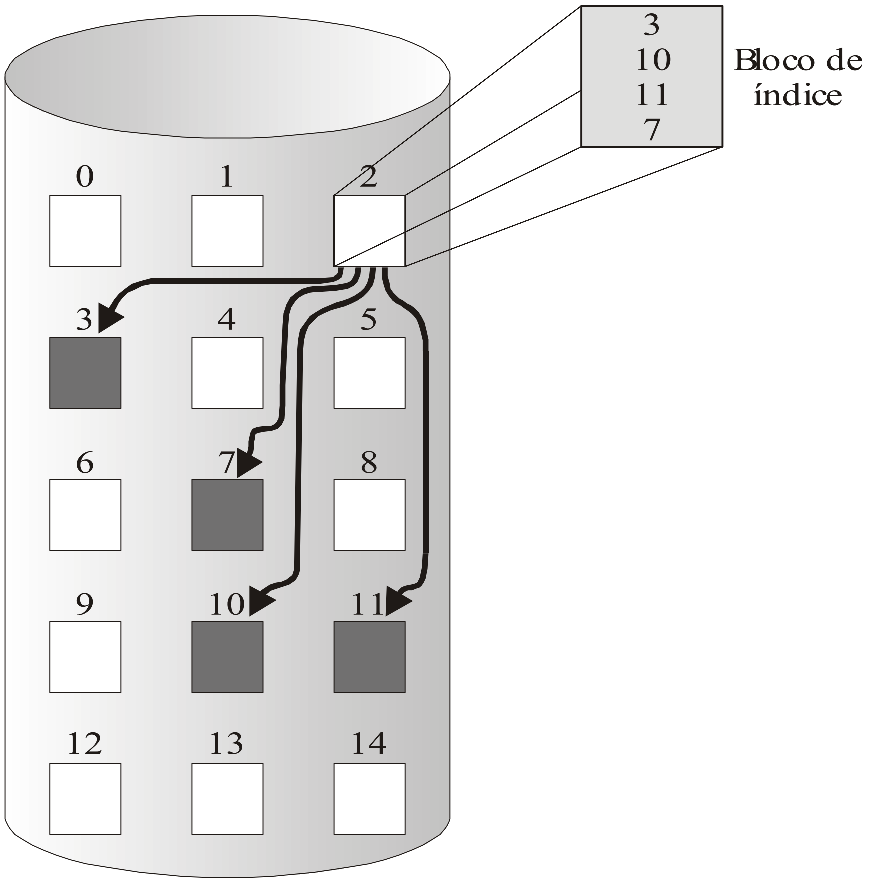
- Para incluir mais um bloco
  - ❖ Ponteiro do bloco 8 aponta para próximo bloco (4)



Arquivo	Bloco
A.TXT	6
...	...
...	...
...	...
...	...

Alocação Indexada

- FCB possui ponteiro para bloco de ponteiros
- Ponteiro do bloco aponta para:
  - Bloco de informações
  - Outro bloco de ponteiros
- Arquivo pode crescer/diminuir facilmente
- Perda de um bloco de ponteiro ocasiona perda parcial dos dados do arquivo



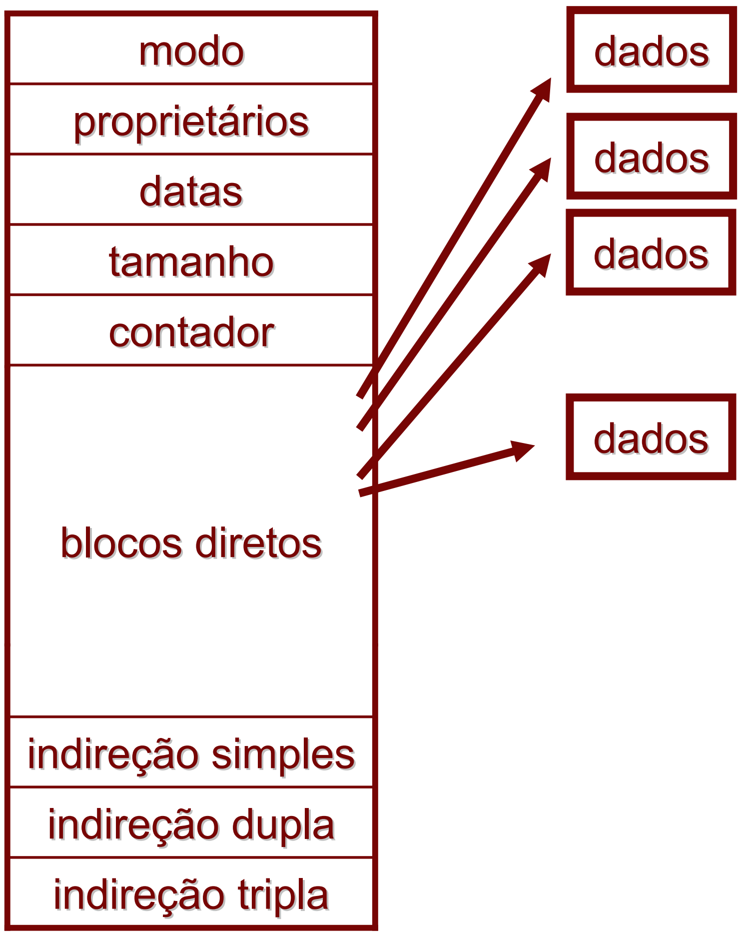
Alocação Indexada

- Exemplo - Unix

modo
proprietários
datas
tamanho
contador
blocos diretos
indireção simples
indireção dupla
indireção tripla

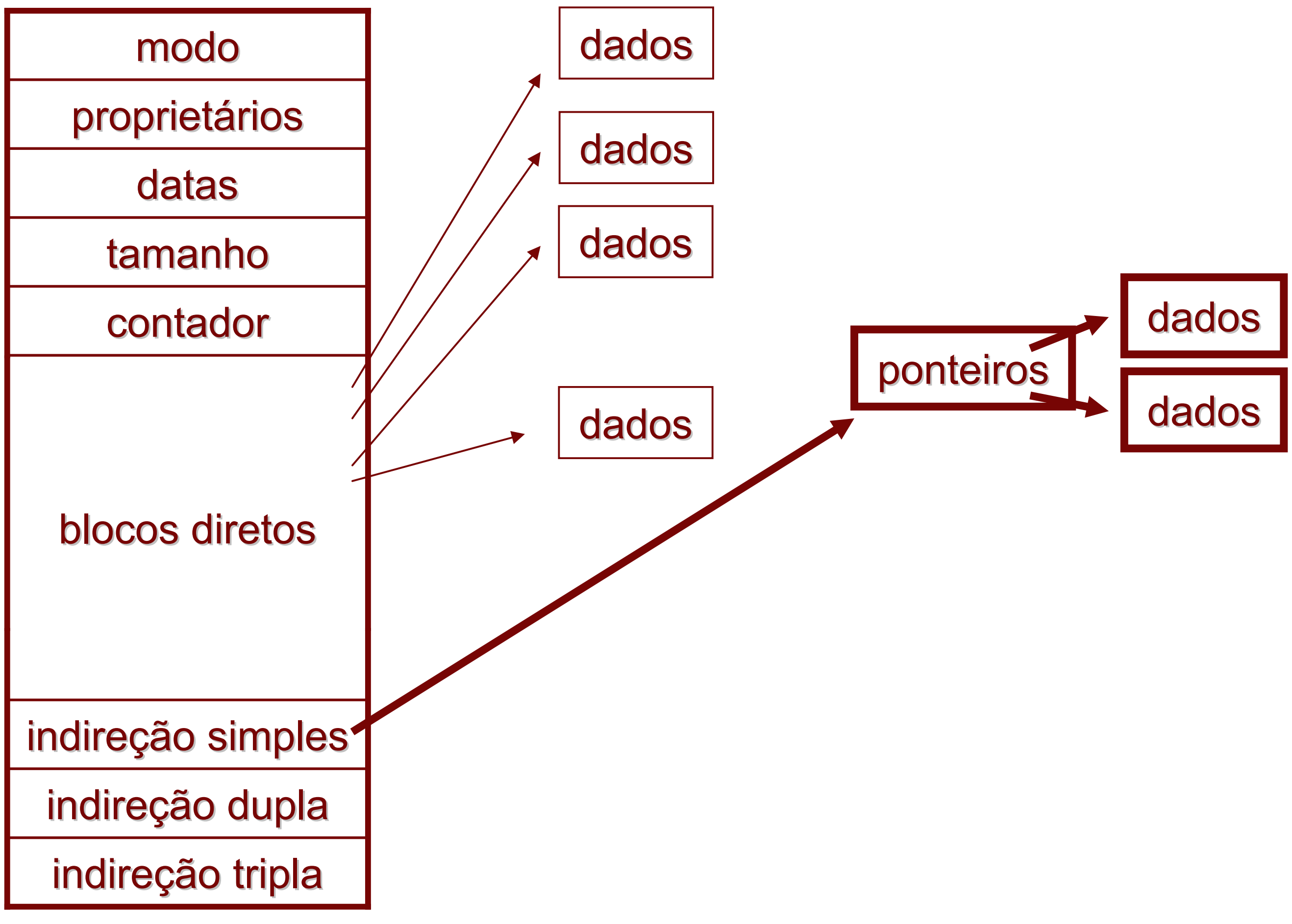
Alocação Indexada

- Exemplo - Unix



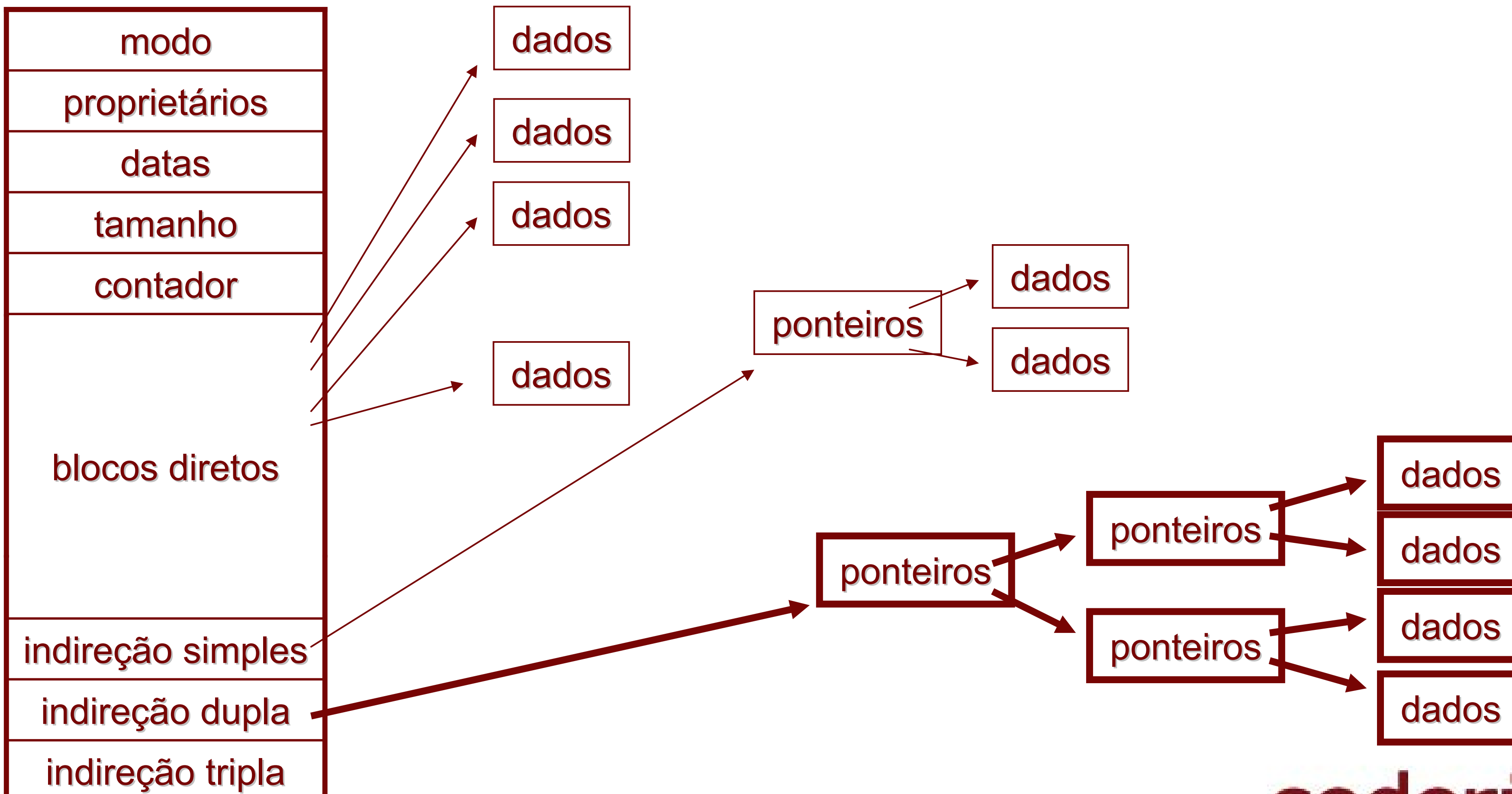
Alocação Indexada

- Exemplo - Unix



Alocação Indexada

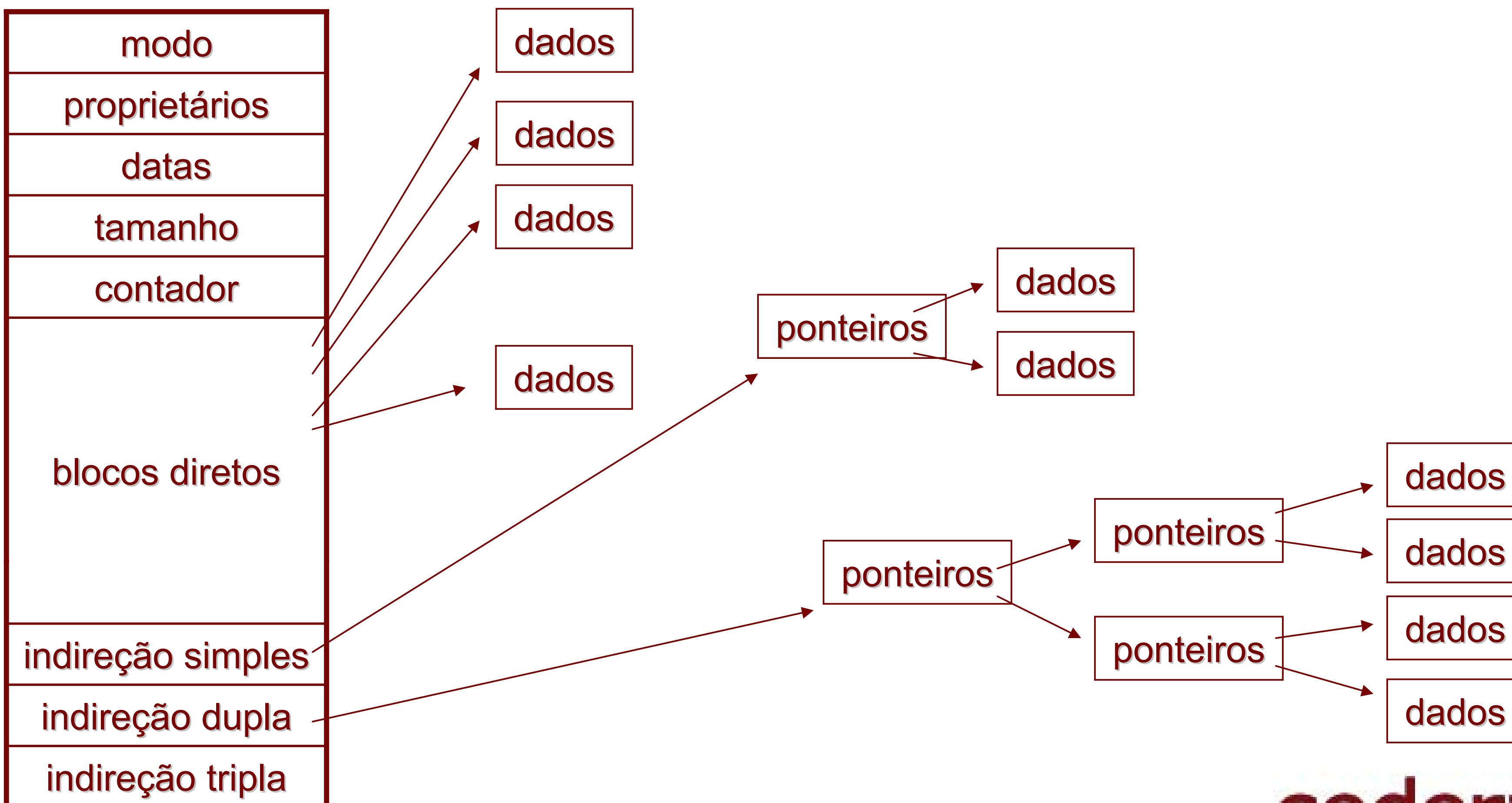
- Exemplo - Unix



cederj

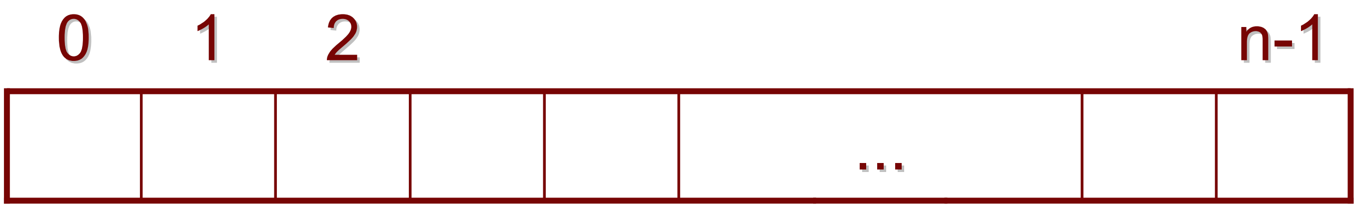
Alocação Indexada

- Exemplo - Unix



Gerência de Espaço Livre

- Vetor de bits
  - Um bit para cada bloco



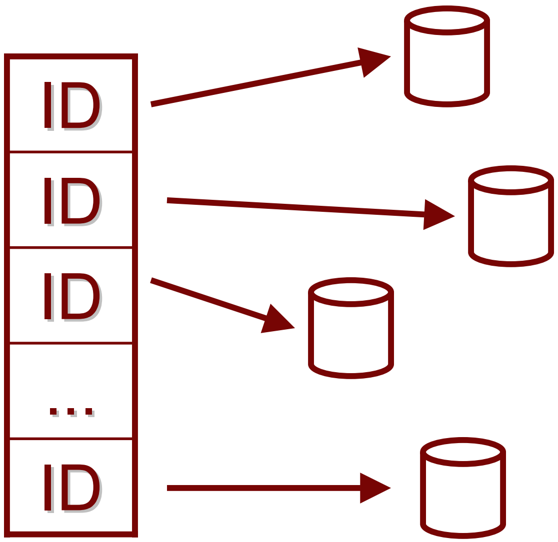
  - $\text{bit[i]} = \begin{cases} 0 \rightarrow \text{bloco i livre} \\ 1 \rightarrow \text{bloco i ocupado} \end{cases}$

Estrutura de Diretórios

- Diretório único
  - ❖ Usado em sistemas mais antigos
  - ❖ Todos os arquivos em um único lugar lógico

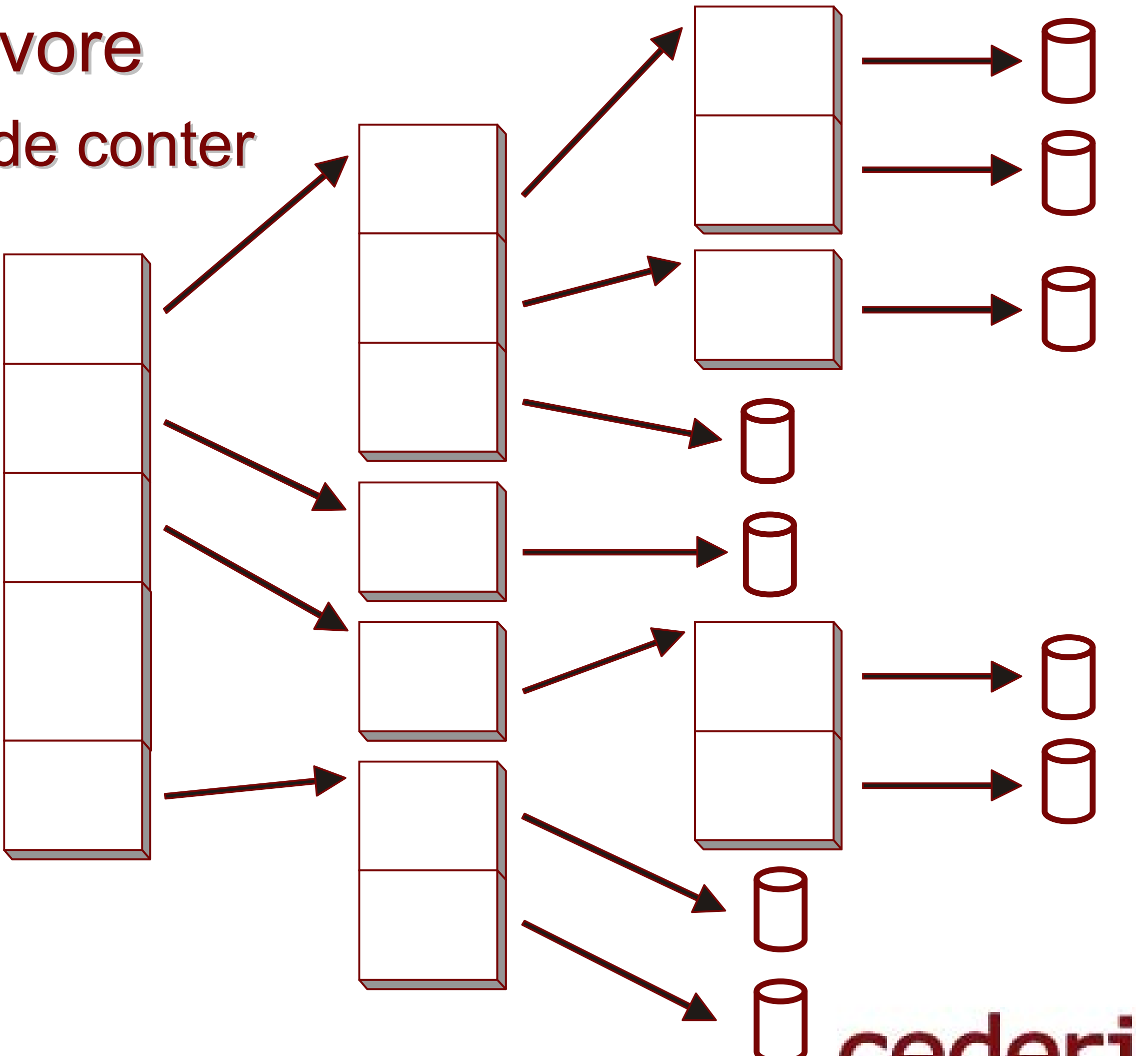
Estrutura de Diretórios

- Diretórios simples
  - ❖ Apenas um nível de diretório
  - ❖ Cada usuário tem um diretório



Estrutura de Diretórios

- Diretórios em árvore
  - Um diretório pode conter
    - Outro diretório
    - Um arquivo



cederj

Tempo de Acesso ao Disco

- Disco muito lento
  - Tempo de acesso na ordem de milisegundos
  - Comparado com a memória principal
    - Tempo de acesso na ordem de até dezenas de nanosegundos
  - Comparado com a CPU
    - Tempo de ciclo na ordem de centenas de picosegundos

Segurança da Informação

- Disco composto por diversos componentes mecânicos

- Sujeito a falhas

- Sistema muito frágil

Solução

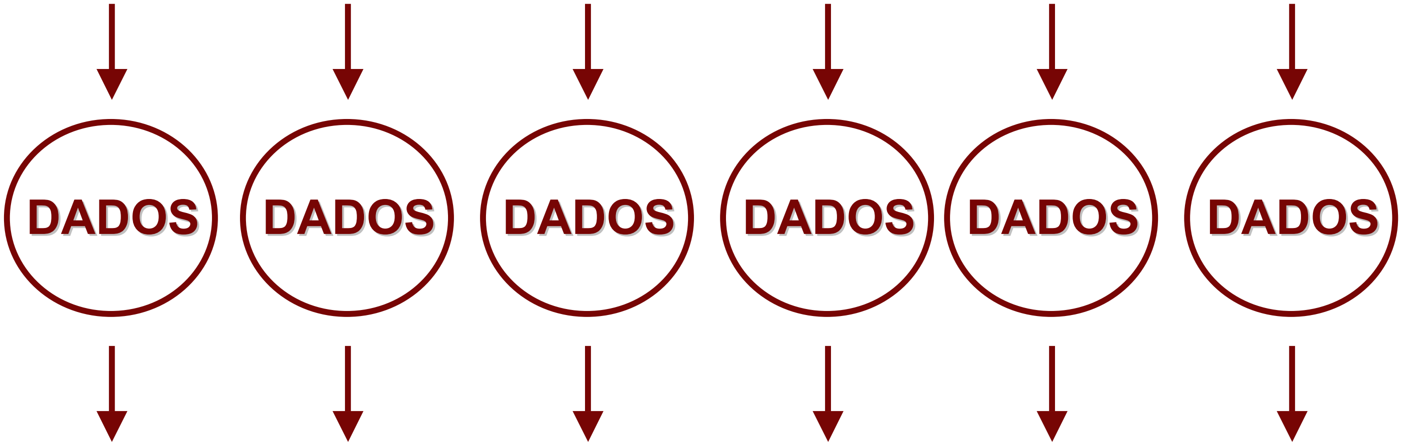
- Backups periódicos
  - Solução de problemas de integridade da informação
- Redundância
  - Resolve integridade e velocidade

O que é RAID

- RAID – Redundant Array of Independent Disks (Conjunto redundante de discos independentes)
- Combinação de conjunto de discos de custo relativamente barato
- O sistema computacional percebe o sistema RAID como um único disco
- Pode proporcionar tolerância a falhas
- Diversas implementações com características próprias
- Em comum: a divisão de dados por diversos discos

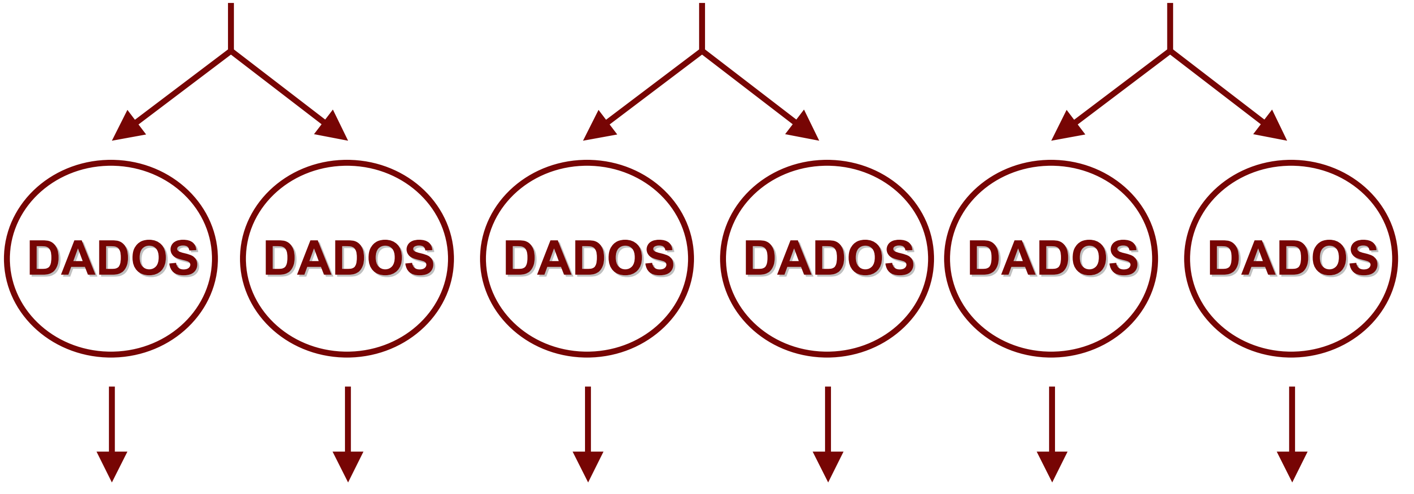
RAID 0

- Divide a informação em diversos discos
- Sem paridade
- Sem redundância
- Permite escritas e leituras simultâneas em cada disco



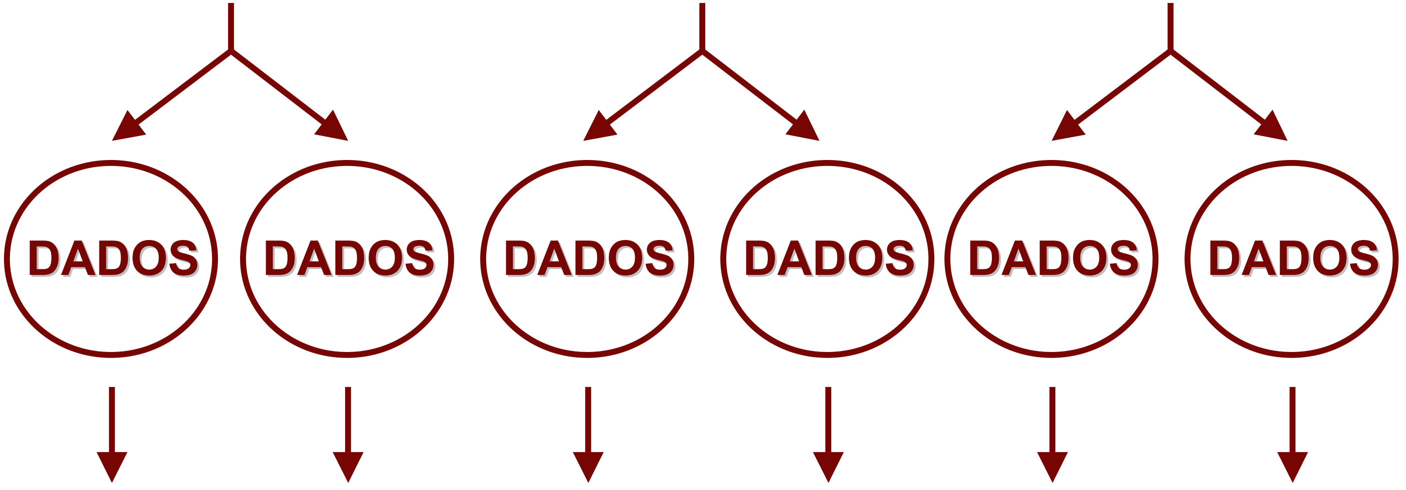
RAID 1

- Espelhamento de disco
- Tolerante a falhas
- Dados duplicados são gravados em pares de discos



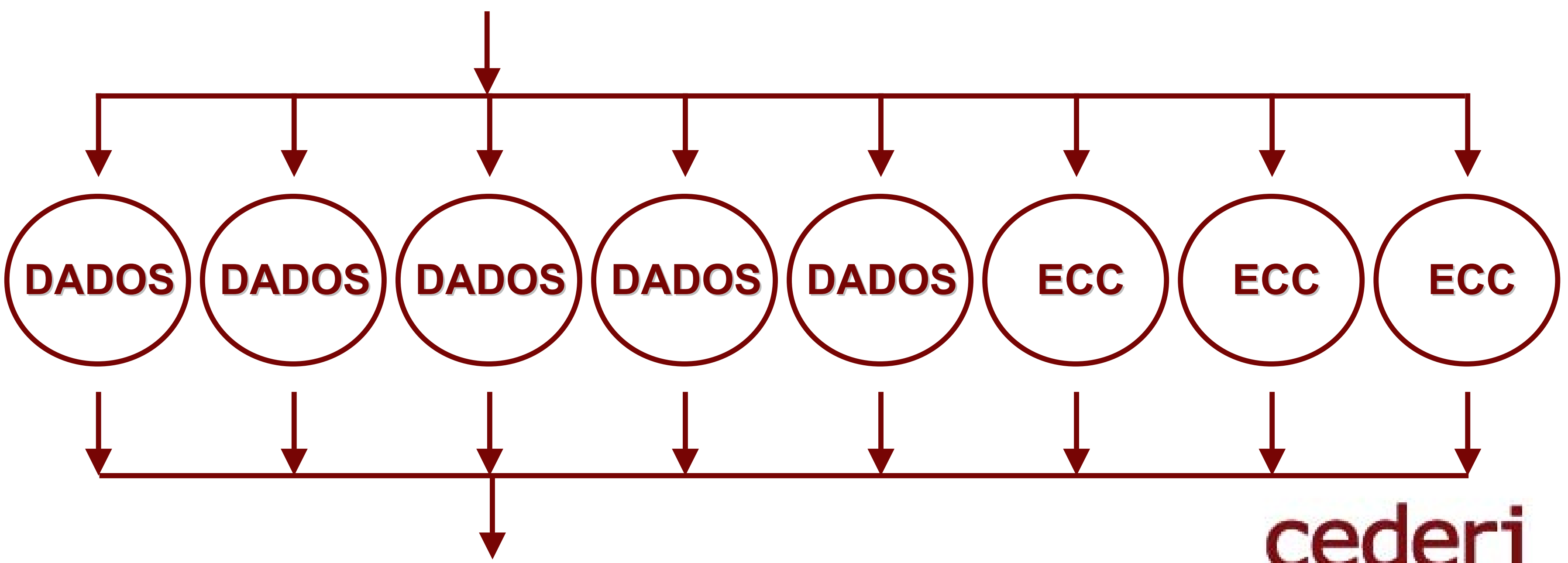
RAID 1

- Escritas precisam ser realizadas em ambos os discos
- Permite leituras simultâneas em todos os discos
- Melhor implementados se houver duas controladoras de disco
- Pode ser implantado com apenas dois discos



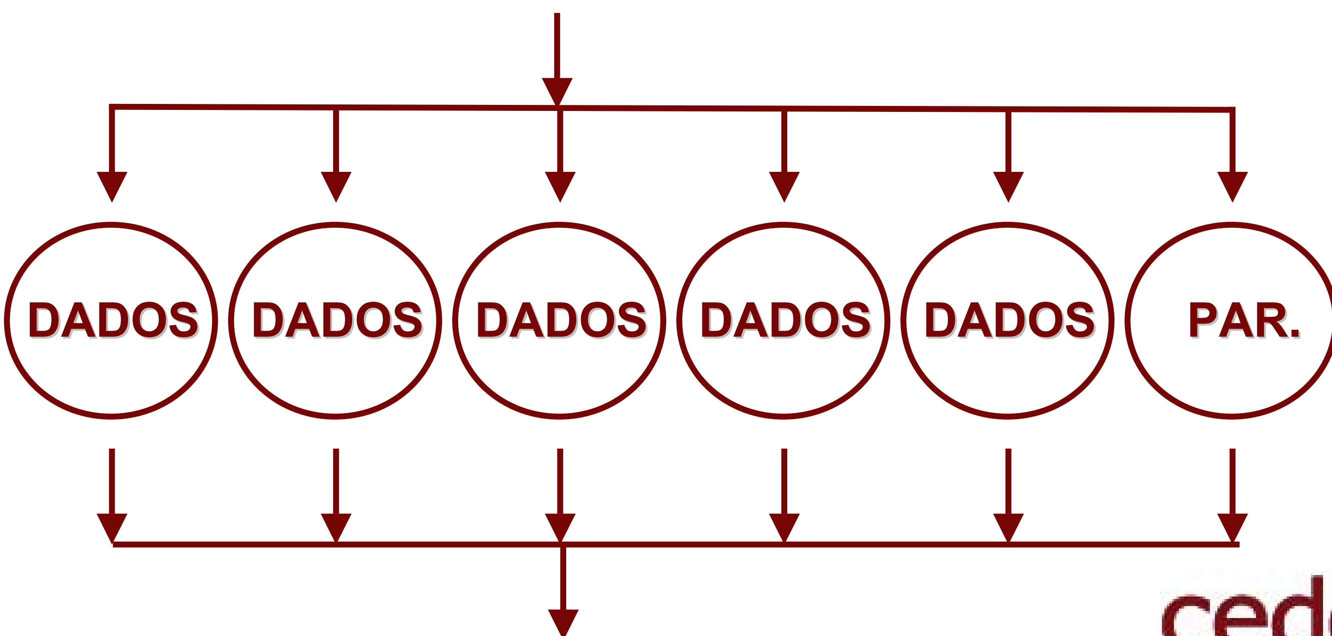
RAID 2

- Grava dados por vários discos
- Possui verificação/correção de erros
- Dispendioso



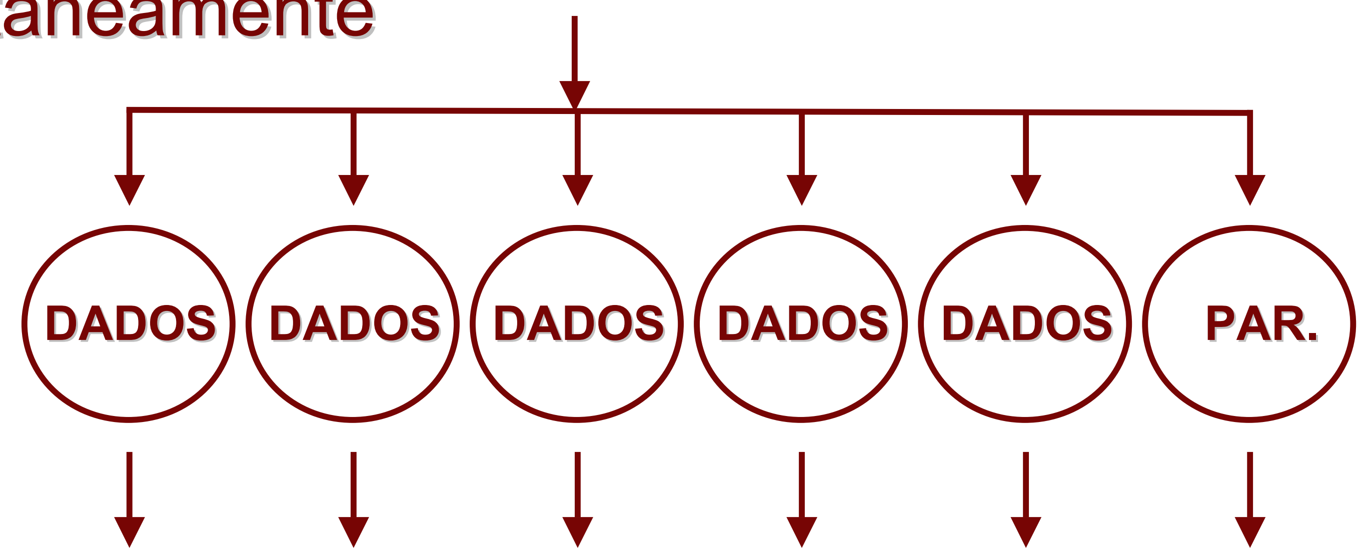
RAID 3

- Grava dados por vários discos
- Possui verificação de erros
- Registros espalhados pelos discos
- Leitura de cada registro pode ser paralelizada



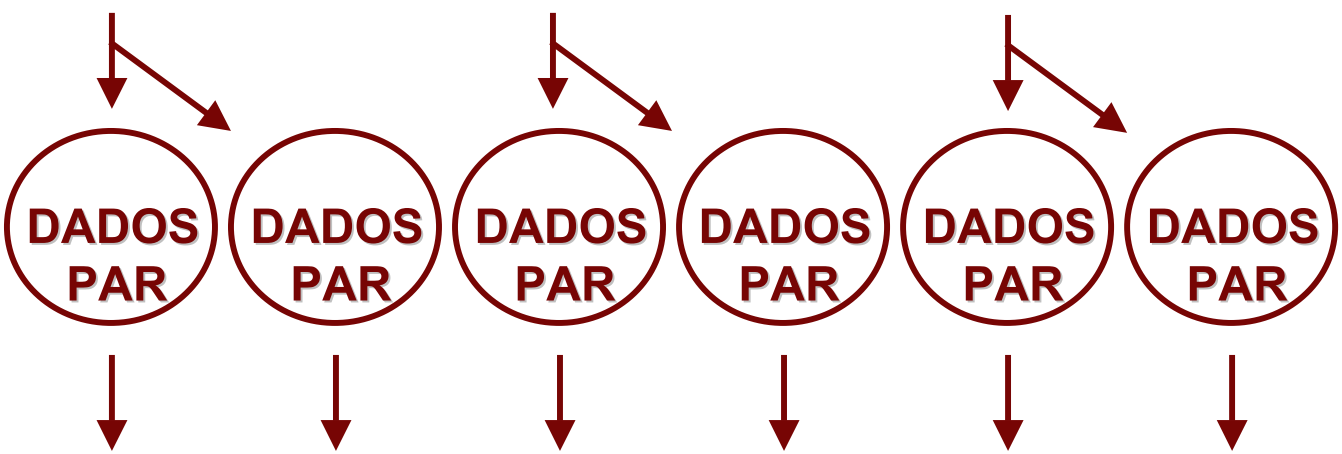
RAID 4

- Idêntico ao RAID 3
- Diferença apenas na divisão da informação
  - ❖ Registros inteiros em cada disco
- Possibilita a leitura de diversos registros simultaneamente



RAID 5

- Dados e paridade espalhados por todos os discos
- Leituras e escritas pode ser realizadas em paralelo
- Necessita de pelo menos 3 discos



RAID em Resumo

- RAIDs mais utilizados
  - ❖RAID 0, 1 e 5
- RAID 0 é o mais rápido e mais eficiente, mas não é tolerante a falhas
- RAID 1 é a melhor escolha para tolerância a falhas com desempenho e custo
- RAID 2, 3 e 4 não são muito utilizados
- RAID 5 combina eficiência, tolerância a falhas e bom desempenho

Próxima Aula

- Exercícios de fixação

cederj