DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

ARQUITETURA DE COMPUTADORES

PARTE 1

Prof. Everson Pedro Burg

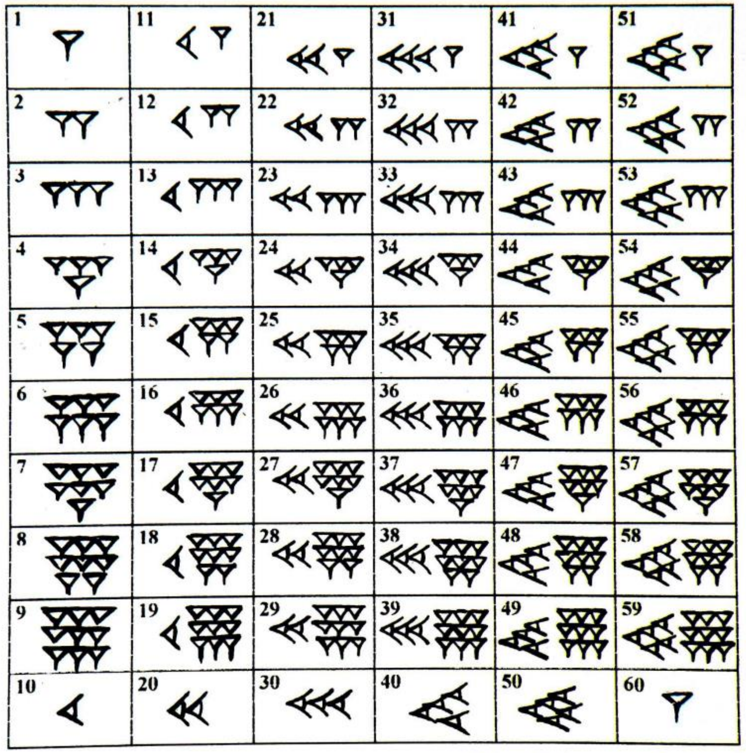
FURB
UNIVERSIDADE DE BLUMENAU

1. SISTEMAS DE NUMERAÇÃO

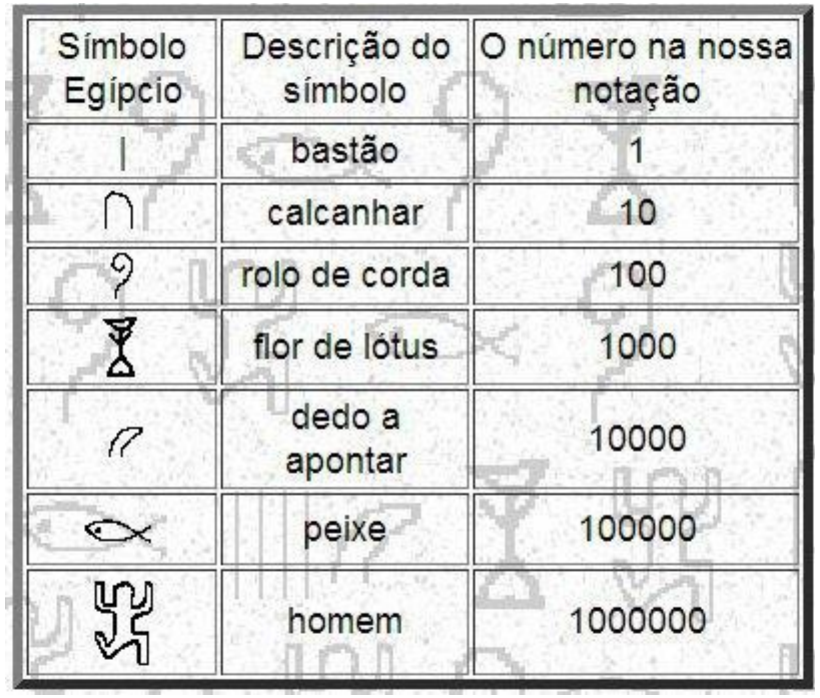
- **Sistemas de numeração ➔** A técnica para representar e trabalhar com números é chamada de sistema numérico.

- Diversos sistemas de numeração foram criados durante a história das civilizações

Sumério



Egípcio



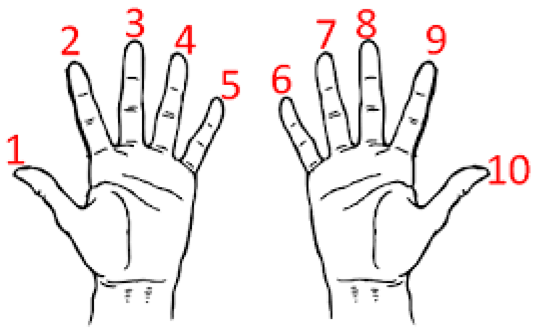
Símbolo Egípcio	Descrição do símbolo	O número na nossa notação
	bastão	1
	calcanhar	10
	rolo de corda	100
	flor de lótus	1000
	dedo a apontar	10000
	peixe	100000
	homem	1000000

1. SISTEMAS DE NUMERAÇÃO

- O sistema de numeração decimal é um dos sistemas de numeração mais comuns

- Outros sistemas de numeração populares incluem sistema de numeração binário, sistema de numeração octal, sistema de numeração hexadecimal, etc

- Tais sistemas foram/são bastante comum na área da computação



1.1 SISTEMAS DE NUMERAÇÃO DECIMAL

- **Sistema Numérico Decimal** ➔ O sistema numérico decimal é um sistema numérico de **base 10** com 10 dígitos de 0 a 9
- Qualquer quantidade numérica pode ser representada usando esses 10 dígitos
- O sistema de numeração decimal também é um sistema de valores posicionais, ou seja, o valor dos dígitos dependerá de sua posição

Exemplo: 126, 367 e 694. O número 6, presente nestes 3 números, possui valores diferentes

126 ➔ 6 unidades ou 6 x 1;

367 ➔ 6 dezenas ou 6 x 10

694 ➔ 6 centenas ou 6 x 100

1.1 SISTEMAS DE NUMERAÇÃO DECIMAL

- O peso de cada posição pode ser representado da seguinte forma

. . .	10^5	10^4	10^3	10^2	10^1	10^0

Exemplo: $5269 = 5 \times 10^3 + 2 \times 10^2 + 6 \times 10^1 + 9 \times 10^0$

- Caso de números depois da vírgula

10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	...

- Exemplo: 12,584 $=1 \times 10^1 + 2 \times 10^0 + 5 \times 10^{-1} + 8 \times 10^{-2} + 4 \times 10^{-3}$

Décimo Centésimo Milésimo

1.1 SISTEMAS DE NUMERAÇÃO DECIMAL

- Nos sistemas digitais, as instruções são dadas por meio de sinais elétricos e utilizar o sistema decimal, não seria fácil de implementar. Assim, outros sistemas numéricos que são mais fáceis de implementar digitalmente foram desenvolvidos
- Em computadores digitais, a maneira mais fácil de variar as instruções através de sinais elétricos é o sistema de dois estados – ligado e desligado. *On* é representado como 1 e *off* como 0
- O sistema numérico que possui apenas esses dois dígitos (0 e 1) é chamado de sistema numérico binário

1.2 SISTEMAS DE NUMERAÇÃO BINÁRIO

- **Sistema Binário** ➔ O sistema binário, ou base 2, é um sistema de numeração posicional em que todas as quantidades se representam com base em dois números, isto é, 0 e 1

- Em geral, em sistemas digitais, o sinal de tensão é utilizado para definir o estado, isto é, com tensão (Vcc) nível lógico 1, sem tensão (0 V) nível lógico 0

- Cada dígito binário também é chamado de bit ➔ Exemplo, o número 1011 possui 4 bits, já o número 01100111 possui 8 bits

- Em qualquer número binário, o dígito mais à direita é chamado de bit menos significativo (LSB) e o dígito mais à esquerda é chamado de bit mais significativo (MSB) ➔ 1 0 0 1 0 0 1 0

MSB LSB

1.2 SISTEMAS DE NUMERAÇÃO BINÁRIO

- O sistema numérico binário também é um sistema de valores posicionais, onde cada dígito tem um valor expresso em potências de 2

...	2^5	2^4	2^3	2^2	2^1	2^0

Exemplo: $1001 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

- E o equivalente decimal deste número é a soma do produto de cada dígito com seu valor posicional

$$1001 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$1001 = 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$$

$$1001_2 = 8 + 0 + 0 + 1 = 9_{10}$$

1.2 SISTEMAS DE NUMERAÇÃO BINÁRIO

- Número de bits x Combinações

Bits	Combinações	Binários
1	$2^1 = 2$	0 e 1
2	$2^2 = 4$	00, 01, 10, 11
3	$2^3 = 8$	000, 001, 010, 011, 100, 101, 110, 111
4	$2^4 = 16$	0000 a 1111
5	$2^5 = 32$	00000 a 11111
6	$2^6 = 64$	000000 a 111111
7	$2^7 = 128$	0000000 a 1111111
8	$2^8 = 256$	00000000 a 11111111
9	$2^9 = 512$	000000000 a 111111111
10	$2^{10} = 1024$	000000000 a 111111111

Quanto maior o número de bits, maior a quantidade de informação que pode ser armazenada

Ex: Profundidade de cor em TVs digitais

1.2 SISTEMAS DE NUMERAÇÃO BINÁRIO

- Unidades binárias ➔ As unidades binárias visam simplificar a notação de números binários. Por exemplo, a memória do computador é medida em termos de quantos bits ela pode armazenar

1 bit (b)	bit que é a menor unidade de informação que pode ser armazenada ou transmitida
1 byte (B)	8 bits
1 Kilobyte (KB)	1024 bytes
1 Megabyte (MB)	1024 KB
1 Gigabyte (GB)	1024 MB
1 Terabyte (TB)	1024 GB
1 Petabyte (PB)	1024 TB
1 Exabyte (EB)	1024 PB
⋮	⋮

Ex: Se cada letra do computador é armazenada a 8 bits, ou seja, 1 byte, a palavra ARTE utiliza 32 bits ou 4 bytes

Qual a capacidade máxima de memória de um sistema operacional de 32 bits?

1.3 SISTEMAS DE NUMERAÇÃO OCTAL

- **Sistema Octal ➔** O sistema octal, ou base 8, é um sistema de numeração posicional no qual usa 8 símbolos para sua representação, isto é, 0, 1, 2, 3, 4, 5, 6 e 7

- Também é um sistema posicional e durante algum tempo, o sistema octal foi bastante utilizado em computadores como uma alternativa de compactação de números binários

...	8^5	8^4	8^3	8^2	8^1	8^0

Exemplo: $721_8 = 7 \times 8^2 + 2 \times 8^1 + 1 \times 8^0$

$$721_8 = 7 \times 64 + 2 \times 8 + 1 \times 1$$

$$721_8 = 448 + 16 + 1$$

$$721_8 = 465_{10}$$

1.4 SISTEMAS DE NUMERAÇÃO HEXADECIMAL

- **Sistema Hexadecimal ➔** O sistema hexadecimal, ou base 16, é um sistema de numeração posicional no qual usa 16 símbolos para sua representação e são utilizados como opção de compactação de notação binária

- O símbolos são os numerais 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9 e as letras A, B, C, D, E, F representam os números 10, 11, 12, 13, 14, 15 respectivamente

. . .	16^5	16^4	16^3	16^2	16^1	16^0

Exemplo:

$$12FC_{16} = 1 \times 16^3 + 2 \times 16^2 + 15 \times 16^1 + 12 \times 16^0$$

$$12FC_{16} = 4860_{10}$$

1.5 CONVERSÃO DE BASES

Decimal	Binário	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Pode-se representar um número em qualquer base bem como converter um número em uma base para outra

1.5 CONVERSÃO DE BASES

➢ **Binário ↔ Decimal**

a) Binário → Decimal ➔ Soma do produto de cada dígito com seu valor posicional

Exemplo:

$$10010001_2 = 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^0 = 145_{10}$$

Qual o número 110101 em decimal?

1.5 CONVERSÃO DE BASES

b) Decimal → Binário ➔ Dividir sucessivamente o número decimal por 2 até que o quociente seja 0.

Ex: Qual o valor de 145 (base 10) em binário?

Divisão	Resultado	Resto
145/2	72	1
72/2	36	0
36/2	18	0
18/2	9	0
9/2	4	1
4/2	2	0
2/2	1	0
		1

LSB

Lê-se de baixo para cima

MSB

$$145_{10} = 10010001_2$$

1.5 CONVERSÃO DE BASES

- **Decimal ↔ Octal**

a) Octal → Decimal ➔ Soma do produto de cada dígito com seu valor posicional

Exemplo:

$$731_8 = 7 \times 8^2 + 3 \times 8^1 + 1 \times 8^0 = 473_{10}$$

1.5 CONVERSÃO DE BASES

b) Decimal → Octal ➔ Dividir sucessivamente o número decimal por 8 até que o quociente seja 0

Ex: Qual o valor de 473 (base 10) em octal (base 8)?

Divisão	Resultado	Resto
473/8	59	1
59/8	7	3
7/8	0	7

Lê-se de baixo para cima

$$473_{10} = 731_8$$

1.5 CONVERSÃO DE BASES

➢ **Decimal ↔ Hexadecimal**

a) Hexadecimal → Decimal ➔ Soma do produto de cada dígito com seu valor posicional

Exemplo:

$$5F3B_{16} = 5 \times 16^3 + 15 \times 16^2 + 3 \times 16^1 + 11 \times 16^0 = 24379_{10}$$

1.5 CONVERSÃO DE BASES

b) Decimal → Hexadecimal → Dividir sucessivamente o número decimal por 16 até que o quociente seja 0

Ex: Qual o valor de 24379 (base 10) em hexadecimal (base 16)?

Divisão	Resultado	Resto
24379/16	1523	11 (B)
1523/16	95	3
95/16	5	15 (F)
5/16	0	5

Lê-se de baixo para cima

$$24379_{10} = 5F3B_{16}$$

1.5 CONVERSÃO DE BASES

➢ **Binário ↔ Octal**

a) Binário → Octal → Para converter binário em octal siga os passos a seguir:

1) A partir do bit menos significativo, faça grupos de três bits. Se precisar, acrescente zeros para completar o grupo;

2) Converta cada grupo em seu número octal equivalente

Exemplo: Qual o valor de 10110011 em octal?

$$\underbrace{010}_{\underline{2}}\ \underbrace{110}_{\underline{6}}\ \underbrace{011}_{\underline{3}}$$

$$10110011_2 = 263_8$$

1.5 CONVERSÃO DE BASES

b) Octal → Binário → Para converter um número octal em binário, cada dígito octal é convertido em seu equivalente binário de 3 bits

Exemplo: Qual o valor de 263 (base 8) em binário?

$$263_8 \rightarrow \begin{matrix} 2 = 010 \\ 6 = 110 \\ 3 = 011 \end{matrix}$$

Agrupando os valores tem-se

$$263_8 = 10110011_2$$

1.5 CONVERSÃO DE BASES

➢ Binário ↔ Hexadecimal

a) Binário → Hexadecimal → Para converter binário em hexadecimal siga os passos a seguir:

1) A partir do bit menos significativo, faça grupos de quatro bits. Se precisar, acrescente zeros para completar o grupo;

2) Converta cada grupo em seu número hexadecimal equivalente

Exemplo: Qual o valor de 110001001110 em hexadecimal?

$$\begin{array}{ccc} 1100 & 0100 & 1110 \\ \longleftrightarrow & \longleftrightarrow & \longleftrightarrow \\ \downarrow & \downarrow & \downarrow \\ \overline{C} & \overline{4} & \overline{E} \end{array}$$

$$110001001110_2 = C4E_{16}$$

1.5 CONVERSÃO DE BASES

b) Hexadecimal → Binário ➔ Para converter um número hexadecimal em binário, cada dígito hexadecimal é convertido em seu equivalente binário de 4 bits

Exemplo: Qual o valor de C4E (base 16) em binário?

$$C4E_{16} \rightarrow \begin{array}{l} C = 1100 \\ 4 = 0100 \\ E = 1110 \end{array}$$

Agrupando os valores tem-se

$$C4E_{16} = 110001001110_2$$

1.6 PADRÕES INFORMÁTICA

- **Tabela ASCII ➔** Código criado na década de 60 por Robert W. Bemer para unificar a representação de caracteres alfanuméricos em computadores
- ASCII significa “American Standard Code for Information Interchange” ou ”Código Padrão Americano para o Intercâmbio de Informação”
- Sua função é padronizar a forma como os computadores representam letras, números, acentos, sinais diversos e alguns códigos de controle
- Vai do número 0 até 127, sendo que os 32 primeiros e o último são considerados de controle, os demais representam "caracteres imprimíveis"
- Alguns caracteres caíram em desuso, como o Line Feed que fazia a impressora avançar o papel

1.6 PADRÕES INFORMÁTICA

ASCII Table

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

Source: www.LookupTables.com

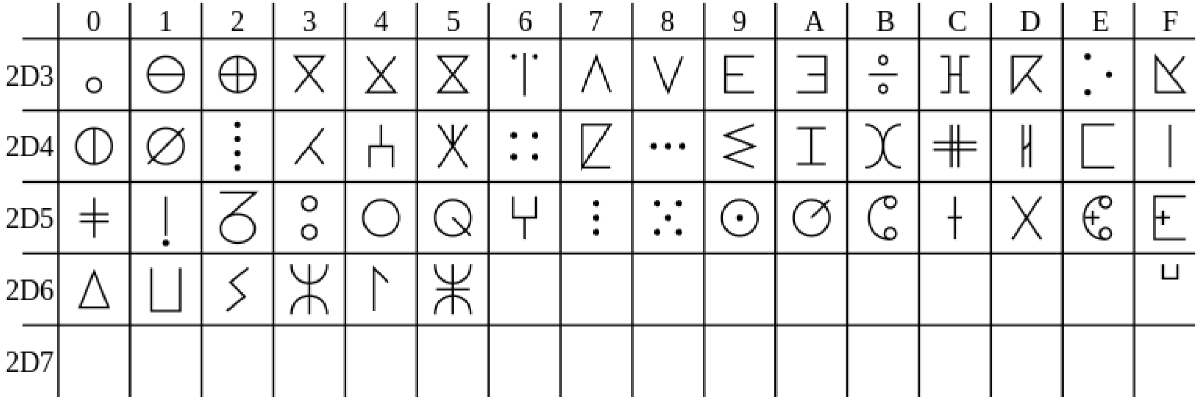
1.6 PADRÕES INFORMÁTICA

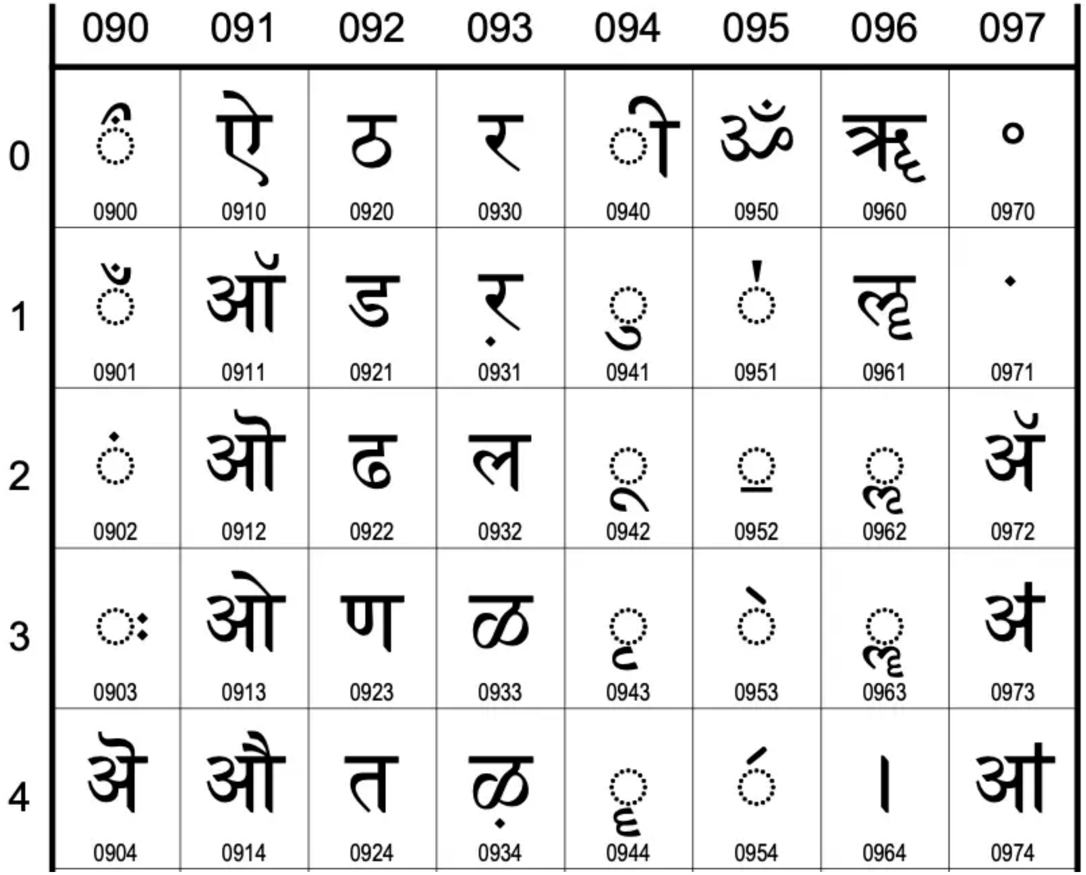
- **Tabela UNICODE ➔** Como o código ASCII possui apenas 128 códigos, na década de 90 grandes empresas da área de informática se uniram para ampliar a quantidade de códigos, incluindo caracteres específicos de diversos idiomas

- A tabela completa possui 32 bits, ou seja, 4 bytes (4 caracteres hexadecimal – 0000 a FFFF) e inclui a Tabela ASCII nos seus primeiros 127 caracteres (0000 a 007F)

- Possui três formatos padrões UTF-8, UTF-16 e UTF-32 visando otimizar o tamanho do código entre 1 e 4 bytes.

1.6 PADRÕES INFORMÁTICA





1.7 ARITMÉTICA DE BASES NUMÉRICAS

- **Operações números binários** → de forma análoga aos números decimais, pode-se fazer operações matemáticas como somar, subtrair, multiplicar, dividir, etc diretamente com números binários

a) Soma de números binários

Regras:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 1 = 0 \text{ (Carrega 1 para o dígito de ordem superior)}$$
$$1 + 1 + 1 = 1 \text{ (Carrega 1 para o dígito de ordem superior)}$$

$$\begin{array}{r} 0101 \\ +\,0010 \\ \hline 0111 \end{array} \qquad \begin{array}{r} 0011 \\ +\,0110 \\ \hline 1001 \end{array}$$

1.7 ARITMÉTICA DE BASES NUMÉRICAS

b) Subtração de números binários

Regras:

$$0 - 0 = 0$$

$$0 - 1 = 1 \quad \text{(Pega 1 "emprestado" do dígito de ordem superior, o que equivale a 2)}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$\begin{array}{r} 0111 \\ -\ 0011 \\ \hline 0100 \end{array} \qquad\qquad \begin{array}{r} 0\cancel{1}\overset{2}{\cancel{1}}\overset{2}{0} \\ -\ 0011 \\ \hline 0011 \end{array}$$

1.7 ARITMÉTICA DE BASES NUMÉRICAS

b) Subtração de números binários

- Uma opção é usar complemento de 2 do numero binário (negativo do número), para isto, faço do complemento de um (negar o número) e somar 1

- A – B = A + (-B)

$$\begin{array}{r} 0111 \\ -\,0011 \\ \hline ? \end{array}$$

$$\text{Complemento de 2 de } 0011 = 1100 + 0001 = 1101$$

$$\begin{array}{r} 0111 \\ +\,1101 \\ \hline 0100 \end{array}$$

1.7 ARITMÉTICA DE BASES NUMÉRICAS

- **Operações números hexadecimais ➔** de forma análoga aos números decimais e binários, pode-se fazer operações matemáticas como somar, subtrair, multiplicar, dividir, etc diretamente com números hexadecimais

a) Soma de números hexadecimais => Somar os elementos e caso "estoure" a contagem, carregar 1 para a próxima coluna

$$\begin{array}{r} 5C12 \\ +\ 27A4 \\ \hline 83B6 \end{array}$$

1.7 ARITMÉTICA DE BASES NUMÉRICAS

b) Subtração de números hexadecimais => Subtrair os elementos e caso "falte", pegar carregar 1 para a próxima coluna (o que equivale a 16)

$$\begin{array}{r} {}^{1\,1} \\ 5C12 \\ -\ 27A4 \\ \hline 346E \end{array}$$