

Figuras do Capítulo 1

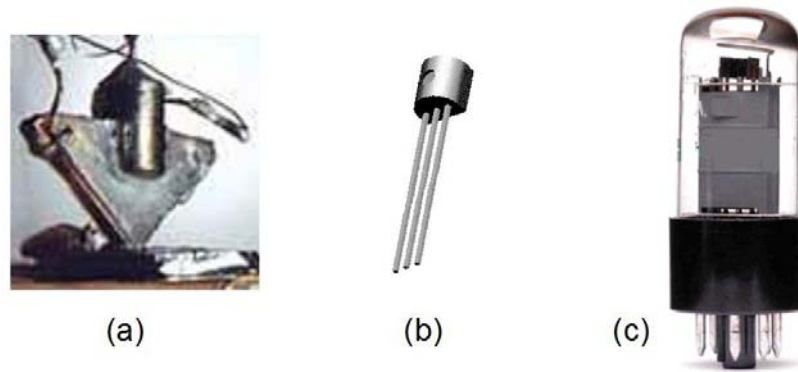


Figura 1.1

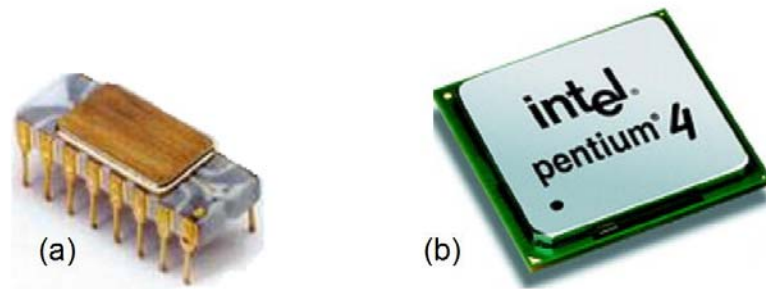


Figura 1.2

Figuras do Capítulo 1

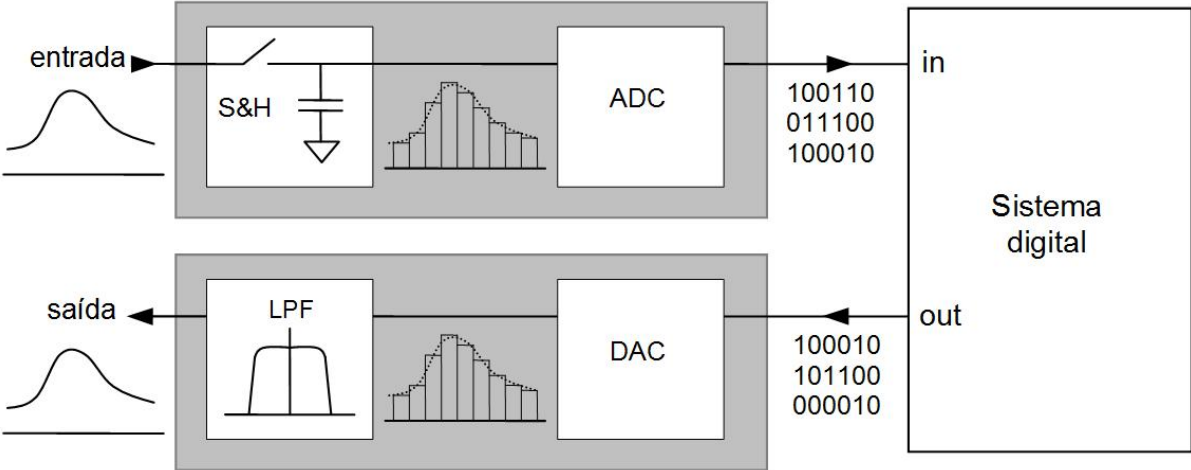


Figura 1.3

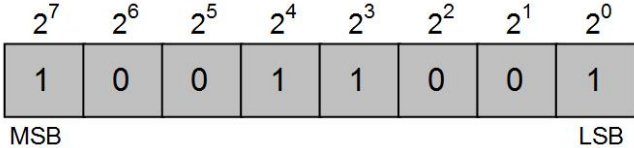


Figura 1.4

Figuras do Capítulo 1

b ₃ b ₂ b ₁ b ₀	b ₇ b ₆ b ₅							
	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Figura 1.5

Figuras do Capítulo 1

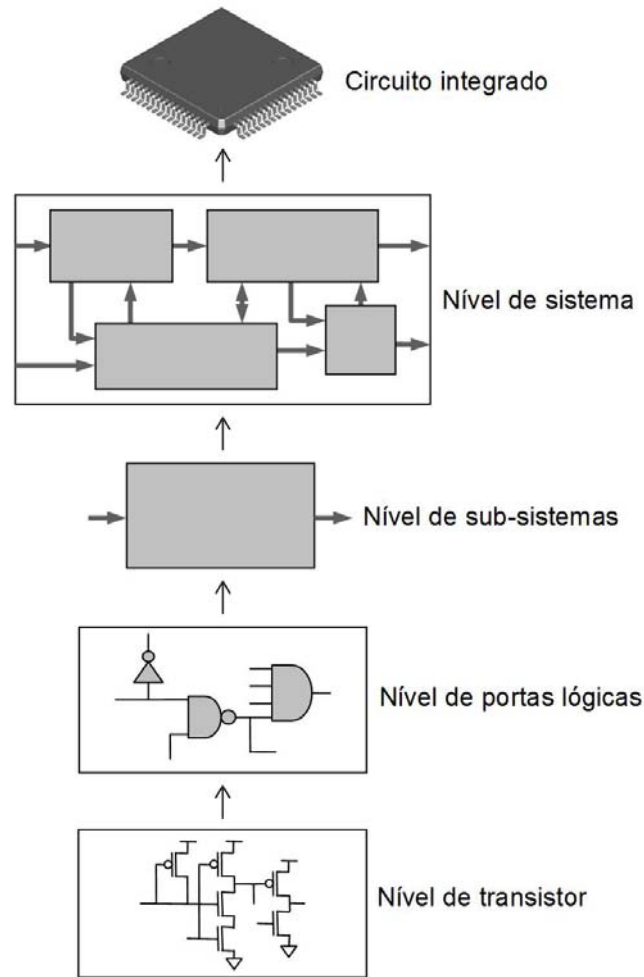


Figura 1.6

Figuras do Capítulo 1

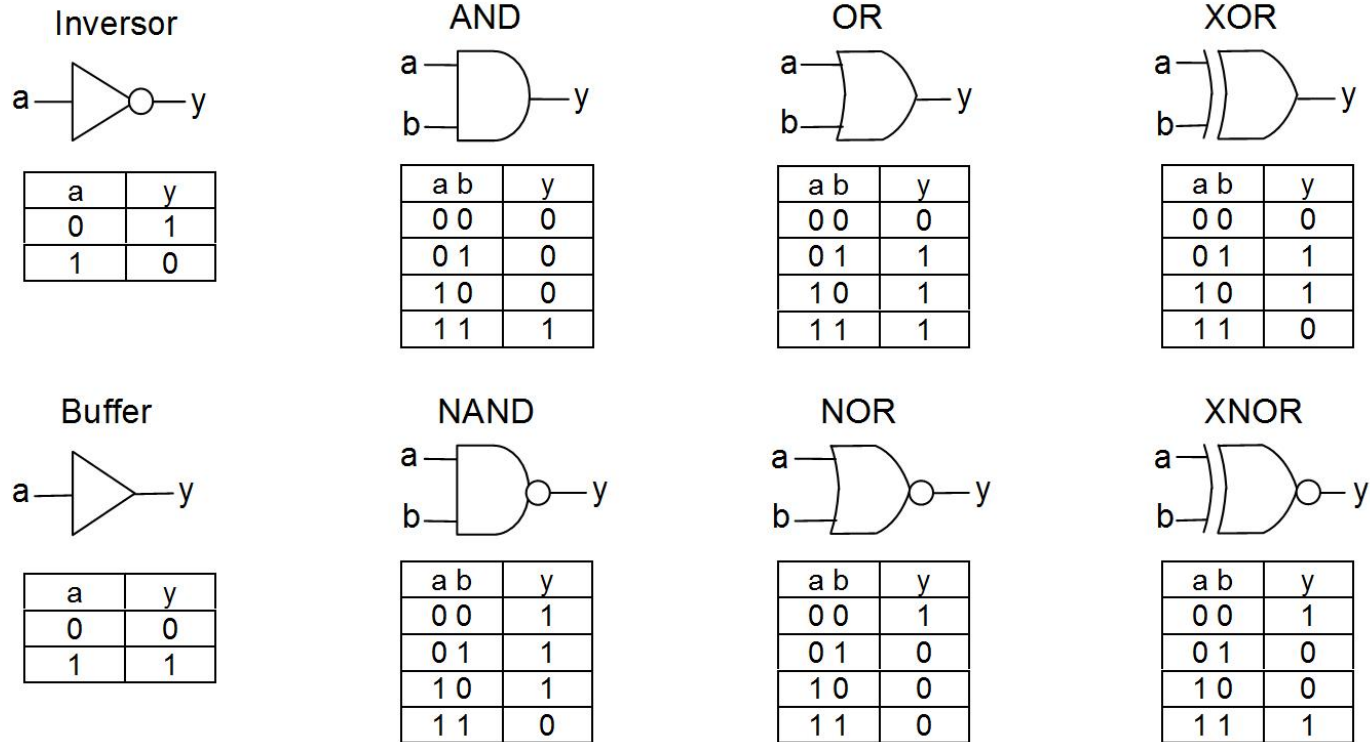


Figura 1.7

Figuras do Capítulo 1

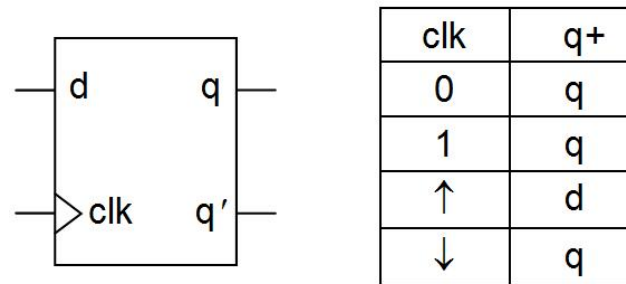


Figura 1.8

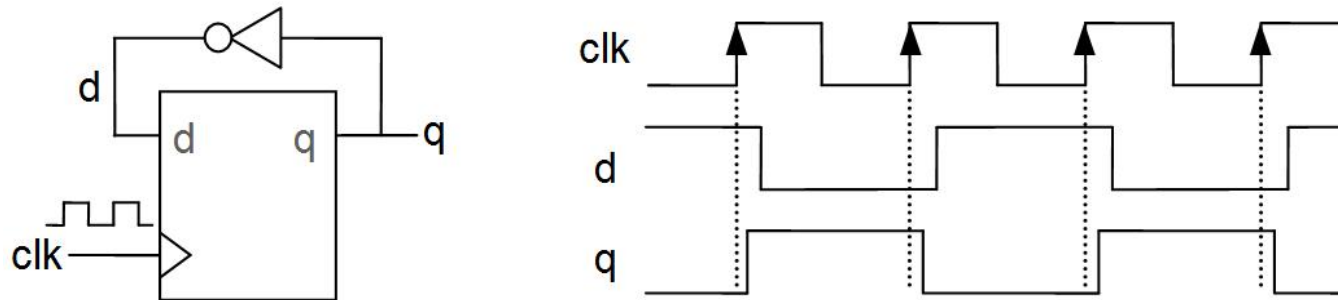


Figura 1.9

Figuras do Capítulo 1

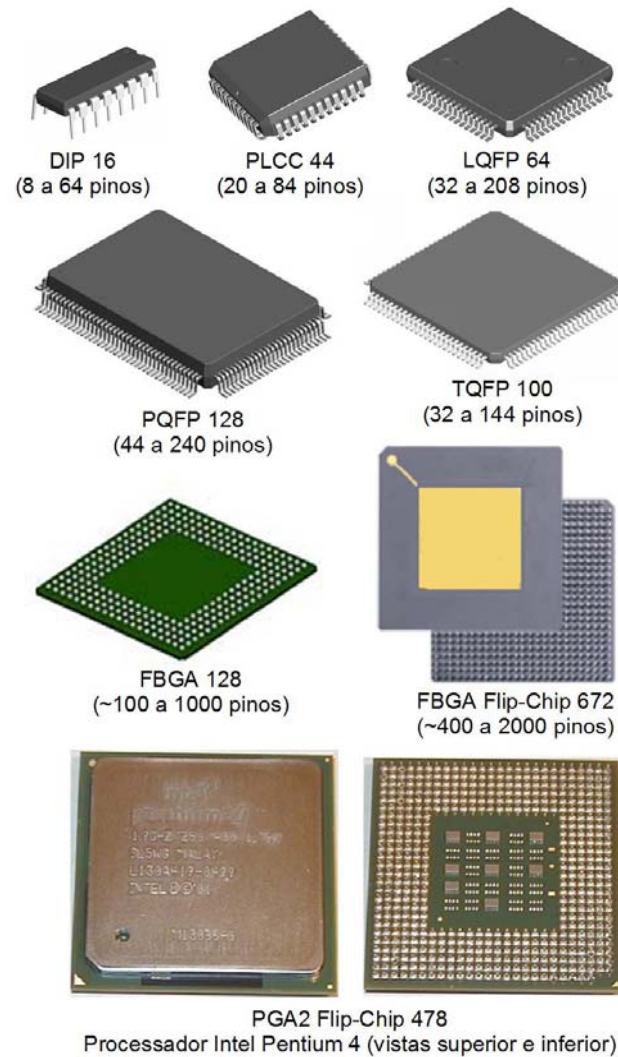
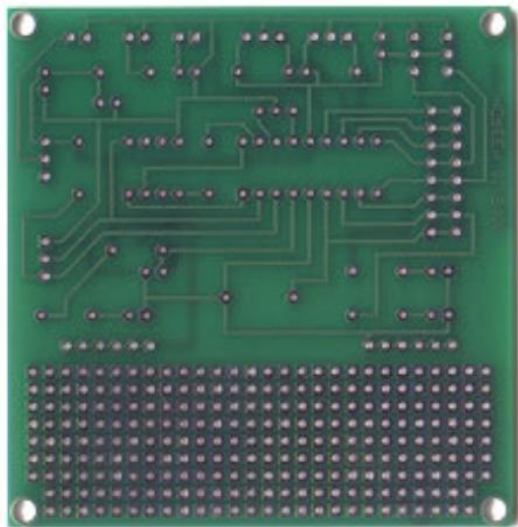
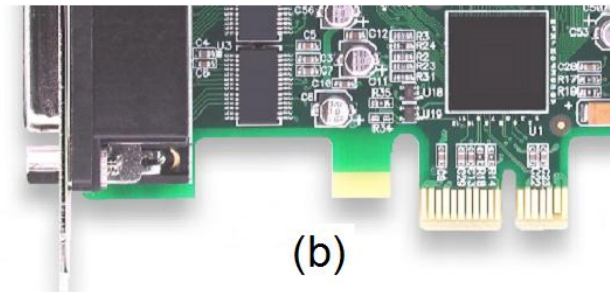


Figura 1.10

Figuras do Capítulo 1



(a)



(b)

Figura 1.11

Figuras do Capítulo 1

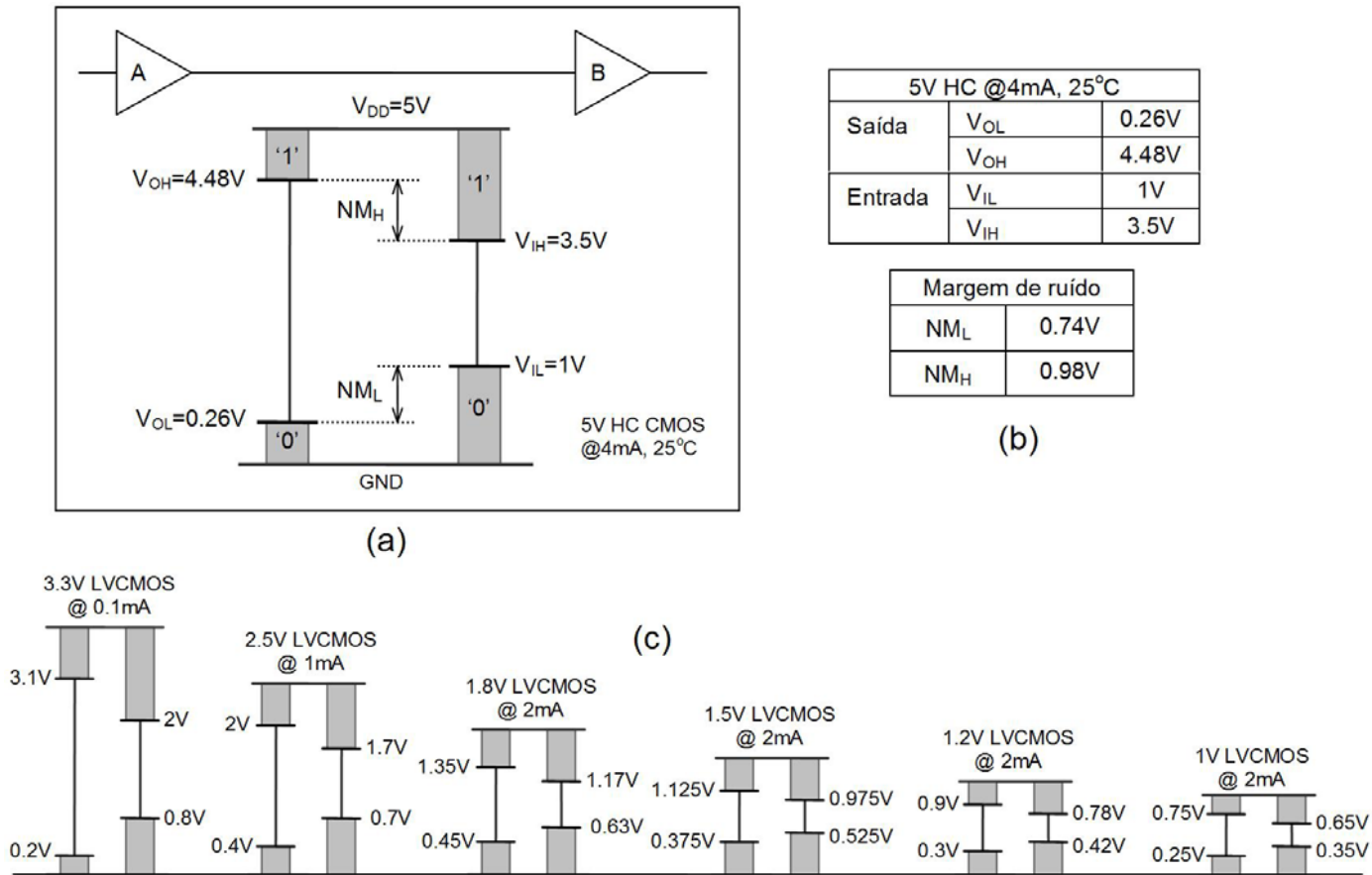


Figura 1.12

Figuras do Capítulo 1

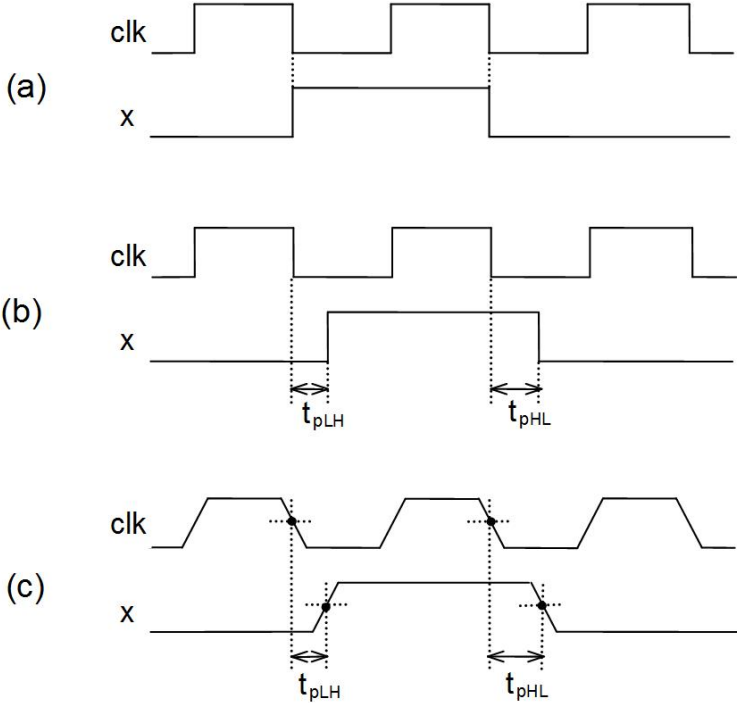


Figura 1.13

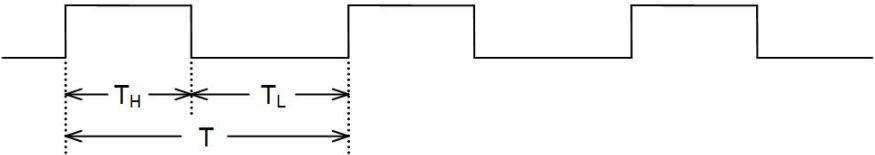


Figura 1.14

Figuras do Capítulo 1

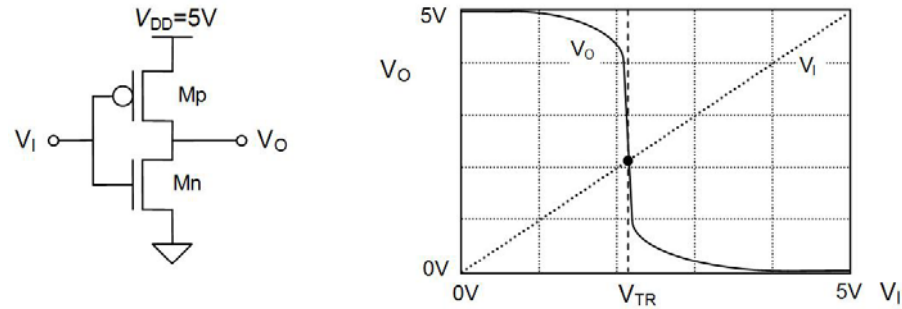


Figura 1.15

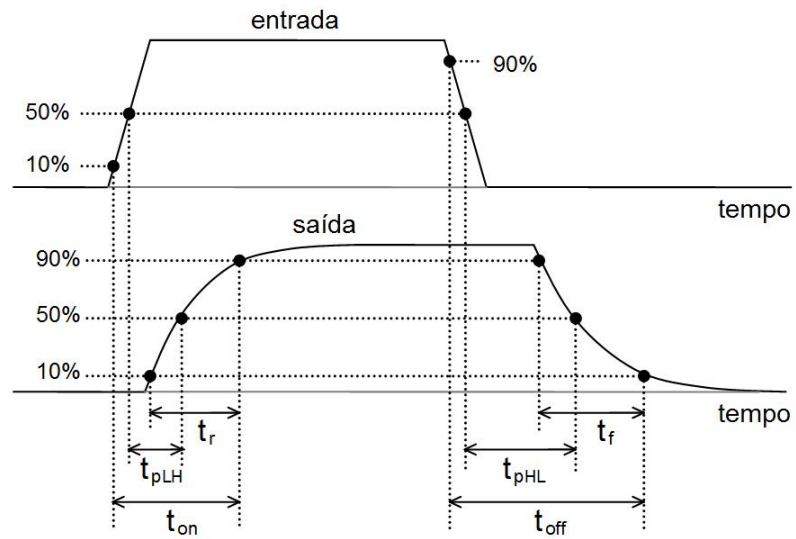


Figura 1.16

Figuras do Capítulo 1

Característica	Xilinx Virtex 5 (série LX)	Altera Stratix III (série L)
Tecnologia	CMOS 65nm (SRAM)	CMOS 65nm (SRAM)
Voltagem do core	1V	0,9V a 1,1V
Número de CLBs (Virtex) Número de LABs (Stratix)	2.400 a 25.920 -----	----- 1.900 a 13.520
Número de Slices (Virtex) Número de ALMs (Stratix)	4.800 a 51.840 -----	----- 19.000 a 135.200
Número de flip-flops	19.200 a 207.360	38.000 a 270.400
Frequência máx. do clock	550MHz	600MHz
Memória SRAM disponível (bits)	1,47M a 13,8M	2,4M a 20,4M
Número de blocos para DSP	32 a 192	27 a 96
Número de PLLs	2 a 6	4 a 12
Número de pinos de I/O dispon.	400 a 1.200	288 a 1.104

Figura 1.17

Figuras do Capítulo 1

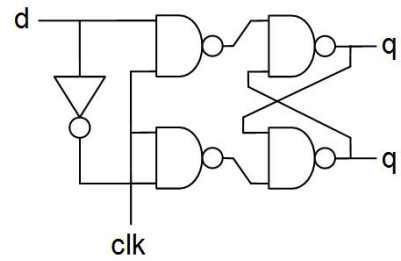


Figura 1.18

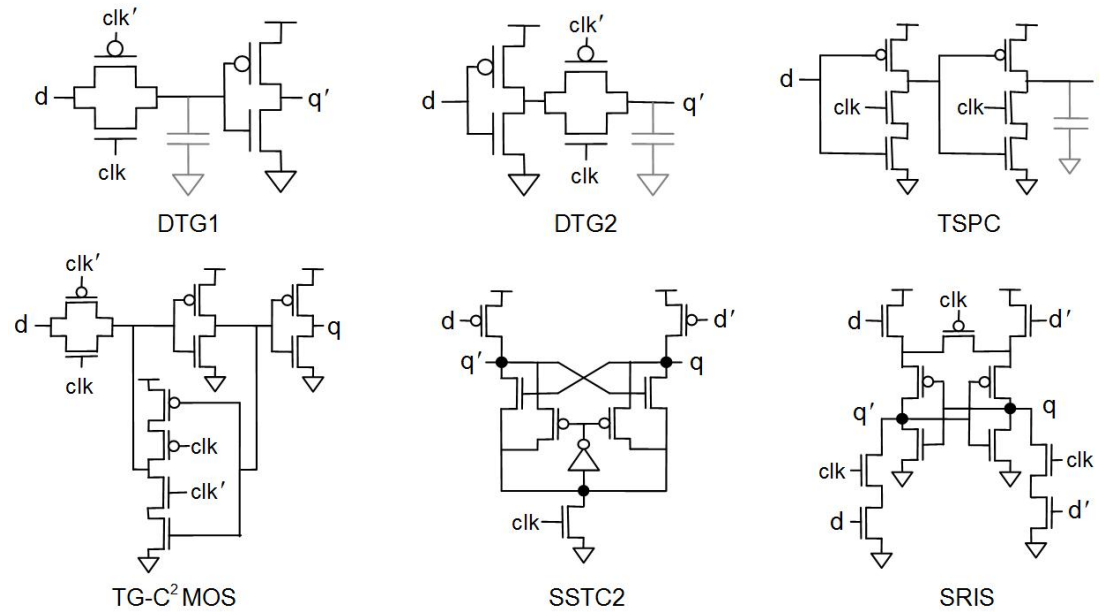


Figura 1.19

Figuras do Capítulo 2

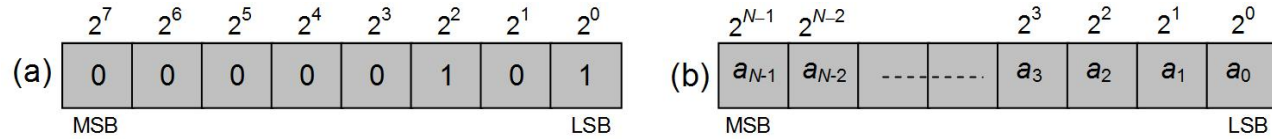


Figura 2.1

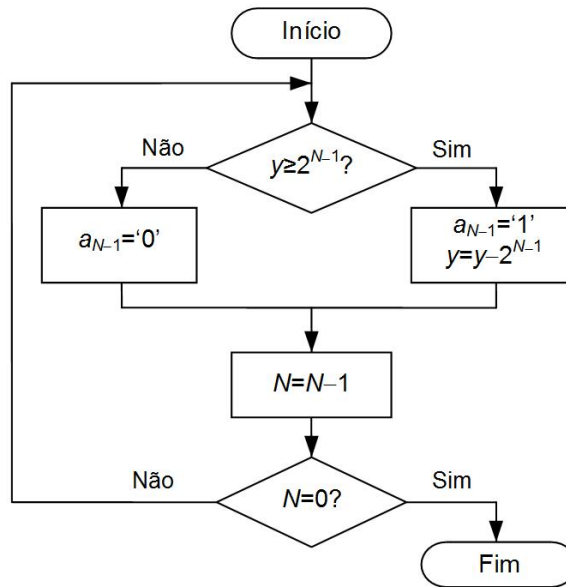


Figura 2.2

Figuras do Capítulo 2

Número decimal	Código binário conv.	Código octal	Código hexadec.	Código Gray	Código one-hot	Código Johnson	Código BCD
0	0000	00	0	0000	0000000000000001	00000000	0000
1	0001	01	1	0001	0000000000000010	10000000	0001
2	0010	02	2	0011	0000000000000100	11000000	0010
3	0011	03	3	0010	0000000000001000	11100000	0011
4	0100	04	4	0110	0000000000100000	11110000	0100
5	0101	05	5	0111	0000000001000000	11111000	0101
6	0110	06	6	0101	0000000010000000	11111100	0110
7	0111	07	7	0100	0000000010000000	11111110	0111
8	1000	10	8	1100	0000000100000000	11111111	1000
9	1001	11	9	1101	0000001000000000	01111111	1001
10	1010	12	A	1111	0000010000000000	00111111	0001 0000
11	1011	13	B	1110	0000100000000000	00011111	0001 0001
12	1100	14	C	1010	0001000000000000	00001111	0001 0010
13	1101	15	D	1011	0010000000000000	00000111	0001 0011
14	1110	16	E	1001	0100000000000000	00000011	0001 0100
15	1111	17	F	1000	1000000000000000	00000001	0001 0101

Figura 2.3

Figuras do Capítulo 2

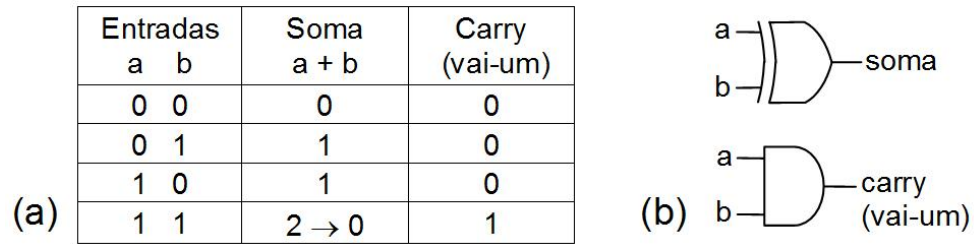


Figura 2.4



Figura 2.5

Figuras do Capítulo 2

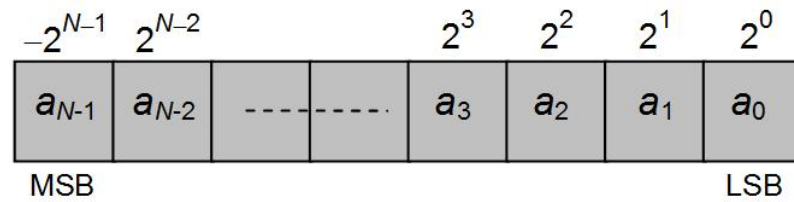


Figura 2.6

Palavra binária	Decimal sem sinal	Decimal com sinal
000	0	0
001	1	1
010	2	2
011	3	3
100	4	-4
101	5	-3
110	6	-2
111	7	-1

Figura 2.7

Figuras do Capítulo 2

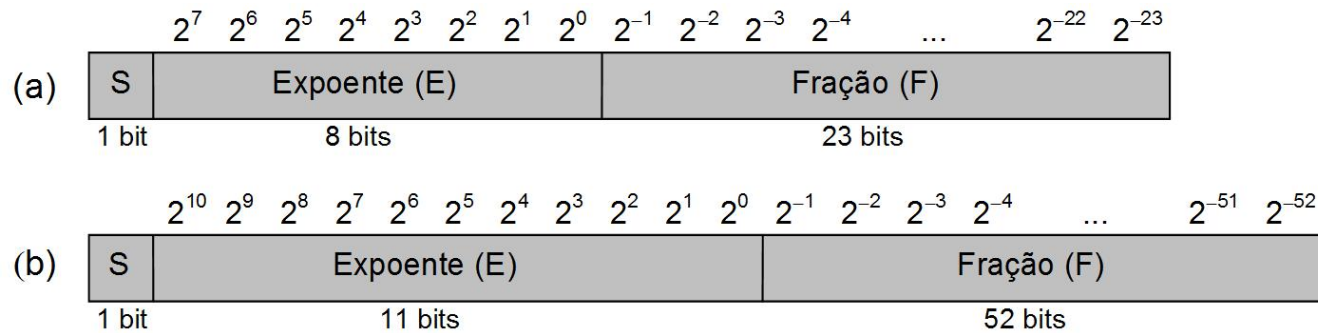


Figura 2.8

Sinal (S)	Expoente (E)	Fração (F)	Valor (y)
0 / 1	0	0	+0 / -0
0 / 1	max	0	$+\infty$ / $-\infty$
0 / 1	max	$\neq 0$	NaN
0 / 1	0	$\neq 0$	não normalizado
0 / 1	1 até max-1	qualquer	normalizado

max=255 para precisão simples ou 2047 para precisão dupla
 NaN= Not a number (não é um número)

Figura 2.9

Figuras do Capítulo 2

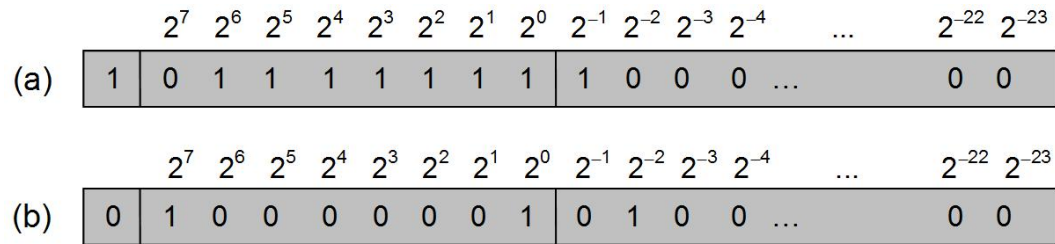
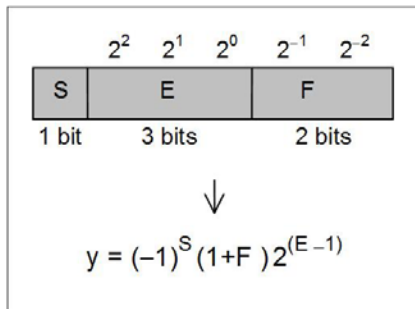


Figura 2.10



(a)

(b)

Exp. (E)	Fração (F)	Valor (y)
000	00	± 0.5
	01	± 0.625
	10	± 0.75
	11	± 0.875
001	00	± 1
	01	± 1.25
	10	± 1.5
	11	± 1.75
010	00	± 2
	01	± 2.5
	10	± 3
	11	± 3.5
011	00	± 4
	01	± 5
	10	± 6
	11	± 7
100	00	± 8
	01	± 10
	10	± 12
	11	± 14
101	00	± 16
	01	± 20
	10	± 24
	11	± 28
110	00	± 32
	01	± 40
	10	± 48
	11	± 56
111	00	± 64
	01	± 80
	10	± 96
	11	± 112

Figura 2.11

Figuras do Capítulo 2

b ₃ b ₂ b ₁ b ₀	b ₆ b ₅ b ₄							
	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Figura 2.12

Figuras do Capítulo 2

Cód. de controle e cód. latim básico (parcial)								Cód. para números gregos antigos					
	000	001	002	003	004	005	006	007	1014	1015	1016	1017	1018
0	NUL: 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070	⸂ 10140	Δ 10150	Σ 10160	⸂ 10170	ƒ 10180
1	SOH: 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071	⸂ 10141	⸂ 10151	⸂ 10161	Υ 10171	⸂ 10181
2	STX: 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072	⸂ 10142	⸂ 10152	⸂ 10162	⸂ 10172	⸂ 10182
3	ETX: 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073	⸂ 10143	⸂ 10153	⸂ 10163	⸂ 10173	⸂ 10183
4	EOT: 0004	DC4 0014	\$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074	⸂ 10144	⸂ 10154	⸂ 10164	⸂ 10174	⸂ 10184

Figura 2.13

Figuras do Capítulo 2

Código Unicode UTF-8					
Faixa decimal	Ponto Unicode	Byte 1	Byte 2	Byte 3	Byte 4
0 a 127	0000 0000 0aaa aaaa	0aaa aaaa			
128 a 2047	0000 0bbb bbaa aaaa	110b bbbb	10aa aaaa		
2.047 a 55.295 e 57.344 a 65.535	cccc bbbb bbaa aaaa	1110 cccc	10bb bbbb	10aa aaaa	
65.536 a 1M (Nota 2)	000d dddd cccc bbbb bbaa aaaa	1111 0ddd	10dd cccc	10bb bbbb	10aa aaaa

Código Unicode UTF-16					
Faixa decimal	Ponto Unicode	Byte 1	Byte 2	Byte 3	Byte 4
0 a 55.295 e 57.344 a 65.535	aaaa aaaa aaaa aaaa	aaaa aaaa	aaaa aaaa		
65.536 a 1M	000b bbbb aaaa aaaa aaaa aaaa	1101 10cc (Nota 3)	ccaa aaaa	1101 11aa	aaaa aaaa

Código Unicode UTF-32					
Faixa decimal	Ponto Unicode	Byte 1	Byte 2	Byte 3	Byte 4
0 a 55.295 e 57.344 a 1M	000a aaaa aaaa aaaa aaaa aaaa	0000 0000	000a aaaa	aaaa aaaa	aaaa aaaa

<p>Nota 1: a, b, c, e d são bits simples. Nota 2: 1M=1.114.111. Nota 3: c=b-1 truncado à esquerda com 4 bits.</p>

Figura 2.14

Figuras do Capítulo 3

$$\begin{array}{rcccc}
 & c_4 & c_3 & c_2 & c_1 & \text{(carry)} \\
 + & & a_3 & a_2 & a_1 & a_0 \\
 & & b_3 & b_2 & b_1 & b_0 \\
 \hline
 s_4 & s_3 & s_2 & s_1 & s_0 & \text{(soma)}
 \end{array}$$

(a)

Carry na entrada (cin)	Entradas a b	Soma cin + a + b	Carry na saída (cout)
0	0 0	0	0
0	0 1	1	0
0	1 0	1	0
0	1 1	2 → 0	1
1	0 0	1	0
1	0 1	2 → 0	1
1	1 0	2 → 0	1
1	1 1	3 → 1	1

(b)

$$\begin{array}{rcccc}
 & 1 & 1 & 0 & 0 \\
 + & & 1 & 1 & 0 & 0 \\
 & & 0 & 1 & 1 & 0 \\
 \hline
 1 & 0 & 0 & 1 & 0
 \end{array}$$

(c)

Figura 3.1

Figuras do Capítulo 3

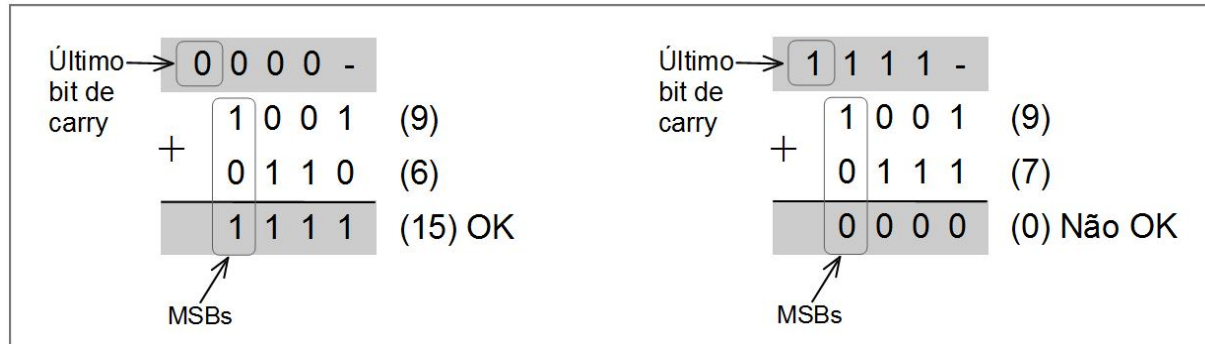


Figura 3.2

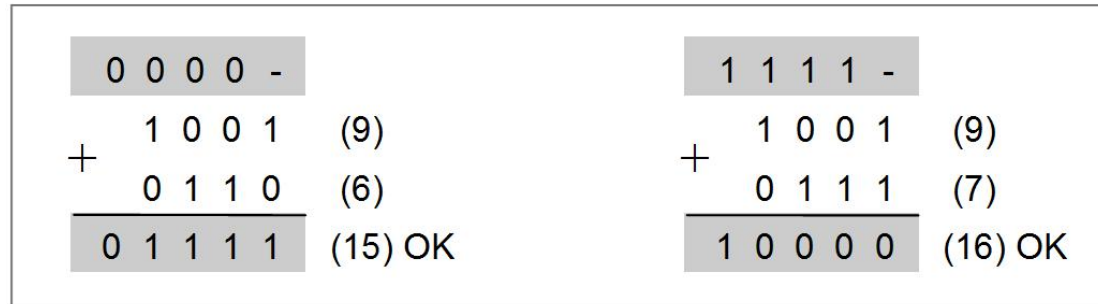


Figura 3.3

Figuras do Capítulo 3

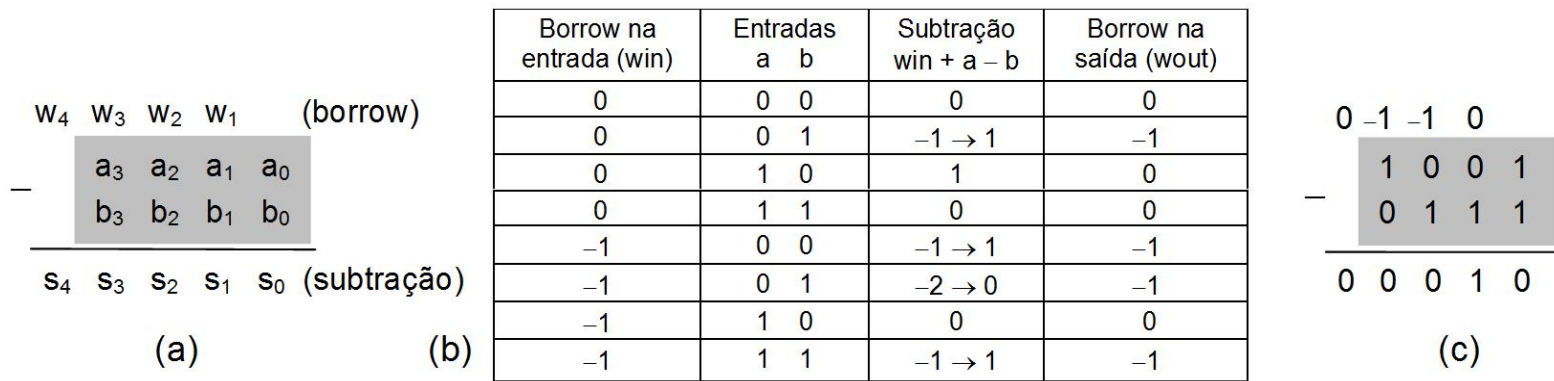


Figura 3.4

Figuras do Capítulo 3

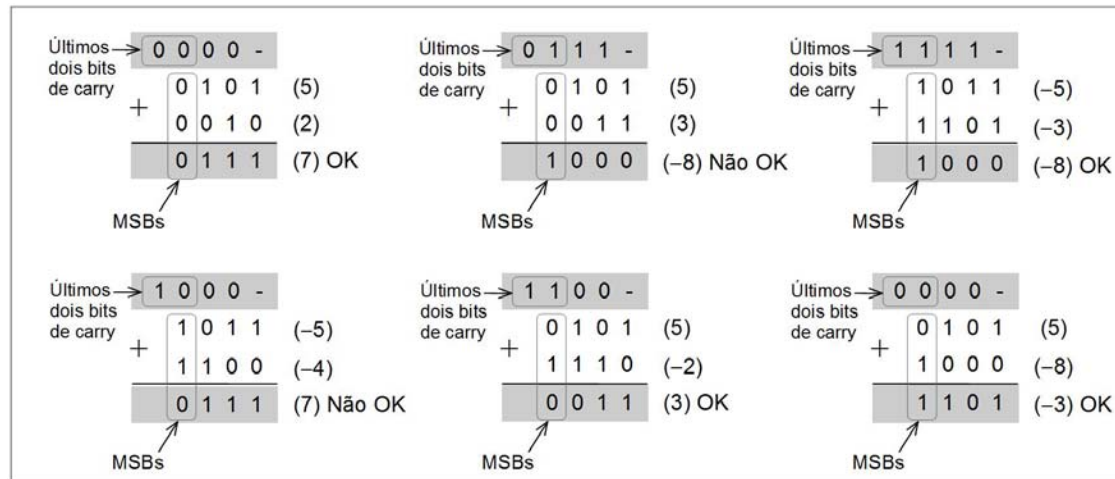


Figura 3.5

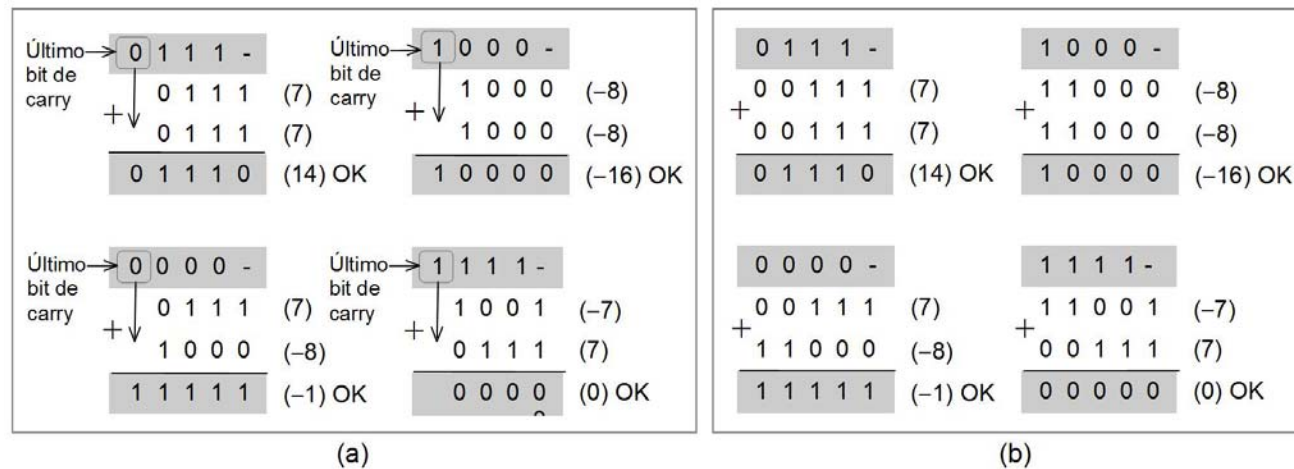


Figura 3.6

Figuras do Capítulo 3

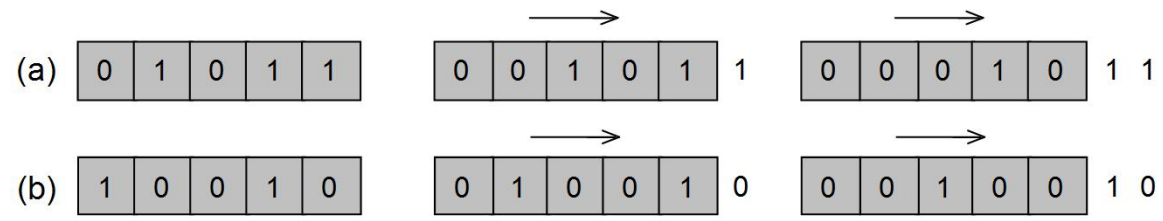


Figura 3.7

Figuras do Capítulo 3

Iteração	Procedimento	Multiplicando	Produto		
			Esquerda	Direito	Extra
0	Inicialização (ProdDireita recebe Multiplicador)	10010	00000	0111 0	0
1	Bits=00 → Sem operação Deslocamento aritmético p/ a direita		00000	0111 0	0
			00000	0011 1	0
2	Bits=10 → ProdEsquerda – Multiplicando Deslocamento aritmético p/ a direita		01110	0011 1	0
			00111	0001 1	1
3	Bits=11 → Sem operação Deslocamento aritmético p/ a direita		00111	0001 1	1
			00011	1000 1	1
4	Bits=11 → Sem operação Deslocamento aritmético p/ a direita		00011	1000 1	1
			00001	1100 0	1
5	Bits=01 → ProdEsquerda + Multiplicando Deslocamento aritmético p/ a direita		10011	11000	
			11001	11100	

Figura 3.12

Figuras do Capítulo 3

Dividendo →	0 0 0 1 1 0 1		0 1 0 1	← Divisor (=5)
(=13)	0 1 0 1		0 0 1 0	← Quociente (=2)
	0 0 0 1 1			← Resto (=3)

Figura 3.13

Iteração	Procedimento	Divisor	Resto Esquerda Direita
0	Inicialização (RestoDireita recebe Dividendo) Deslocamento p/ a esquerda com entrada de '0'	0101	0000 1101 0001 1010
1	RestoEsquerda – Divisor Bit=1 → RestoEsquerda + Divisor Bit=1 → Desloc. p/ a esquerda com entrada de '0'		1 100 1010 0001 1010 0011 0100
2	RestoEsquerda – Divisor Bit=1 → RestoEsquerda + Divisor Bit=1 → Desloc. p/ a esquerda com entrada de '0'		1 110 0100 0011 0100 0110 1000
3	RestoEsquerda – Divisor Bit=0 → Sem operação Bit=0 → Desloc. p/ a esquerda com entrada de '1'		0 001 1000 0001 1000 0011 0001
4	RestoEsquerda – Divisor Bit=1 → RestoEsquerda + Divisor Bit=1 → Desloc. p/ a esquerda com entrada de '0'		1 110 0001 0011 0001 0110 0010
	Deslocamento RestoEsquerda p/ direita com '0'		0011 0010

Figura 3.14

Figuras do Capítulo 3

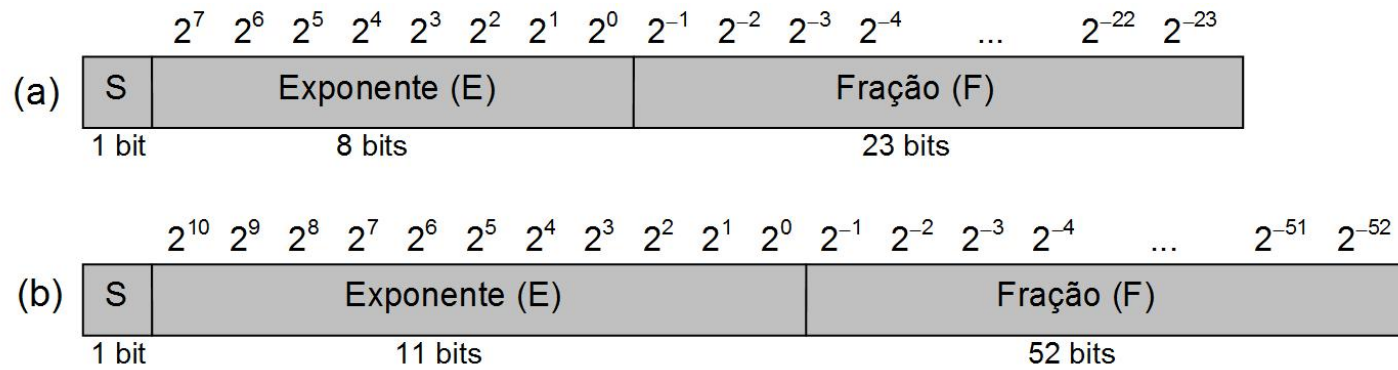


Figura 3.15