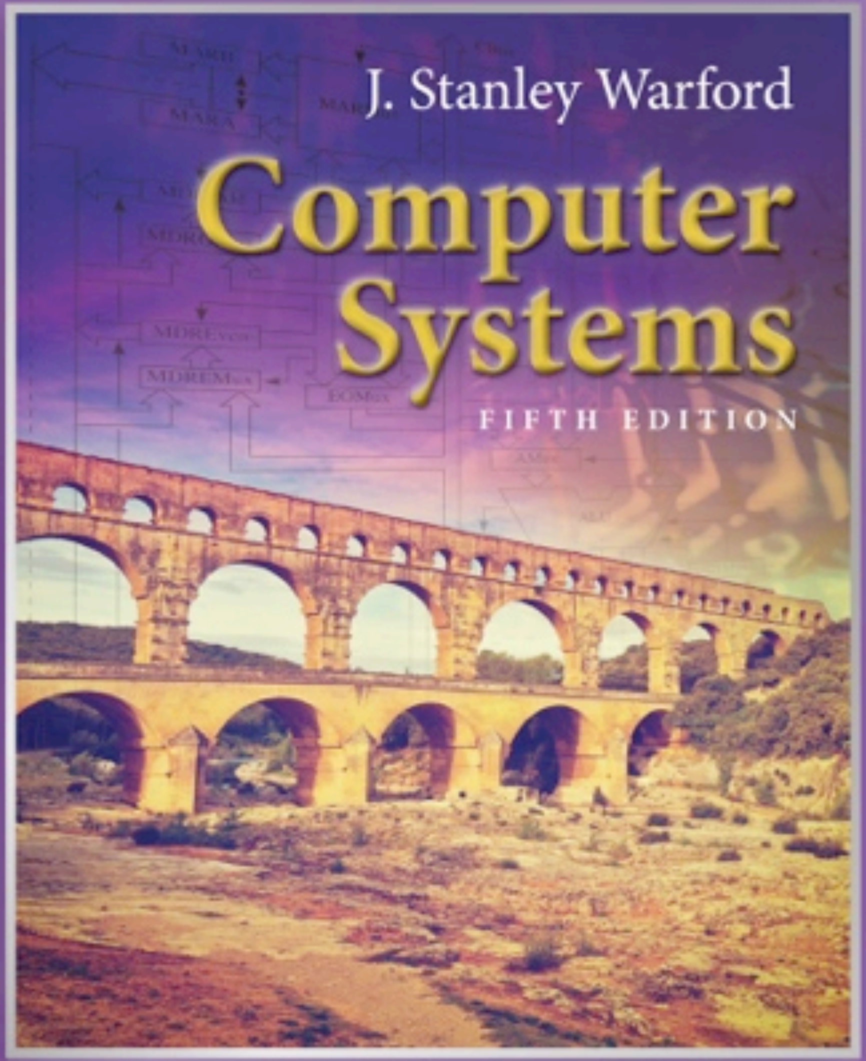
Chapter 3

Information Representation



Instruction Set Architecture

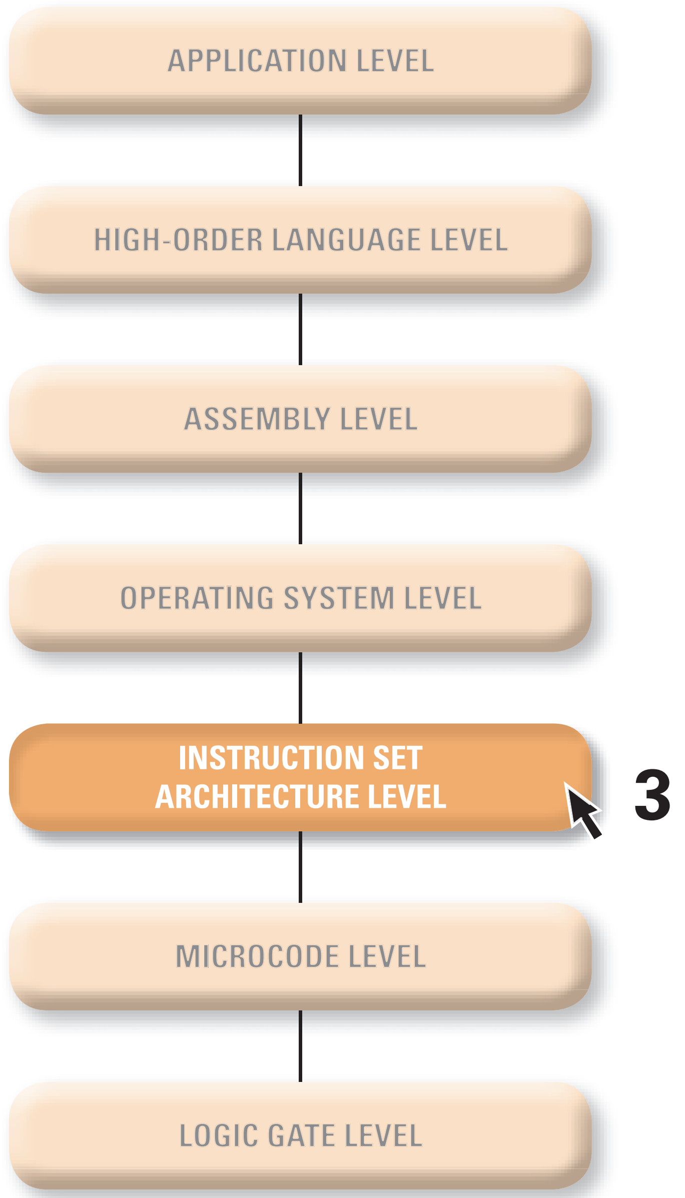
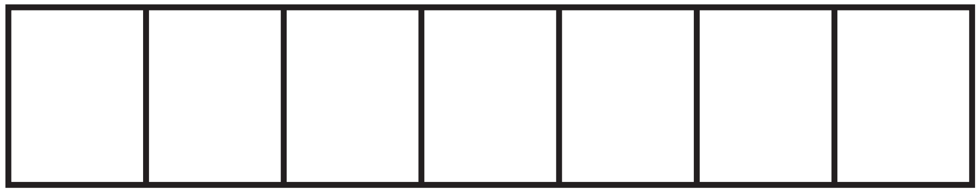
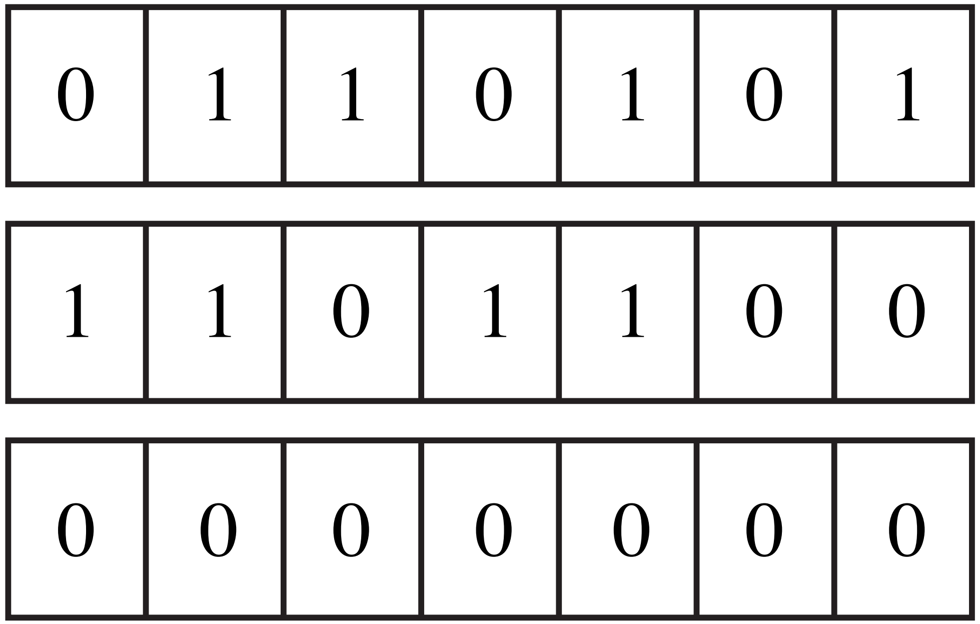


Figure 3.1



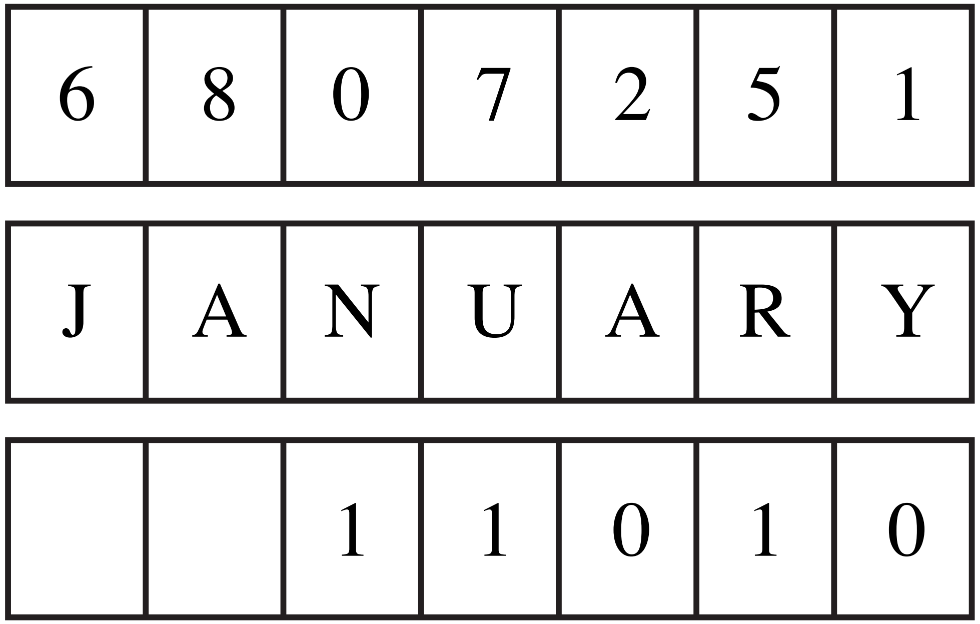
(a) A seven-bit cell.

Figure 3.1 (continued)



(b) Some possible values in a seven-bit cell.

Figure 3.1 (continued)



(c) Some impossible values in a seven-bit cell.

Counting in decimal

0	7	14	21	28	35
1	8	15	22	29	36
2	9	16	23	30	37
3	10	17	24	31	38
4	11	18	25	32	.
5	12	19	26	33	.
6	13	20	27	34	.

Counting in octal

0	7	16	25	34	43
1	10	17	26	35	44
2	11	20	27	36	45
3	12	21	30	37	46
4	13	22	31	40	.
5	14	23	32	41	.
6	15	24	33	42	.

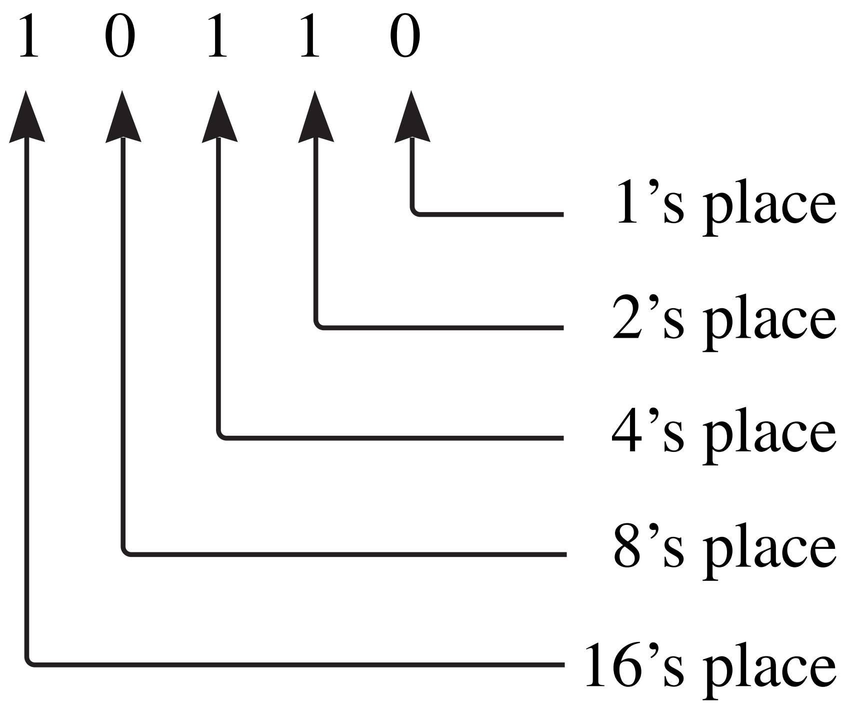
Counting in base 3

0	21	112	210	1001	1022
1	22	120	211	1002	1100
2	100	121	212	1010	1101
10	101	122	220	1011	1102
11	102	200	221	1012	.
12	110	201	222	1020	.
20	111	202	1000	1021	.

Counting in binary

0	111	1110	10101	11100	100011
1	1000	1111	10110	11101	100100
10	1001	10000	10111	11110	100101
11	1010	10001	11000	11111	100110
100	1011	10010	11001	100000	.
101	1100	10011	11010	100001	.
110	1101	10100	11011	100010	.

Figure 3.2



(a) The place values for 10110 (bin).

0	1's place	=	0
1	2's place	=	2
1	4's place	=	4
0	8's place	=	0
1	16's place	=	16
			22 (dec)

(b) Converting 10110 (bin) to decimal.

Figure 3.3

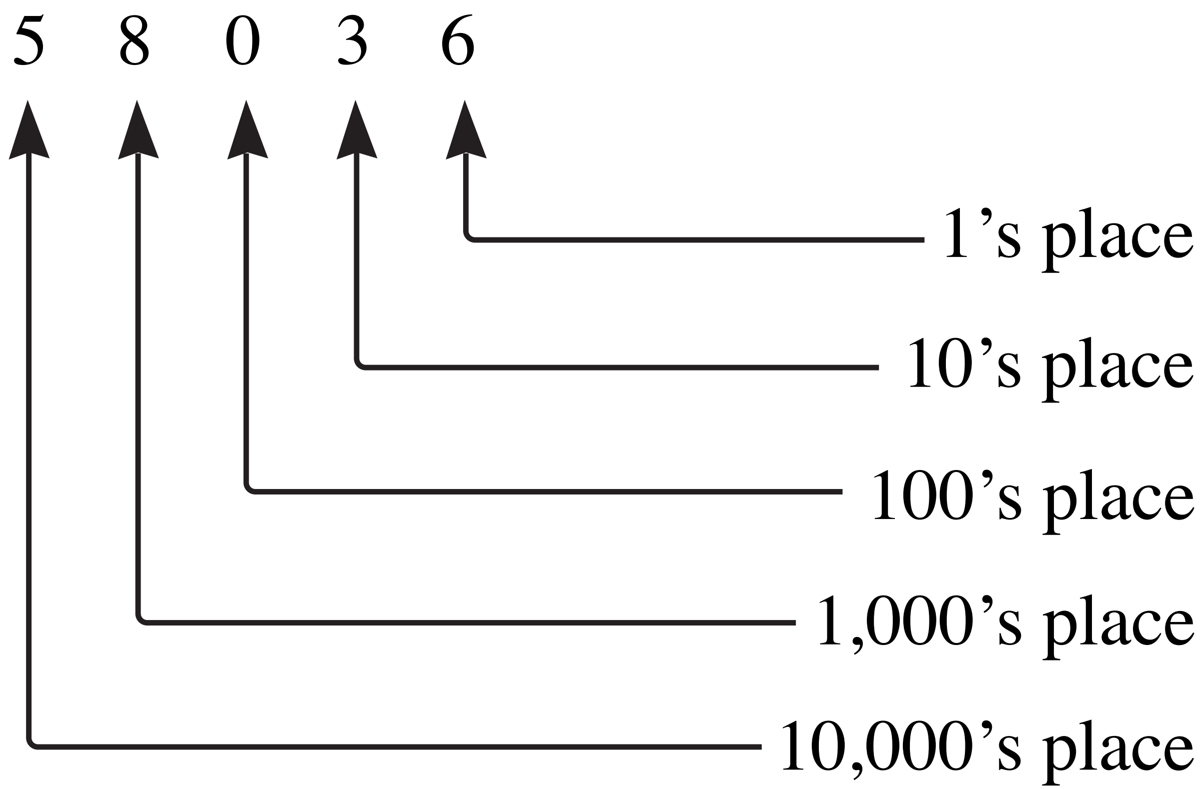

5 8 0 3 6
1's place
10's place
100's place
1,000's place
10,000's place

Figure 3.4

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

(a) The binary number 10110.

$$5 \times 10^4 + 8 \times 10^3 + 0 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$$

(b) The decimal number 58,036.

Figure 3.5

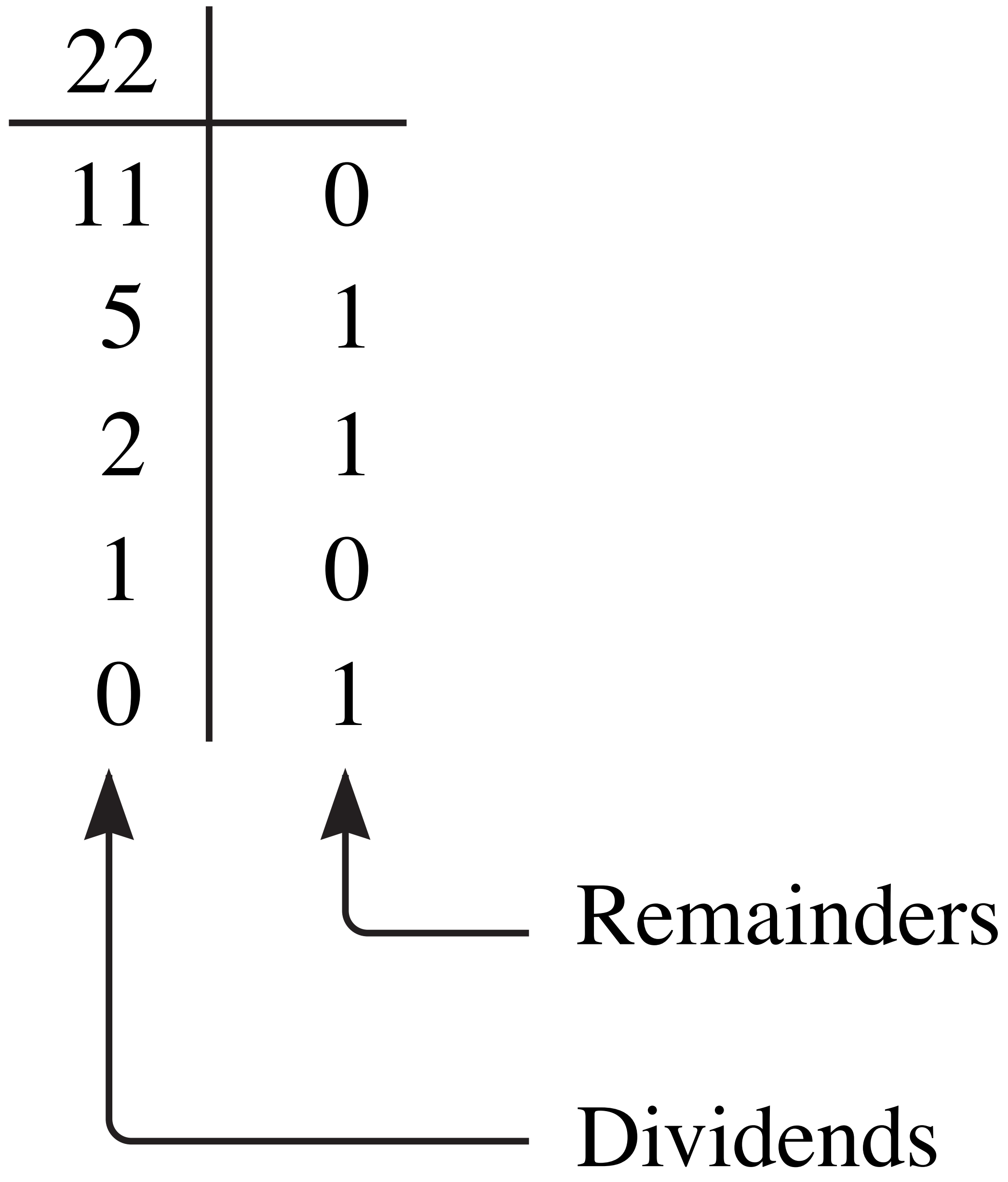
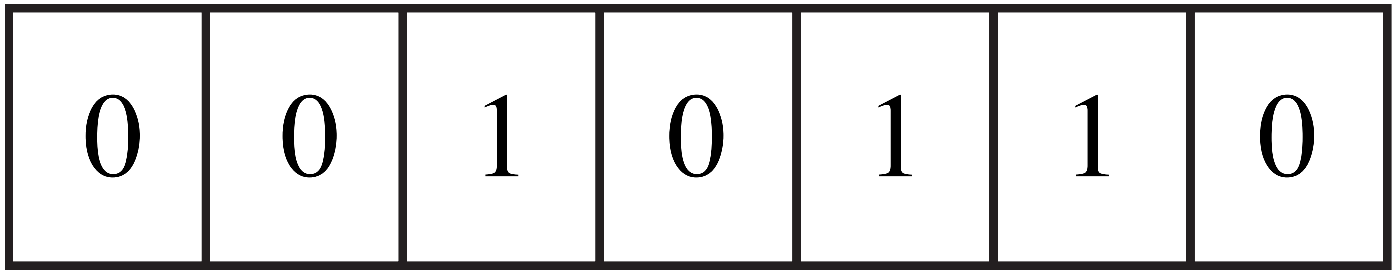


Figure 3.6



Binary addition rules

$0 + 0 = 0$

$0 + 1 = 1$

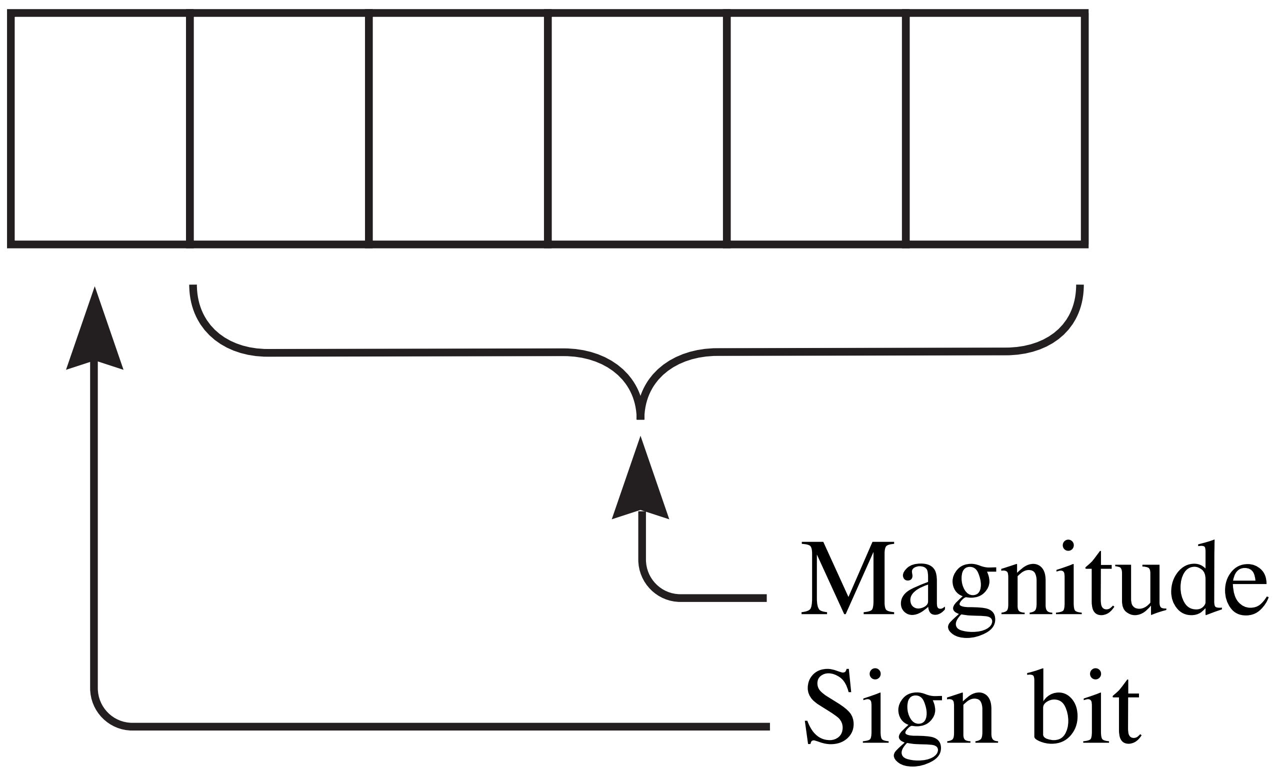
$1 + 0 = 1$

$1 + 1 = 10$

Figure 3.7

−	0	0	1	0	1

Figure 3.8



- The NEG operation
  - Taking the two’s complement
- The NOT operation
  - Change the 1’s to 0’s and the 0’s to 1’s

The two's complement rule

- The two's complement of a number is 1 plus its one's complement
- NEG x = 1 + NOT x

Figure 3.9

Decimal	Binary
−7	1001
−6	1010
−5	1011
−4	1100
−3	1101
−2	1110
−1	1111

Figure 3.10

Decimal	Binary		
−8	1000	0	0000
−7	1001	1	0001
−6	1010	2	0010
−5	1011	3	0011
−4	1100	4	0100
−3	1101	5	0101
−2	1110	6	0110
−1	1111	7	0111

Figure 3.11

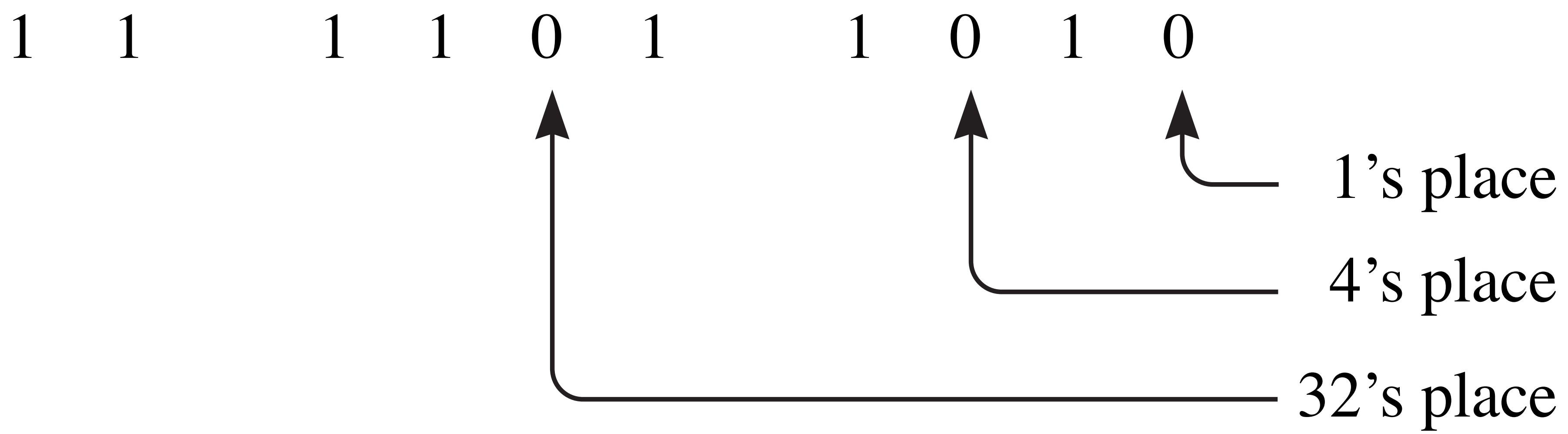


Figure 3.12

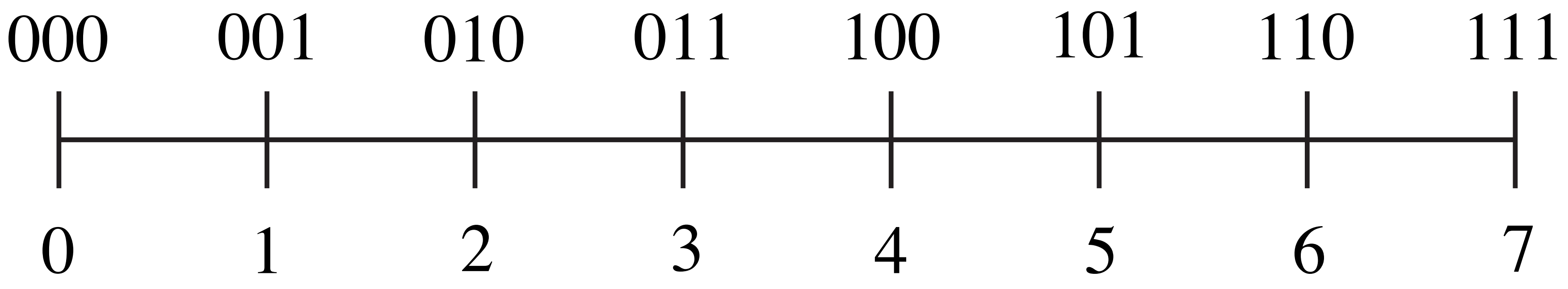
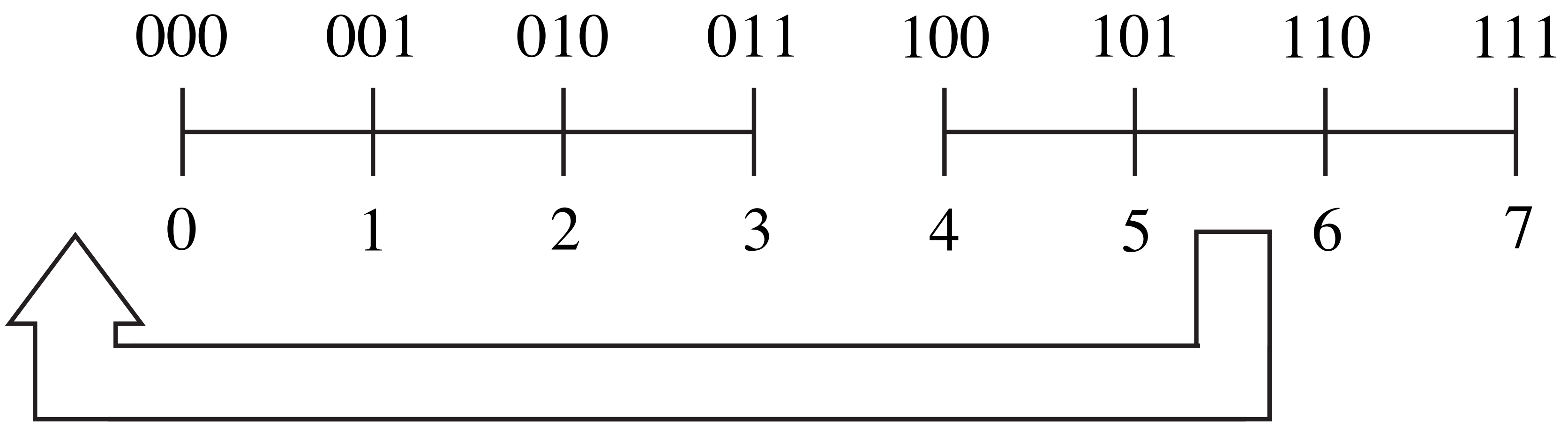
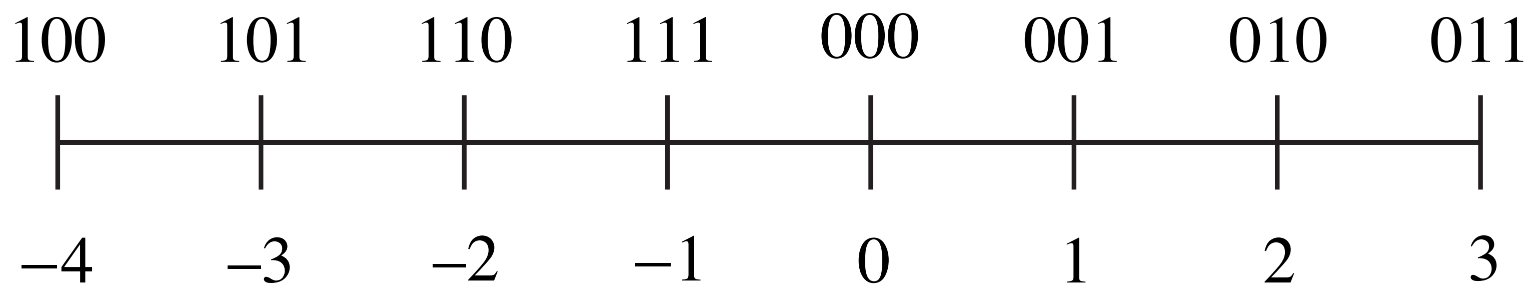


Figure 3.13



(a) Breaking the number line in the middle.



(b) Shifting the right part to the left side.

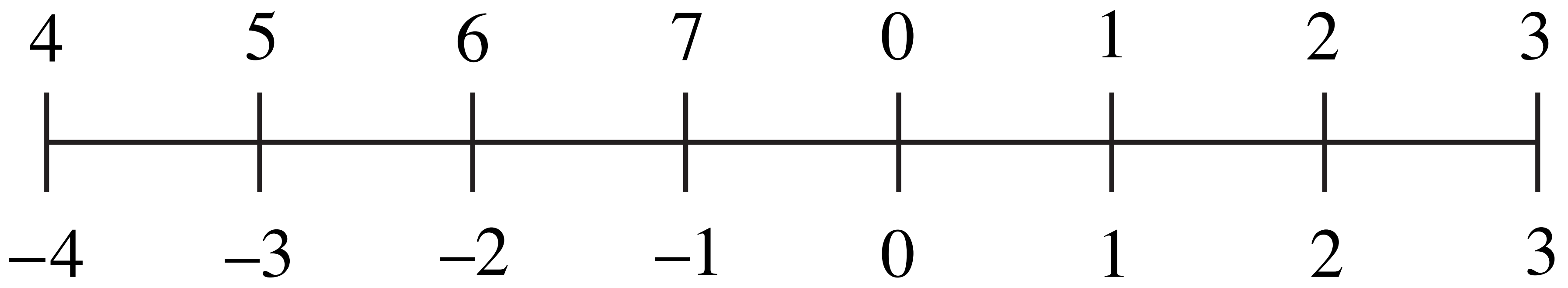

4
5
6
7
0
1
2
3
–4
–3
–2
–1
0
1
2
3

Figure 3.15

```
#include <stdio.h>
#include <limits.h>

int main() {
   int n = INT_MAX - 2;
   for (int i = 0; i < 6; i++) {
      printf("n == %d\n", n);
      n++;
   }
   return 0;
}
```

Output

```
n == 2147483645
n == 2147483646
n == 2147483647
n == -2147483648
n == -2147483647
n == -2147483646
```

The status bits

- N = 1 if the result is negative

  N = 0 otherwise
- Z = 1 if the result is all zeros

  Z = 0 otherwise
- V = 1 if a signed integer overflow occurred

  V = 0 otherwise
- C = 1 if an unsigned integer overflow occurred

  C = 0 otherwise

Addition with a 6-bit cell

Adding two positives:		00 0011		01 0110
	ADD	01 0101	ADD	00 1100
	V = 0	01 1000	V = 1	10 0010
	C = 0		C = 0	
Adding a positive and a negative:		00 0101		00 1000
	ADD	11 0111	ADD	11 1010
	V = 0	11 1100	V = 0	00 0010
	C = 0		C = 1	
Adding two negatives:		11 1010		10 0110
	ADD	11 0111	ADD	10 0010
	V = 0	11 0001	V = 1	00 1000
	C = 1		C = 1	

Computer Systems FIFTH EDITION

Figure 3.16

p	*q*	*p* AND *q*
0	0	0
0	1	0
1	0	0
1	1	1

(a) ISA3 table for AND.

p	*q*	*p* OR *q*
0	0	0
0	1	1
1	0	1
1	1	1

(b) ISA3 table for OR.

p	*q*	*p* XOR *q*
0	0	0
0	1	1
1	0	1
1	1	0

(c) ISA3 table for XOR.

Figure 3.17

p	*q*	*p* AND *q*
true	true	true
true	false	false
false	true	false
false	false	false

(a) HOL6 table for AND.

p	*q*	*p* OR *q*
true	true	true
true	false	true
false	true	true
false	false	false

(b) HOL6 table for OR.

p	*q*	*p* XOR *q*
true	true	false
true	false	true
false	true	true
false	false	false

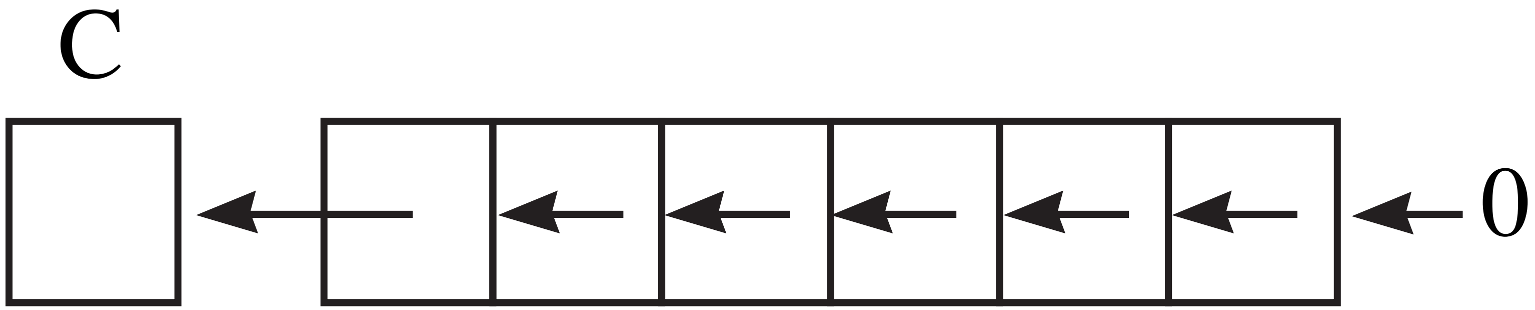
(c) HOL6 table for XOR.

Operation	RTL Symbol
AND	$\wedge$
OR	$\vee$
XOR	$\oplus$
NOT	$\neg$
Implies	$\Rightarrow$
Transfer	$\leftarrow$
Bit index	$\langle\ \rangle$
Informal description	$\{\ \}$
Sequential separator	;
Concurrent separator	,

RTL specification of OR operation

$$c \leftarrow a \vee b \,;\, \mathrm{N} \leftarrow c < 0 \,,\, \mathrm{Z} \leftarrow c = 0$$

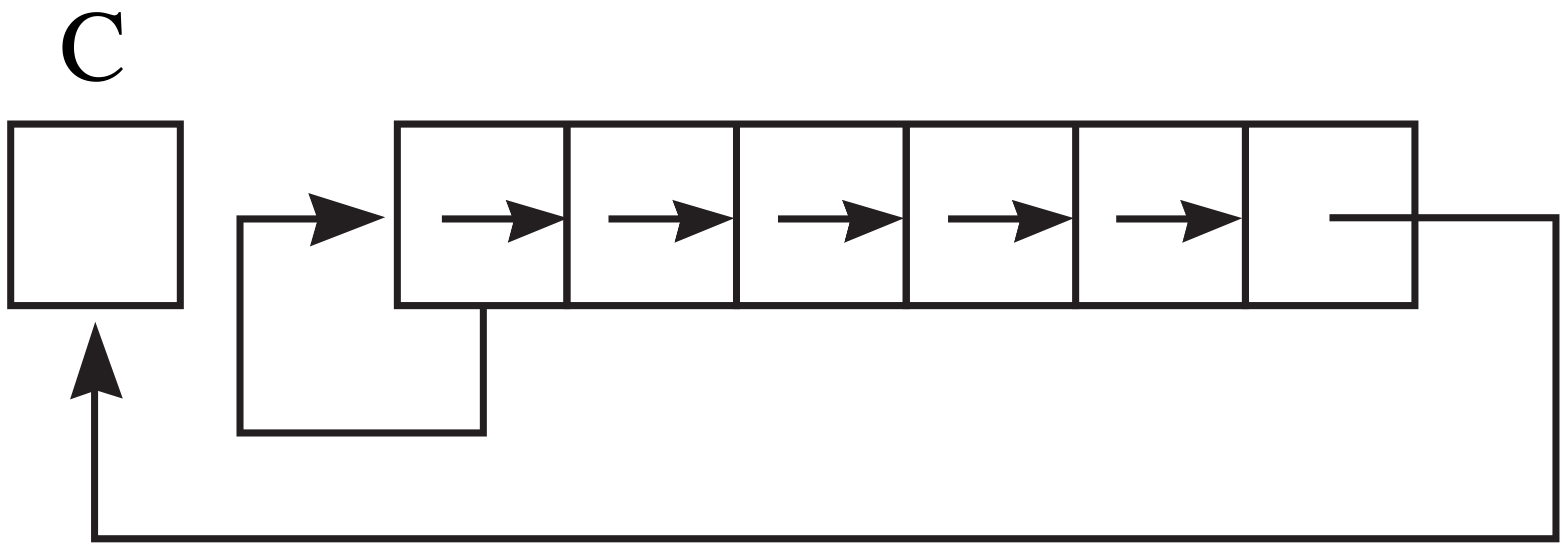
Arithmetic shift left (ASL)



$$C \leftarrow r\langle 0 \rangle\ ,\ r\langle 0..4 \rangle \leftarrow r\langle 1..5 \rangle\ ,\ r\langle 5 \rangle \leftarrow 0\ ;$$

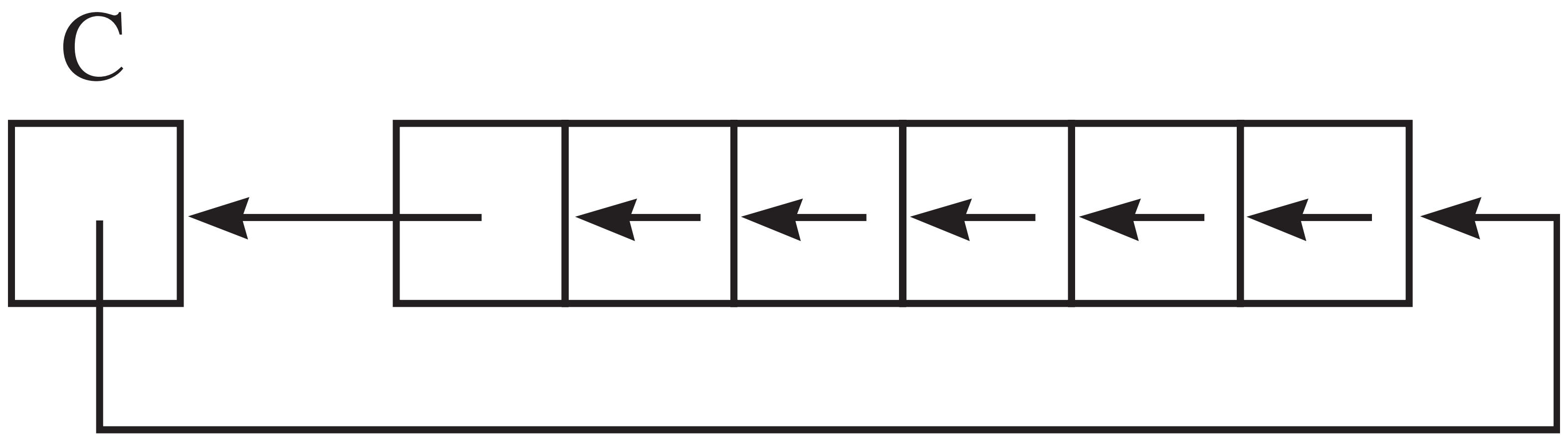
$$N \leftarrow r < 0\ ,\ Z \leftarrow r = 0\ ,\ V \leftarrow \{overflow\}$$

Arithmetic shift right (ASR)

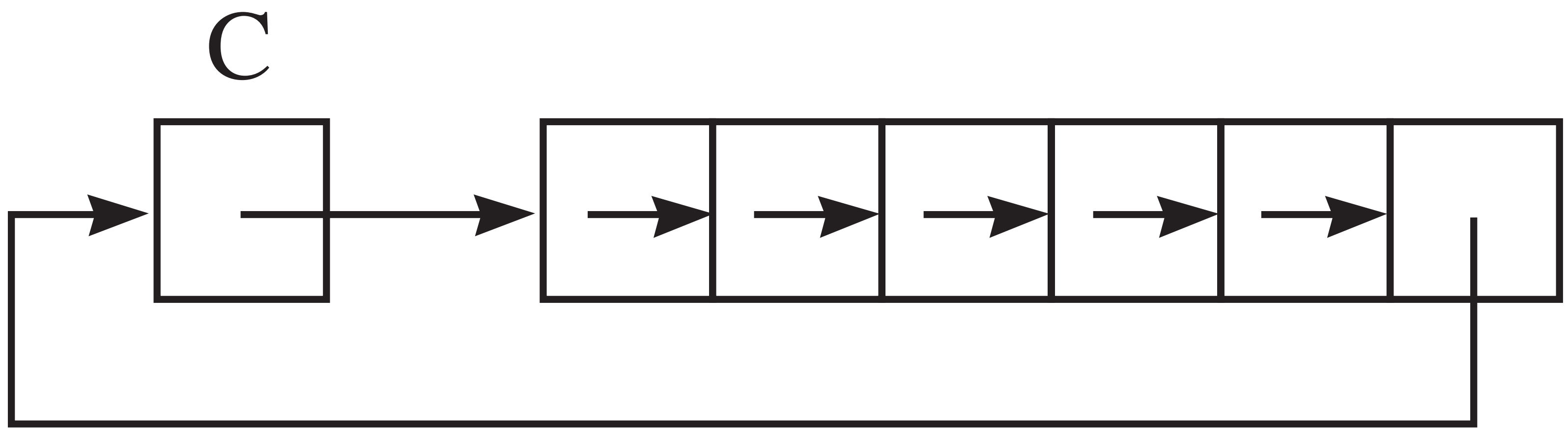


RTL specification is a problem for the student

Rotate left (ROL)



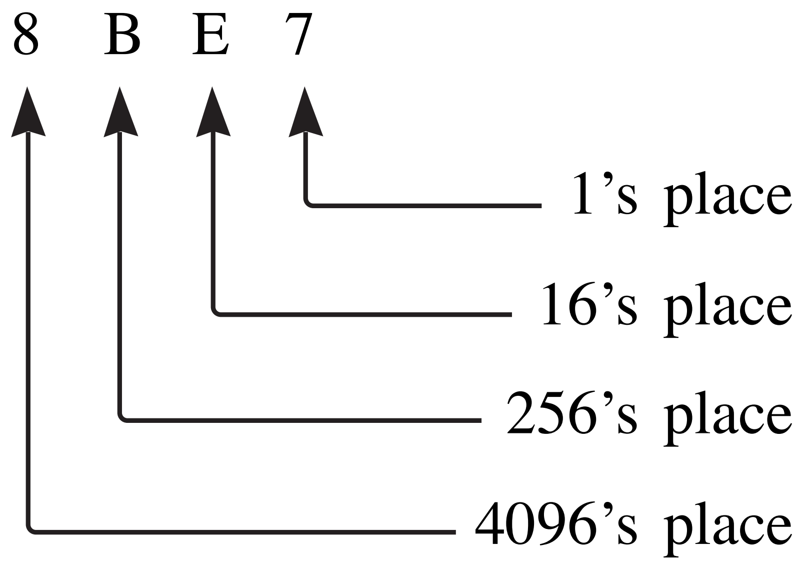
Rotate right (ROR)



Counting in hexadecimal

0	7	E	15	1C	23
1	8	F	16	1D	24
2	9	10	17	1E	25
3	A	11	18	1F	26
4	B	12	19	20	.
5	C	13	1A	21	.
6	D	14	1B	22	.

Figure 3.22



(a) The place values for 8BE7.

$$\begin{array}{rcrcr} 7 & \times & 1 & = & 7 \\ 14 & \times & 16 & = & 224 \\ 11 & \times & 256 & = & 2{,}816 \\ 8 & \times & 4096 & = & 32{,}768 \\ \hline & & & & 35{,}815 \end{array}$$

(b) Converting 8BE7 to decimal.

Figure 3.23

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0_	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1_	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2_	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3_	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4_	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5_	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6_	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7_	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8_	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9_	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A_	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B_	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C_	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D_	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E_	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F_	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Computer Systems FIFTH EDITION

Figure 3.24

Hexadecimal	Binary	Hexadecimal	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Char	Bin	Hex	Char	Bin	Hex	Char	Bin	Hex	Char	Bin	Hex
NUL	000 0000	00	SP	010 0000	20	@	100 0000	40	`	110 0000	60
SOH	000 0001	01	!	010 0001	21	A	100 0001	41	a	110 0001	61
STX	000 0010	02	"	010 0010	22	B	100 0010	42	b	110 0010	62
ETX	000 0011	03	#	010 0011	23	C	100 0011	43	c	110 0011	63
EOT	000 0100	04	$	010 0100	24	D	100 0100	44	d	110 0100	64
ENQ	000 0101	05	%	010 0101	25	E	100 0101	45	e	110 0101	65
ACK	000 0110	06	&	010 0110	26	F	100 0110	46	f	110 0110	66
BEL	000 0111	07	'	010 0111	27	G	100 0111	47	g	110 0111	67
BS	000 1000	08	(	010 1000	28	H	100 1000	48	h	110 1000	68
HT	000 1001	09	)	010 1001	29	I	100 1001	49	i	110 1001	69
LF	000 1010	0A	*	010 1010	2A	J	100 1010	4A	j	110 1010	6A
VT	000 1011	0B	+	010 1011	2B	K	100 1011	4B	k	110 1011	6B
FF	000 1100	0C	,	010 1100	2C	L	100 1100	4C	l	110 1100	6C
CR	000 1101	0D	-	010 1101	2D	M	100 1101	4D	m	110 1101	6D
SO	000 1110	0E	.	010 1110	2E	N	100 1110	4E	n	110 1110	6E
SI	000 1111	0F	/	010 1111	2F	O	100 1111	4F	o	110 1111	6F
DLE	001 0000	10	0	011 0000	30	P	101 0000	50	p	111 0000	70
DC1	001 0001	11	1	011 0001	31	Q	101 0001	51	q	111 0001	71
DC2	001 0010	12	2	011 0010	32	R	101 0010	52	r	111 0010	72
DC3	001 0011	13	3	011 0011	33	S	101 0011	53	s	111 0011	73
DC4	001 0100	14	4	011 0100	34	T	101 0100	54	t	111 0100	74
NAK	001 0101	15	5	011 0101	35	U	101 0101	55	u	111 0101	75
SYN	001 0110	16	6	011 0110	36	V	101 0110	56	v	111 0110	76
ETB	001 0111	17	7	011 0111	37	W	101 0111	57	w	111 0111	77
CAN	001 1000	18	8	011 1000	38	X	101 1000	58	x	111 1000	78
EM	001 1001	19	9	011 1001	39	Y	101 1001	59	y	111 1001	79
SUB	001 1010	1A	:	011 1010	3A	Z	101 1010	5A	z	111 1010	7A
ESC	001 1011	1B	;	011 1011	3B	[	101 1011	5B	{	111 1011	7B
FS	001 1100	1C	<	011 1100	3C	\	101 1100	5C	|	111 1100	7C
GS	001 1101	1D	=	011 1101	3D	]	101 1101	5D	}	111 1101	7D
RS	001 1110	1E	>	011 1110	3E	^	101 1110	5E	~	111 1110	7E
US	001 1111	1F	?	011 1111	3F	_	101 1111	5F	DEL	111 1111	7F

Figure 3.25 (continued)

Abbreviations for Control Characters

NUL	null, or all zeros	**FF**	form feed	**CAN**	cancel
SOH	start of heading	**CR**	carriage return	**EM**	end of medium
STX	start of text	**SO**	shift out	**SUB**	substitute
ETX	end of text	**SI**	shift in	**ESC**	escape
EOT	end of transmission	**DLE**	data link escape	**FS**	file separator
ENQ	enquiry	**DC1**	device control 1	**GS**	group separator
ACK	acknowledge	**DC2**	device control 2	**RS**	record separator
BEL	bell	**DC3**	device control 3	**US**	unit separator
BS	backspace	**DC4**	device control 4	**SP**	space
HT	horizontal tabulation	**NAK**	negative acknowledge	**DEL**	delete
LF	line feed	**SYN**	synchronous idle		
VT	vertical tabulation	**ETB**	end of transmission block		

Figure 3.26

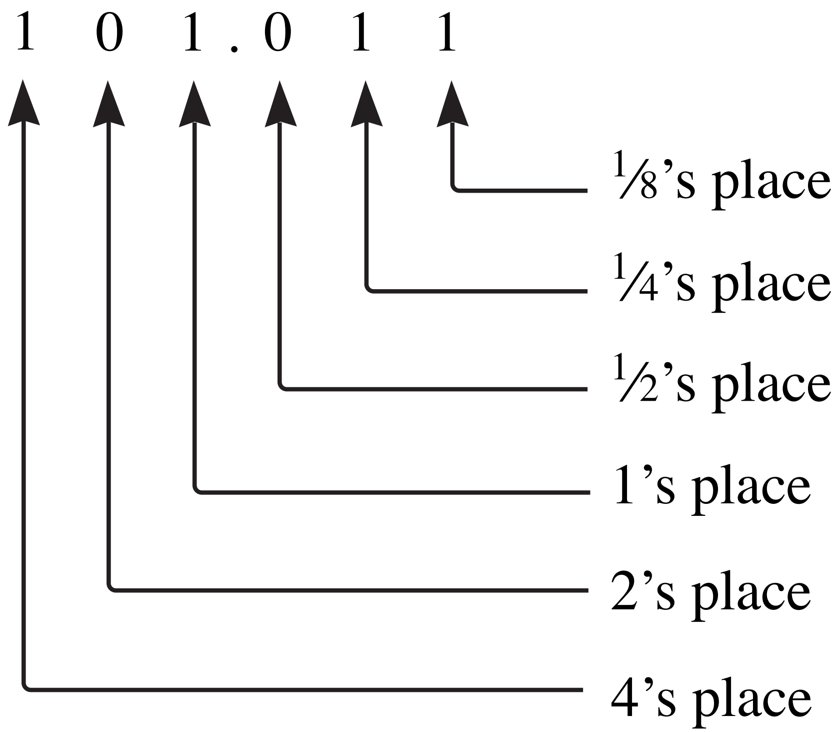
Unicode Script	Code Point	Glyphs							
		0	1	2	3	4	5	6	7
Arabic	U+063_	ذ	ر	ز	س	ش	ص	ض	ط
Armenian	U+054_	Հ	Ձ	Ղ	Ճ	Մ	Յ	Ն	Շ
Braille Patterns	U+287_	⡰	⡱	⡲	⡳	⡴	⡵	⡶	⡷
CJK Unified	U+4EB_	亰	亱	亲	亳	亴	亵	亶	亷
Cyrillic	U+041_	А	Б	В	Г	Д	Е	Ж	З
Egyptian Hieroglyphs	U+1300_	𓀀	𓀁	𓀂	𓀃	𓀄	𓀅	𓀆	𓀇
Emoticons	U+1F61_	😐	😑	😒	😓	😔	😕	😖	😗
Hebrew	U+05D_	א	ב	ג	ד	ה	ו	ז	ח
Basic Latin (ASCII)	U+004_	@	A	B	C	D	E	F	G
Latin-1 Supplement	U+00E_	à	á	â	ã	ä	å	æ	ç

UTF-8 encoding

Bits	First Code Point	Last Code Point	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	0xxxxxxx			
11	U+0080	U+07FF	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Floating point representation

Figure 3.28



1	⅛'s place	= 0.125
1	¼'s place	= 0.25
0	½'s place	= 0.0
1	1's place	= 1.0
0	2's place	= 0.0
1	4's place	= 4.0
		5.375 (dec)

(a) The place values for 101.011 (bin).

(b) Converting 101.011 (bin) to decimal.

Figure 3.29

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

(a) The binary number 101.011.

$$5 \times 10^2 + 0 \times 10^1 + 6 \times 10^0 + 7 \times 10^{-1} + 2 \times 10^{-2} + 1 \times 10^{-3}$$

(b) The decimal number 506.721.

Figure 3.30

6.5859375

$\uparrow$

6 (dec) = 110 (bin)

(a) Convert the whole part

	.5859375
1	.171875
0	.34375
0	.6875
1	.375
0	.75
1	.5
1	.0

(b) Convert the fractional part

Normalized

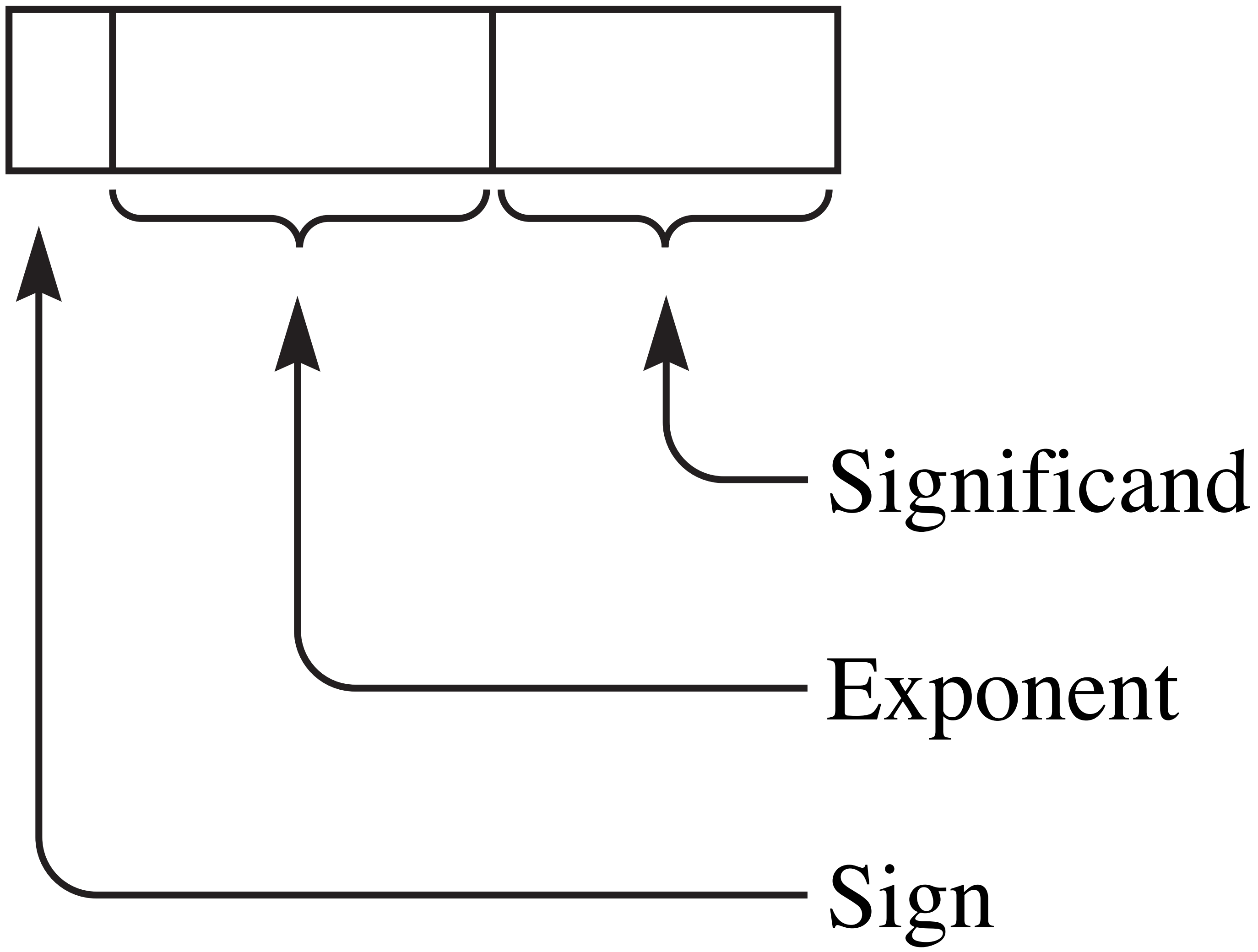
- Leading 1 on the left of the binary point
- 6.5859375 (dec) = 110.1001011 (bin)
- Normalized scientific notation:

  1.101001011×2^2

Figure 3.31

	.2
0	.4
0	.8
1	.6
1	.2
0	.4
0	.8
1	.6
⋮	⋮

Figure 3.32



The hidden bit

- Normalized scientific notation always has 1 to the left of the binary point
- So, do not store it
- Increases precision in the significand
- Floating point unit inserts hidden bit before doing computation
- Floating point unit removes leading 1 from significand before storing result

Figure 3.33

Decimal	Excess 3	Two's Complement
−4		100
−3	000	101
−2	001	110
−1	010	111
0	011	000
1	100	001
2	101	010
3	110	011
4	111	

Round to nearest Ties to even

Decimal	Decimal Rounded	Binary	Binary Rounded
23.499	23	1011.011	1011
23.5	24	1011.1	1100
23.501	24	1011.101	1100
24.499	24	1100.011	1100
24.5	24	1100.1	1100
24.501	25	1100.101	1101

Special value

- Zero
  - Exponent field all 0's
  - Significand all 0's
  - There is a +0 and a –0

Computer Systems FIFTH EDITION

Figure 3.35

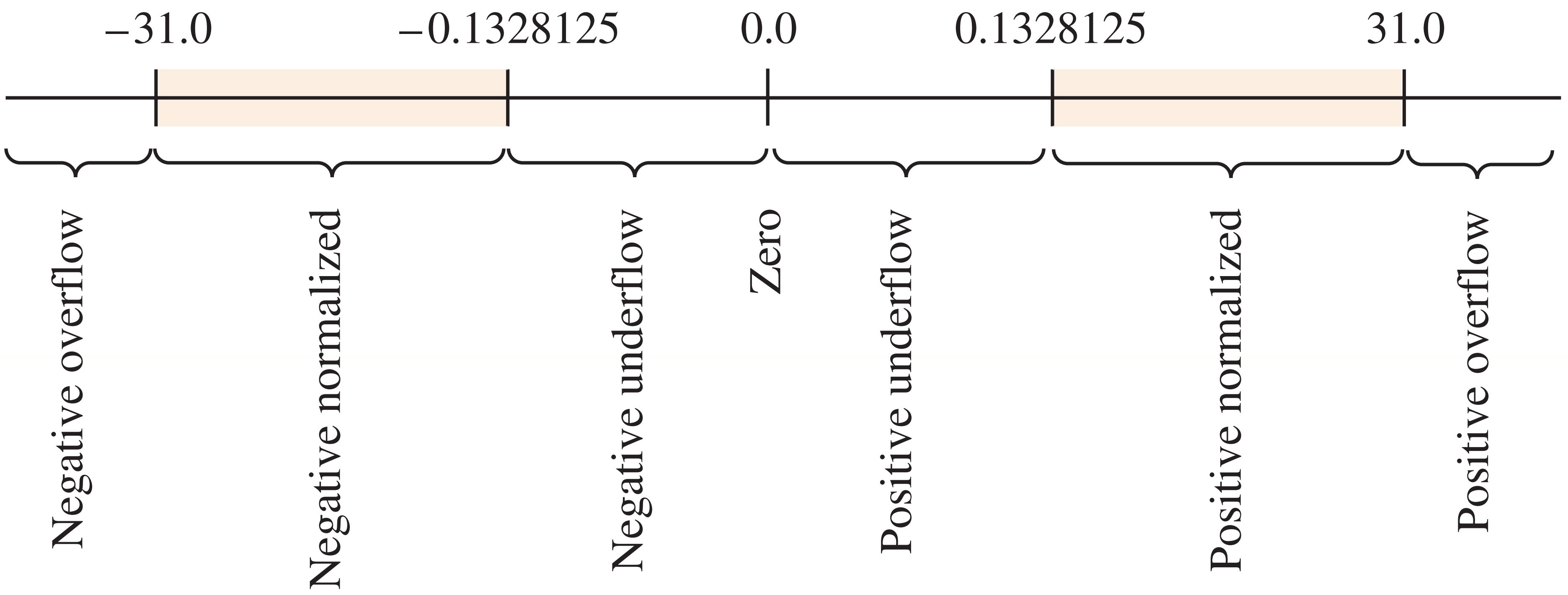


Figure 3.36

Special Value	Exponent	Significand
Zero	All zeros	All zeros
Denormalized	All zeros	Nonzero
Infinity	All ones	All zeros
Not a number	All ones	Nonzero

Special value

- Infinity
  - Exponent field all 1's
  - Significand all 0's
  - There is a $+\infty$ and a $-\infty$
  - Produced by operation that gives result in overflow region

Special value

- Not a Number (NaN)
  - Exponent field all 1's
  - Significand nonzero
  - Produced by illegal math operations

Special value

- Denormalized number
  - Exponent field all 0’s
  - Significand nonzero
  - Hidden bit is assumed to be 0 instead of 1
  - If the exponent is stored in excess *n* for normalized numbers, it is stored in excess *n* – 1 for denormalized numbers

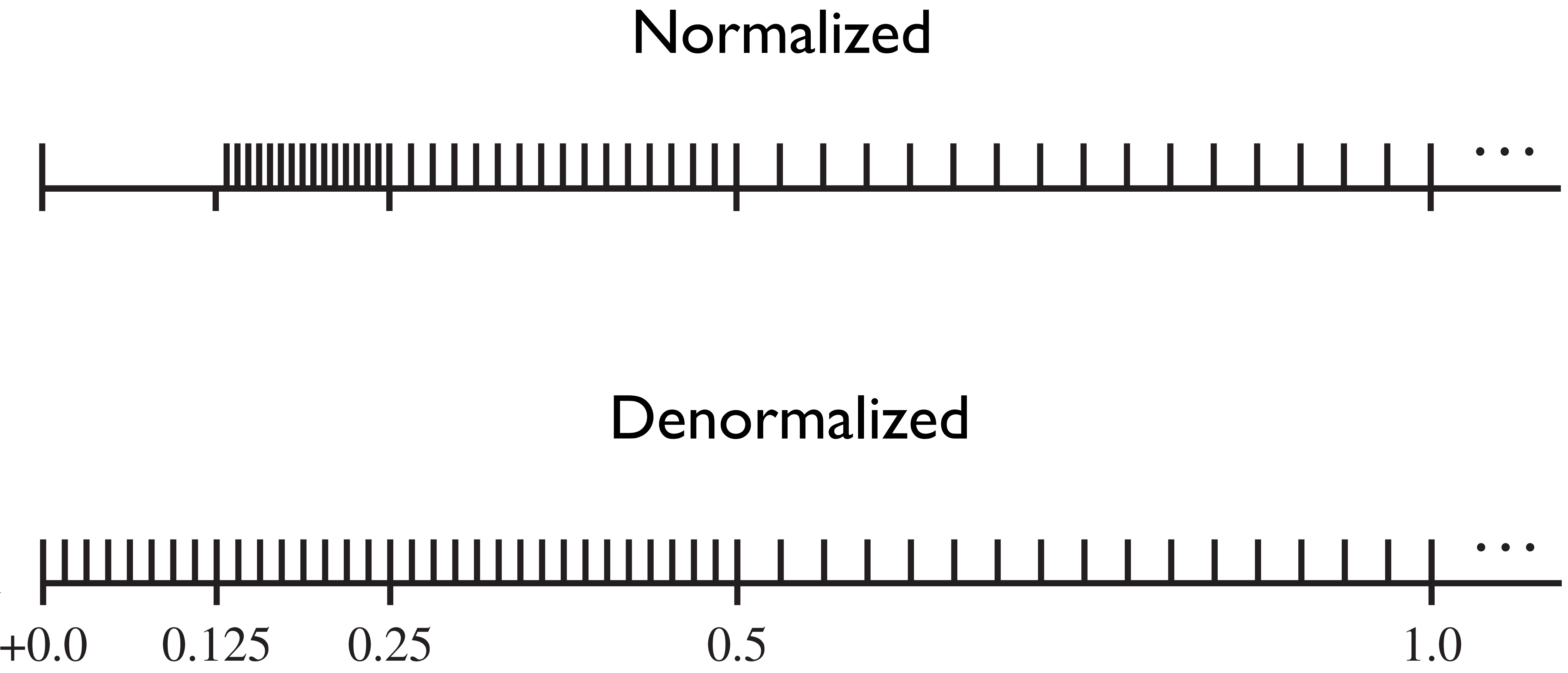

Normalized
Denormalized
+0.0
0.125
0.25
0.5
1.0

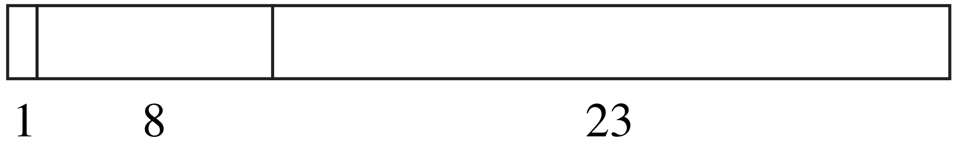
Figure 3.38

	Binary	Scientific Notation	Decimal
Not a number	1 111 nonzero		
Negative infinity	1 111 0000		$-\infty$
Negative normalized	1 110 1111	-1.1111×2^{3}	−15.5
	1 110 1110	-1.1110×2^{3}	−15.0
	. . .	. . .	. . .
	1 011 0001	-1.0001×2^{0}	−1.0625
	1 011 0000	-1.0000×2^{0}	−1.0
	1 010 1111	-1.1111×2^{-1}	−0.96875
	. . .	. . .	. . .
	1 001 0001	-1.0001×2^{-2}	−0.265625
	1 001 0000	-1.0000×2^{-2}	−0.25
Negative denormalized	1 000 1111	-0.1111×2^{-2}	−0.234375
	1 000 1110	-0.1110×2^{-2}	−0.21875
	. . .	. . .	. . .
	1 000 0010	-0.0010×2^{-2}	−0.03125
	1 000 0001	-0.0001×2^{-2}	−0.015625
Negative zero	1 000 0000		−0.0

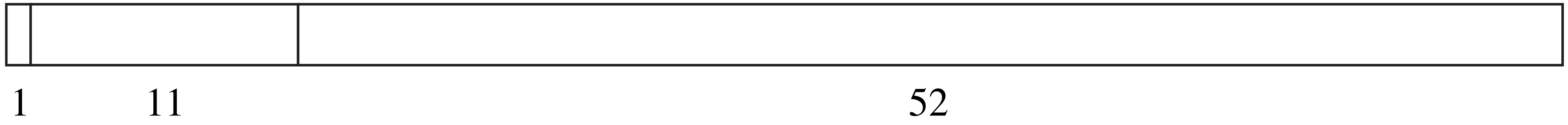
Figure 3.38 (continued)

Positive zero	0 000 0000		+0.0
Positive denormalized	0 000 0001	0.0001×2^{-2}	0.015625
	0 000 0010	0.0010×2^{-2}	0.03125
	. . .	. . .	. . .
	0 000 1110	0.1110×2^{-2}	0.21875
	0 000 1111	0.1111×2^{-2}	0.234375
Positive normalized	0 001 0000	1.0000×2^{-2}	0.25
	0 001 0001	1.0001×2^{-2}	0.265625
	. . .	. . .	. . .
	0 010 1111	1.1111×2^{-1}	0.96875
	0 011 0000	1.0000×2^{0}	1.0
	0 011 0001	1.0001×2^{0}	1.0625
	. . .	. . .	. . .
	0 110 1110	1.1110×2^{3}	15.0
	0 110 1111	1.1111×2^{3}	15.5
Positive infinity	0 111 0000		$+\infty$
Not a number	0 111 nonzero		

IEEE 754 floating point



(a) Single precision



(b) Double precision

Single precision

- C type: `float`
- Exponent: 8-bit cell
  - Excess 127 representation
  - Excess 126 for denormalized numbers
- Exponent: 8-bit cell
  - Significand: 23-bit cell

Double precision

- C type: `double`
- Exponent: 11-bit cell
  - Excess 1023 representation
  - Excess 1022 for denormalized numbers
- Exponent: 8-bit cell
  - Significand: 52-bit cell