1001 = –7 +0101 = 5 1110 = –2 (a) (–7) + (+5)	1100 = –4 +0100 = 4 10000 = 0 (b) (–4) + (+4)
0011 = 3 +0100 = 4 0111 = 7 (c) (+3) + (+4)	1100 = –4 +1111 = –1 11011 = –5 (d) (–4) + (–1)
0101 = 5 +0100 = 4 1001 = Overflow (e) (+5) + (+4)	1001 = –7 +1010 = –6 10011 = Overflow (f) (–7) + (–6)

Figure 9.3 Addition of Numbers in Twos Complement Representation

```
     0010 =  2
    +1001 = –7
     1011 = –5

(a) M = 2 = 0010
    S = 7 = 0111
   –S =     1001
```

```
     0101 =  5
    +1110 = –2
    10011 =  3

(b) M = 5 = 0101
    S = 2 = 0010
   –S =     1110
```

```
     1011 = –5
    +1110 = –2
    11001 = –7

(c) M =–5 = 1011
    S = 2 = 0010
   –S =     1110
```

```
     0101 = 5
    +0010 = 2
     0111 = 7

(d) M = 5 = 0101
    S =–2 = 1110
   –S =     0010
```

```
     0111 = 7
    +0111 = 7
     1110 = Overflow

(e) M =  7 = 0111
    S = –7 = 1001
   –S =      0111
```

```
     1010 = –6
    +1100 = –4
    10110 = Overflow

(f) M = –6 = 1010
    S =  4 = 0100
   –S =      1100
```

Figure 9.4 Subtraction of Numbers in Twos Complement Representation (M – S)

```
    1001  (9)
   ×0011  (3)
00001001  1001 × 2^0
00010010  1001 × 2^1
00011011  (27)
```

(a) Unsigned integers

```
    1001  (–7)
   ×0011  (3)
11111001  (–7) × 2^0 = (–7)
11110010  (–7) × 2^1 = (–14)
11101011  (–21)
```

(b) Twos complement integers

Figure 9.11 Comparison of Multiplication of Unsigned and Twos Complement Integers

```
    0111
   ×0011    (0)
11111001    1–0
0000000     1–1
000111      0–1
00010101    (21)
```

(a) (7) × (3) = (21)

```
    0111
   ×1101    (0)
11111001    1–0
0000111     0–1
111001      1–0
11101011    (–21)
```

(b) (7) × (–3) = (–21)

```
    1001
   ×0011    (0)
00000111    1–0
0000000     1–1
111001      0–1
11101011    (–21)
```

(c) (–7) × (3) = (–21)

```
    1001
   ×1101    (0)
00000111    1–0
1111001     0–1
000111      1–0
00010101    (21)
```

(d) (–7) × (–3) = (21)

Figure 9.14 Examples Using Booth's Algorithm

A	Q	M = 0011
0000	0111	Initial value
0000	1110	shift
1101		subtract
0000	1110	restore
0001	1100	shift
1110		subtract
0001	1100	restore
0011	1000	shift
0000		subtract
0000	1001	set $Q_0 = 1$
0001	0010	shift
1110		subtract
0001	0010	restore

(a) (7)/(3)

A	Q	M = 1101
0000	0111	Initial value
0000	1110	shift
1101		add
0000	1110	restore
0001	1100	shift
1110		add
0001	1100	restore
0011	1000	shift
0000		add
0000	1001	set $Q_0 = 1$
0001	0010	shift
1110		add
0001	0010	restore

(b) (7)/(–3)

A	Q	M = 0011
1111	1001	Initial value
1111	0010	shift
0010		add
1111	0010	restore
1110	0100	shift
0001		add
1110	0100	restore
1100	1000	shift
1111		add
1111	1001	set $Q_0 = 1$
1111	0010	shift
0010		add
1111	0010	restore

(c) (–7)/(3)

A	Q	M = 1101
1111	1001	Initial value
1111	0010	shift
0010		subtract
1111	0010	restore
1110	0100	shift
0001		subtract
1110	0100	restore
1100	1000	shift
1111		subtract
1111	1001	set $Q_0 = 1$
1111	0010	shift
0010		subtract
1111	0010	restore

(d) (–7)/(–3)

Figure 9.17 Examples of Twos Complement Division

$x = 1.000.....00 \times 2^{1}$ $\underline{-y} = \underline{0.111.....11} \times 2^{1}$ $z = 0.000.....01 \times 2^{1}$ $= 1.000.....00 \times 2^{-22}$ (a) Binary example, without guard bits	$x = .100000 \times 16^{1}$ $\underline{-y} = \underline{.0FFFFF} \times 16^{1}$ $z = .000001 \times 16^{1}$ $= .100000 \times 16^{-4}$ (c) Hexadecimal example, without guard bits
$x = 1.000.....00\ 0000 \times 2^{1}$ $\underline{-y} = \underline{0.111.....11\ 1000} \times 2^{1}$ $z = 0.000.....00\ 1000 \times 2^{1}$ $= 1.000.....00\ 0000 \times 2^{-23}$ (b) Binary example, with guard bits	$x = .100000\ 00 \times 16^{1}$ $\underline{-y} = \underline{.0FFFFF\ F0} \times 16^{1}$ $z = .000000\ 10 \times 16^{1}$ $= .100000\ 00 \times 16^{-5}$ (d) Hexadecimal example, with guard bits

Figure 9.25 The Use of Guard Bits