Number, Arithmetic Operations, and Characters

Signed Integer

- 3 major representations:

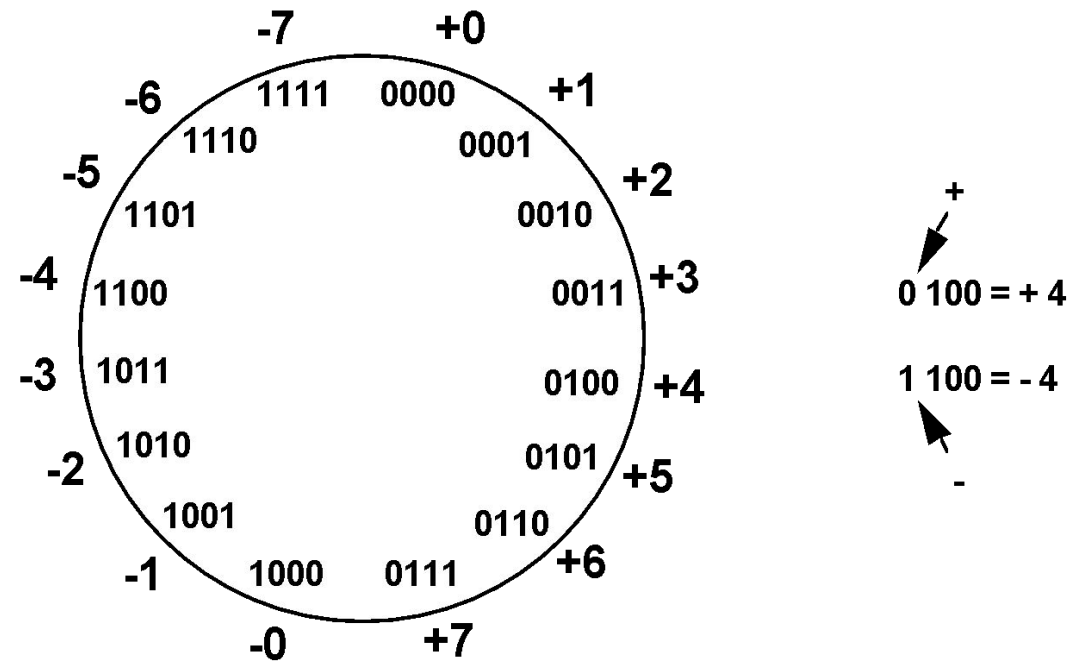 Sign and magnitude

  One's complement

  Two's complement

- Assumptions:

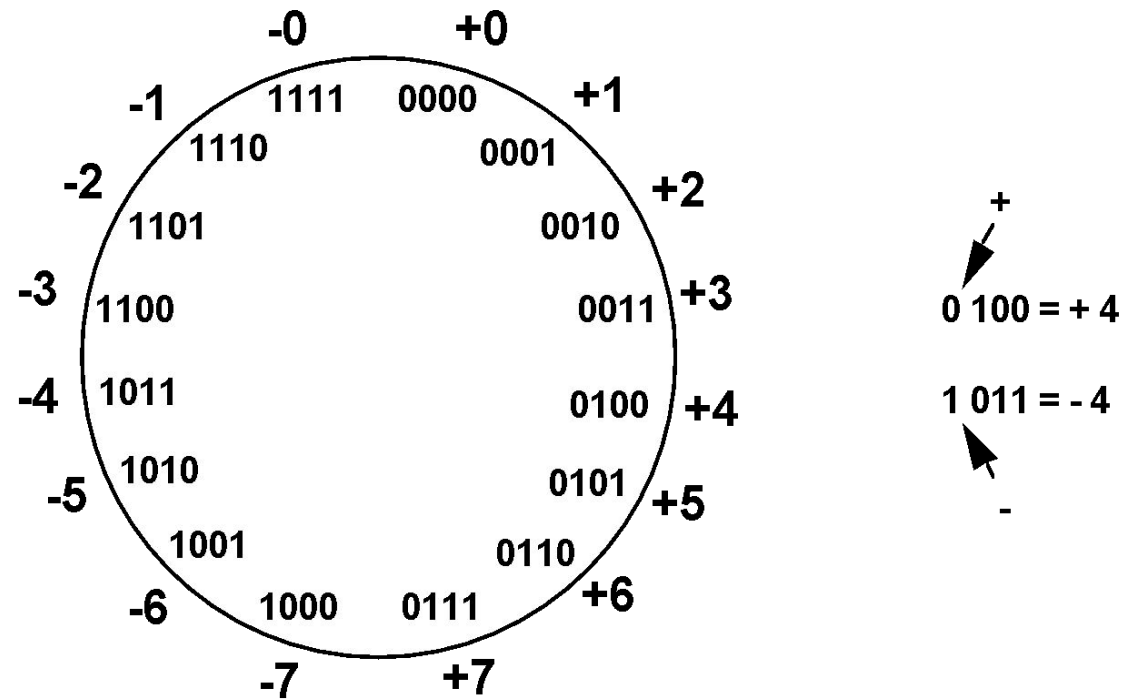 4-bit machine word

  16 different values can be represented

  Roughly half are positive, half are negative

Sign and Magnitude Representation

High order bit is sign: 0 = positive (or zero), 1 = negative
Three low order bits is the magnitude: 0 (000) thru 7 (111)
Number range for n bits = +/-2^{n-1} -1
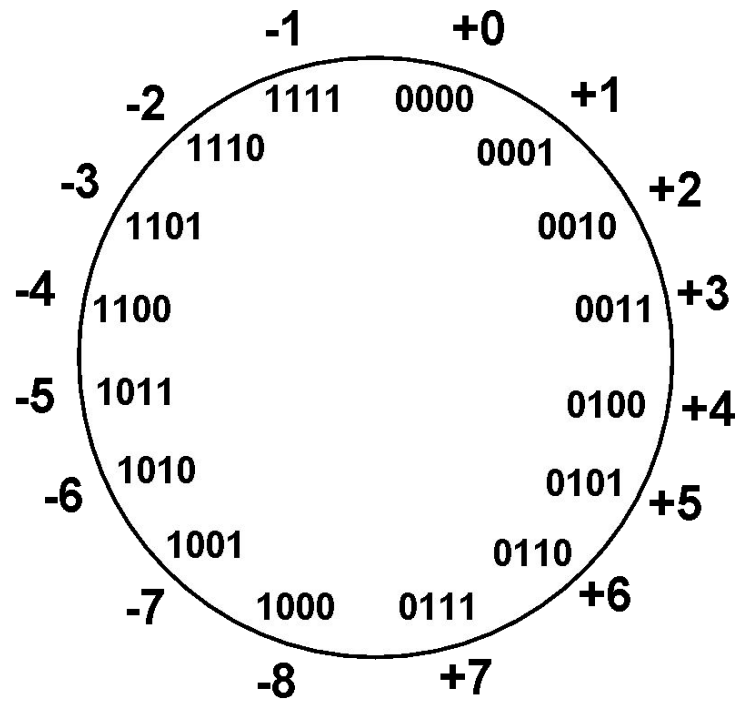**Two representations for 0**

One's Complement Representation

- Subtraction implemented by addition & 1's complement
- Still two representations of 0! This causes some problems
- Some complexities in addition

Two's Complement Representation

like 1's comp
except shifted
one position
clockwise

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

+
0 100 = + 4
1 100 = - 4
-

- Only one representation for 0

- One more negative number than positive number

Binary, Signed-Integer Representations

Page 28

$b_3 b_2 b_1 b_0$	Sign and magnitude	1's complement	2's complement
	B	Values represented	
0 1 1 1	+ 7	+ 7	+ 7
0 1 1 0	+ 6	+ 6	+ 6
0 1 0 1	+ 5	+ 5	+ 5
0 1 0 0	+ 4	+ 4	+ 4
0 0 1 1	+ 3	+ 3	+ 3
0 0 1 0	+ 2	+ 2	+ 2
0 0 0 1	+ 1	+ 1	+ 1
0 0 0 0	+ 0	+ 0	+ 0
1 0 0 0	- 0	- 7	- 8
1 0 0 1	- 1	- 6	- 7
1 0 1 0	- 2	- 5	- 6
1 0 1 1	- 3	- 4	- 5
1 1 0 0	- 4	- 3	- 4
1 1 0 1	- 5	- 2	- 3
1 1 1 0	- 6	- 1	- 2
1 1 1 1	- 7	- 0	- 1

Figure 2.1. Binary, signed-integer representations.

Addition and Subtraction – 2's Complement

	4	0100	-4	1100
	+ 3	0011	+ (-3)	1101
	7	0111	-7	11001

If carry-in to the high order bit = carry-out then ignore carry

if carry-in differs from carry-out then overflow

	4	0100	-4	1100
	- 3	1101	+ 3	0011
	1	10001	-1	1111

Simpler addition scheme makes twos complement the most common choice for integer number systems within digital systems

2's-Complement Add and Subtract Operations

Page 31

(a)
```
  0 0 1 0      (+2)
+ 0 0 1 1      (+3)
─────────      ─────
  0 1 0 1      (+5)
```

(b)
```
  0 1 0 0      (+4)
+ 1 0 1 0      (- 6)
─────────      ─────
  1 1 1 0      (- 2)
```

(c)
```
  1 0 1 1      (- 5)
+ 1 1 1 0      (- 2)
─────────      ─────
  1 0 0 1      (- 7)
```

(d)
```
  0 1 1 1      (+7)
+ 1 1 0 1      (- 3)
─────────      ─────
  0 1 0 0      (+4)
```

(e)
```
  1 1 0 1      (- 3)
- 1 0 0 1      (- 7)
─────────      ─────
```
⟹
```
  1 1 0 1
+ 0 1 1 1
─────────      ─────
  0 1 0 0      (+4)
```

(f)
```
  0 0 1 0      (+2)
- 0 1 0 0      (+4)
─────────      ─────
```
⟹
```
  0 0 1 0
+ 1 1 0 0
─────────      ─────
  1 1 1 0      (- 2)
```

(g)
```
  0 1 1 0      (+6)
- 0 0 1 1      (+3)
─────────      ─────
```
⟹
```
  0 1 1 0
+ 1 1 0 1
─────────      ─────
  0 0 1 1      (+3)
```

(h)
```
  1 0 0 1      (- 7)
- 1 0 1 1      (- 5)
─────────      ─────
```
⟹
```
  1 0 0 1
+ 0 1 0 1
─────────      ─────
  1 1 1 0      (- 2)
```

(i)
```
  1 0 0 1      (- 7)
- 0 0 0 1      (+1)
─────────      ─────
```
⟹
```
  1 0 0 1
+ 1 1 1 1
─────────      ─────
  1 0 0 0      (- 8)
```

(j)
```
  0 0 1 0      (+2)
- 1 1 0 1      (- 3)
─────────      ─────
```
⟹
```
  0 0 1 0
+ 0 0 1 1
─────────      ─────
  0 1 0 1      (+ 5)
```

Figure 2.4. 2's-complement Add and Subtract operations.