Signed Integer Representation

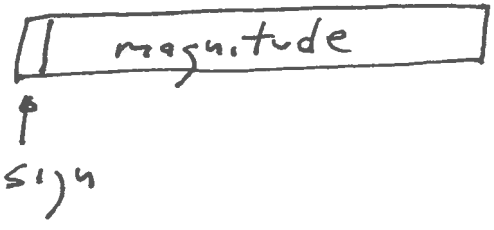
CS520

Dept. of Computer Science
Univ. of New Hampshire

We will consider three possibilities:

1. sign magnitude
2. one's complement
3. two's complement

sign magnitude



15_{10} = 0000 1111

-15_{10} = 1000 1111

$37_{10} = 25_{16}$ = 0 0010 0101

$+(-15_{10})$ + 1 0000 1111

22_{10} 0 0001 0110 = 22_{10}

negate: easy

add: test sign bits

if same sign,
add magnitudes
& keep sign

if different signs,
subtract smaller
magnitude from
larger & keep
sign of larger

Sign magnitude: summary

negate: easy

addition: complicated

two representations of zero

0000 0000 +0

1000 0000 -0

one's complement

express absolute value

if negative, complement all bits

↳ i.e. flip/reverse

negate: complement all bits

addition: like unsigned <u>except</u> if 1 carried out of leftmost bit position, add 1 to rightmost bit position

$15_{10} = 0000\ 1111$

$-15_{10} = 1111\ 0000$

$37_{10} = 25_{16} = 0010\ 0101$

$+(-15_{10})$ $+1111\ 0000$

22_{10} $0001\ 0101$

$+1$

$0001\ 0110 = 16_{16} = 22_{10}$

10_{10} $0000\ 0101$

$+(-5_{10})$ $+5$ $+1111\ 1010$

5_{10} $0000\ 0100$

$+1$

$0000\ 0101 = 5$

$$\begin{array}{r} 5_{10} \\ +(-10_{10}) \\ \hline -5_{10} \end{array}$$

$$\underbrace{00001010}_{+10} + 11110101$$

$$\begin{array}{r} {\scriptstyle 1\ \ 1\ \ \ \ \ \ \ \ } \\ 00001010 \\ +\ 11110101 \\ \hline 11111010 = -5 \end{array}$$

no carry, so no second addition

$$00000101 = +5$$

One's complement: Summary

negate: easy, do all bits at same time

addition: adding in carry could take as much time as original add

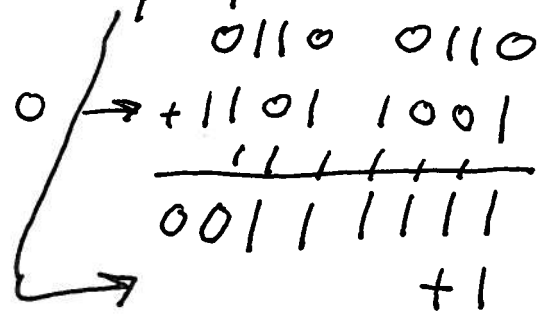
$102_{10} = 66_{16}$

$+(-38_{10}) = -26_{16}$

64_{10}

0010 0110

+26



$40_{16} = 64_{10}$

One's complement: more summary

two representations for zero

0000 0000	+0
1111 1111	-0

sign bit is still leftmost bit

0 | 000 1111	1000 0000	if positive, it is too big to fit in 8 bits for signed integer
1 | 111 0000	0111 1111	

two's complement

express absolute value

if negative, complement and add 1

^always

$+15_{10} = 0000\ 1111$

$-15_{10} = 1111\ 0000$

$+1$

$1111\ 0001$

negate: complement and add 1

add: same as unsigned, ignore carry

$37_{10} = 25_{16} = 0010\ 0101$

$+(-15_{10}) = +\ 1111\ 0001$

$22_{10} = 0001\ 0110$

(1) ignored!

$16_{16} = 22_{10}$

two's complement: summary

negate: more complicated

addition: easy

```
 0000 0000 = ~0
(1)1111 1111
ignored.   +1
-----------
0000 0000 = 0
```

now only one representation of zero

but one more negative number than positive

```
1111 1111 = -1
0000 0000
     +1
---------
0000 0001
```

```
1000 0000 = -128_10
0111 1111
     +1
---------
1000 0000  ← overflow
```

$2 + 128_{10}$

$-128_{10} \ldots -1\ 0\ 1 \ldots 127_{10}$

two's complement: more summary

leftmost b.t still indicates sign

1111 1111 = -1

0000 0000

1

0000 0001 +1

all modern machines are two's complement

moving values between different sized containers

two's complement

promotion (small → large): sign extend

0000 1111 → 0000 0000 0000 1111

1111 0001 → 1111 1111 1111 0001 = -15_{10}

demotion (large → small): chop

0000 0000 0000 1111 = $+15_{10}$

overflow?

0000 0000 0000 1111

1111 1111 1111 0001

0000 1110

+1

0000 1111

257_{10} 0000 0001 0000 0001 = 257_{10}

0000 0001 = 1_{10}

overflow if the chopped bits are not all the same.

-257_{10} 1111 1110 1111 1110

+1

1111 1110 1111 1111 = -257_{10}

1111 1111 = -1_{10}