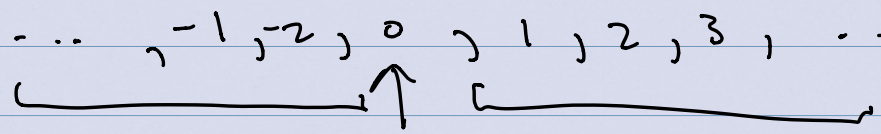
Negative Numbers

int whole numbers



3.781 $\quad$ -1.5 $\quad$ 3.1×10^{-2}

Integers

n-bit numbers

3-bit numbers $\quad$ unsigned

Bin	Dec
000	0 ←
001	1
010	2
011	3
100	4
101	5
110	6
111	7 ←

n-bit number

2^n different values

0 to $2^n - 1$

→ 0 to $2^3 - 1$

0 to $8 - 1$

0 to 7

Negatives $\quad$ 1/2 positive $\quad$ 1/2 negative

Sign magnitude

Bin	Dec
000	0
001	1
010	2
011	3
100	-0
101	-1
110	-2
111	-3

pluses
1) simple
2) easy to determine if a value is negative

if(x == 0) ≤ ⇒ x == 0 || x == +0

2

Arithmetic with sign magnitude

-1 + 3 = 2

```
  1 11
    101
+   011
  ------
  [000]
```

10

-1 + 3 ≠ 0

Two's Complement

Bin	
000	0
001	1
010	2
011	3
100	-4 *
101	-3 *
110	-2
111	-1

Converting pos to neg and neg to pos

2 (pos) what is the 2's comp neg representation

Invert(v) + 1 = 2's comp

3 = 0b11

~0b011 = 0b100

+ 1

0b101 (= 3)

$-3 = 0b101$

$\sim 0b101 = 0b010$

$+1$

$0b011$ (3)

Two's Complement Arithmetic

$-1 + 3 = ?$ $0b111 + 0b011$

```
 1 1 1
  111      (-1)
+ 011      (3)
-----
  010      010 == 2
```

$2 - 4$

$2 + (-4)$

$2 \quad (-4)$

$0b010 + 0b100$

```
  010      (2)
+ 100      (-4)
-----
  110      (-2)
```

$001 + 1 = 010$ (2)

Range for 2's comp signed ints

$-(2^{n-1})$ to $(2^{n-1}-1)$

... 0 ...

Sign Extension and ASR

3-bit	4bit	8-bit
001	0001	0000 0001
011	0011	0000 0011

3-bit -2 010 I(101) +1 = 110

110

4-bit -2 0010 1101 +1 1110

1110

8-bit -2

1111 1110

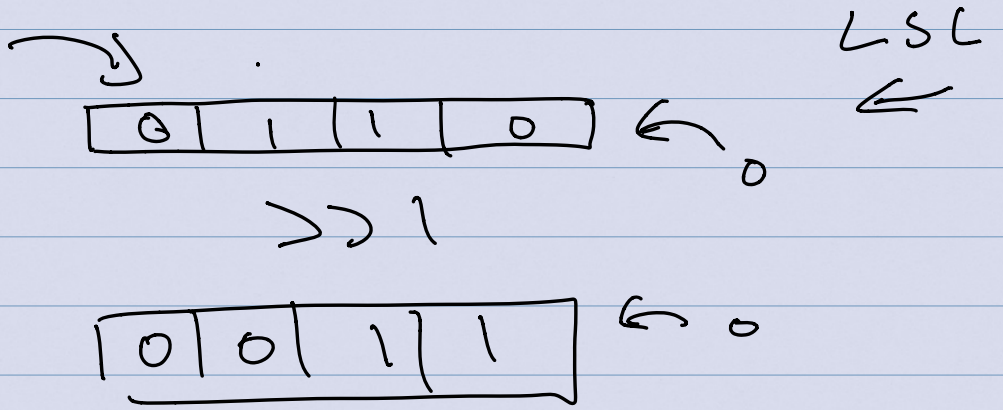
sign extension sign

3bit ⇒ 8bit

1111 1110

ASR Arithmetic Shift Right

LSR Logical Shift Right



---

in C

int x, y $-(2^{31}) \ldots 0 \ldots (2^{31}-1)$

unsigned int ux, uy $0 \ldots 2^{32}-1$

x + y $x = -4$ $y = -8$

ux + uy ux = (unsigned int) x;
uy = (unsigned int) y;

int r; $r = (int)(ux + uy)$

$r = -12$

ASR

in

x >> 1 (ASR) >-

ux >> 1 (LSR) >>

ASR

eval

uint32_t

↓

((int) v1) >> v2

↑ ↗ ↑

int ASR

1100 → 0011 + 1 (0100)

↑ 4

neg