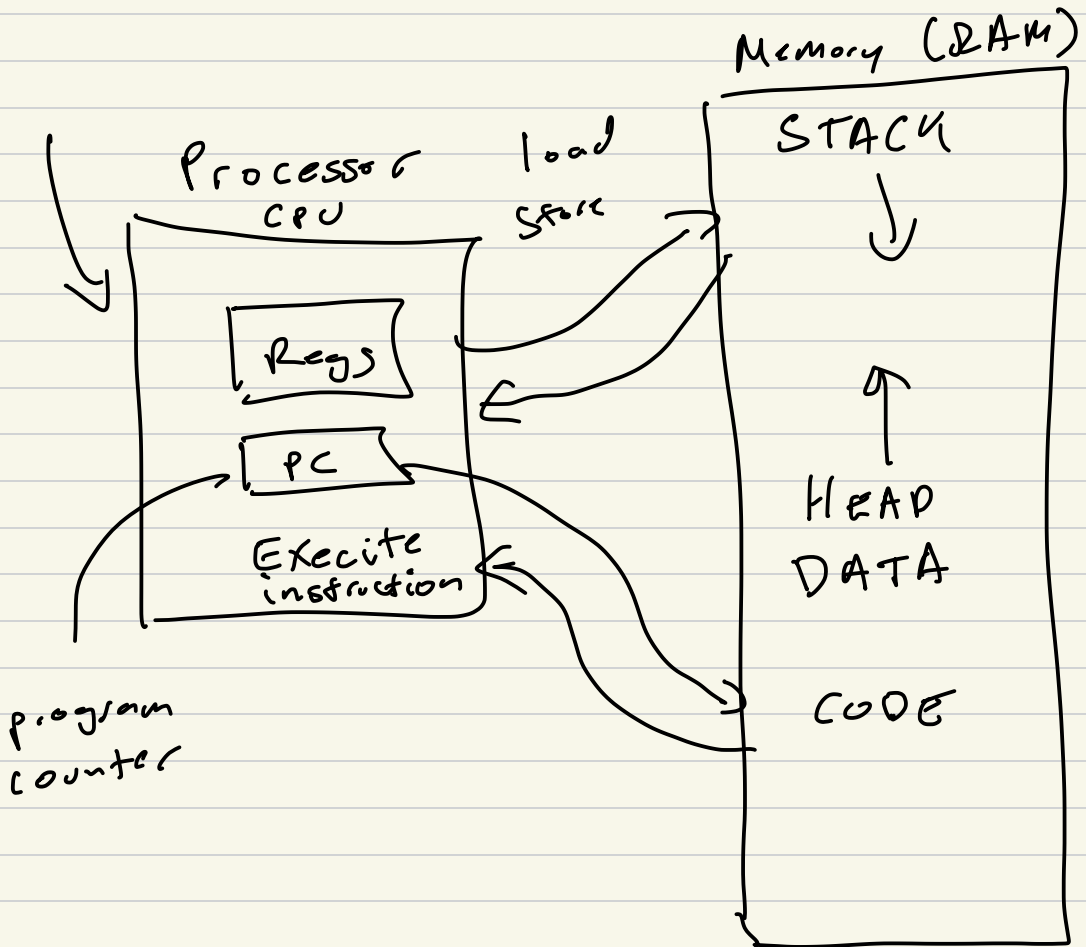


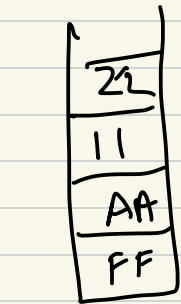
CS 315-02 RISC-V Assembly 6

Memory, 2s complement, strings

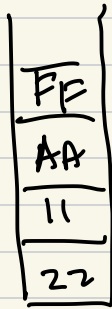


Memory $\rightarrow$ Array of bytes

`int x = 0xFFAA1122`

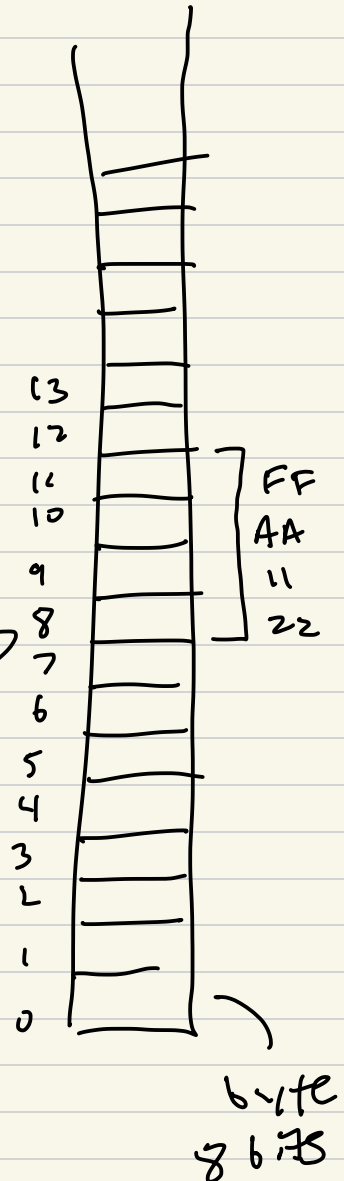


big
endian



little
endian

byte
ordering
endianness



Binary representation of integers

implied to be signed

→ Two's Complement

4-bit numbers

Unsigned Decimal	Binary	signed magnitude	Two's Complement
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-0	-8
9	1001	-1	-7
10	1010	-2	-6
11	1011	-3	-5
12	1100	-4	-4
13	1101	-5	-3
14	1110	-6	-2
15	1111	-7	-1

	①	1 1 1 0 1 0 1 (5)	Two's comp	①	1 1 0 1 0 1 (5)
Signed magnitude		+ 1 0 1 1 (-3)			+ 1 1 0 1 (-3)
		<u>0 0 0 0 (-0)</u>			<u>0 0 1 0 (2)</u>

How to get neg from pos?

$$3 \rightarrow -3$$

$$0011 \rightarrow 1100 + 1 \rightarrow \underline{1101} \quad (-3)$$

$$\text{invert}(0) + 1$$

$$\begin{matrix} (-3) \\ 1101 \end{matrix} \rightarrow 0010 + 1 \rightarrow 0011 \quad (3)$$

Other nice things about 2's comp

$$n \text{ bit} \rightarrow m \text{ bit} \quad \text{where } m > n$$

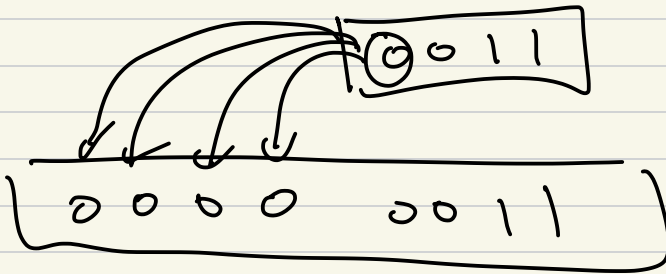
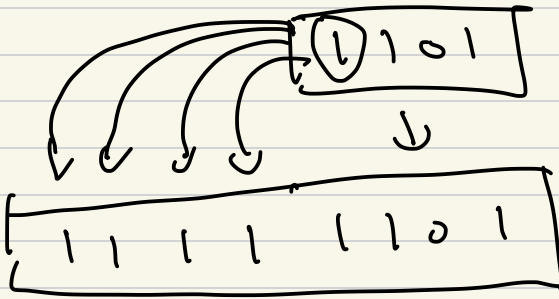
$$4 \text{ bits} \quad \boxed{1101} \quad (-3)$$

$$8 \text{ bits}$$

$$\begin{array}{r} 0000 \ 0011 \ (3) \\ + 1111 \ 1100 \\ \hline \end{array}$$

1111 1101

Sign extension



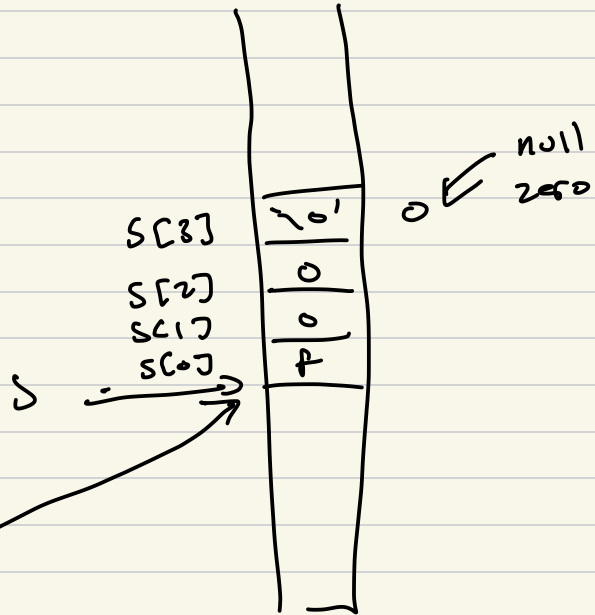
Strings arrays of characters (bytes)

char s = "foo"

1w (32 bits)

1d (64 bits)

1b (8 bits)



1b to, (00)

lower 8 bits
will be the value