

Unsigned and signed integer numbers

Carry

C

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

Subtraction sets C flag opposite of carry (ARM specialty)!

- if (carry = 0) then C=1
- if (carry = 1) then C=0

overflow

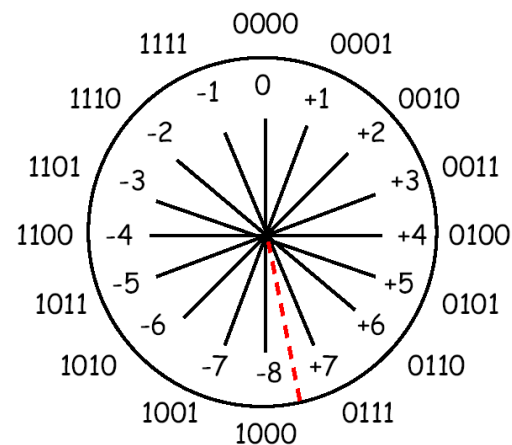
$$\updownarrow V = A_{n-1}B_{n-1}\bar{S}_{n-1} \vee \bar{A}_{n-1}\bar{B}_{n-1}S_{n-1}$$

Unsigned and *signed* integer numbers

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

C ↓

V

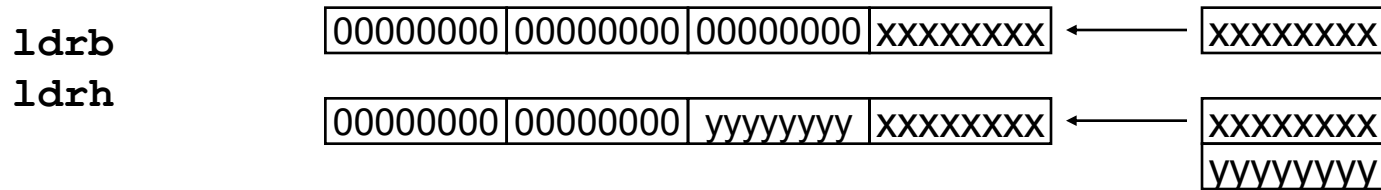


Overflow

$$V = A_{n-1}B_{n-1}\bar{S}_{n-1} \vee \bar{A}_{n-1}\bar{B}_{n-1}S_{n-1}$$

Signed/unsigned extension to 32 bits

- when reading 8 and 16-bit operands from memory – they need to be extended by sign or zeros to full length (registers are 32bit long). Only load and store instructions are accessing operands in memory.
- unsigned operands from memory are extended with zeros:



- signed operands from memory are extended with sign bit:

