1's Complement

- Bitwise inverse of the number as its negative number

NO.	Binary	Unsigned	Signed Magnitude	1's
7	111	+7	-3	-0
6	110	+6	-2	-1
5	101	+5	-1	-2
4	100	+4	-0	-3
3	011	+3	+3	+3
2	010	+2	+2	+2
1	001	+1	+1	+1
0	000	+0	+0	+0

1's Complement

- Bitwise inverse of the number as its negative number

只看符号位/其他位取反

NO.	Binary	Unsigned	Signed Magnitude	1's
7	111	+7	-3	-0
6	110	+6	-2	-1
5	101	+5	-1	-2
4	100	+4	-0	-3
3	011	+3	+3	+3
2	010	+2	+2	+2
1	001	+1	+1	+1
0	000	+0	+0	+0

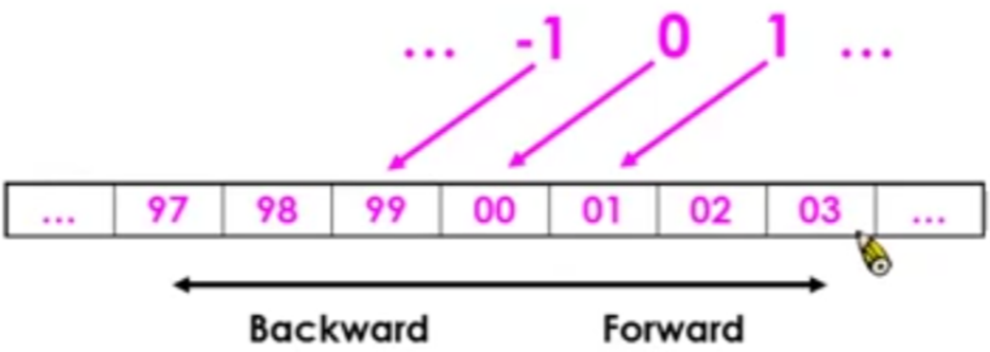
$1\underline{\underline{00}} \rightarrow 0\underline{\underline{11}}$

Signed Number

- Both signed magnitude and 1's complement are not quite suitable for representing the numbers, since…
- 2 zeros (positive zero, negative zero)
- A special adder is required to perform addition
- Ex: 1 + (-1) = 0
  - Signed Magnitude: 001 + 101 = 110 (-2)
  - 1's Complement: 001 + 110 = 111 (-0)
- We need a GOOD representation for signed number

Odometer (2/2)

- Use the odometer to represent negative mileage



- Where is the new zero?
  - 49? 50? 51? ← Definitely a bad Idea! We want to use 00
- Since 99+1 = 00, let's rotate the numbers!

Binary Odometer

- 2 possible representations...

All positive numbers begin in 0

All negative numbers begin in 1

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

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

- Which one is better?

Insight of 2's Complement

3		0	1	1	3		0		3
2		0	1	0	2		0		2
1		0	0	1	1		0		1
0	=	0	0	0	0	+	0	=	0
-1		1	1	1	3		-4		-1
-2		1	1	0	2		-4		-2
-3		1	0	1	1		-4		-3
-4		1	0	0	0		-4		-4

2's Complement

- MSB represents the negative number

B =	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

$$B = -b_7\times2^7 + b_6\times2^6 + b_5\times2^5 + b_4\times2^4 + b_3\times2^3 + b_2\times2^2 + b_1\times2^1 + b_0\times2^0$$

$$(10000000)_2 \le B \le (01111111)_2$$

$$-2^7 \le B \le (2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0) = 2^7-1$$

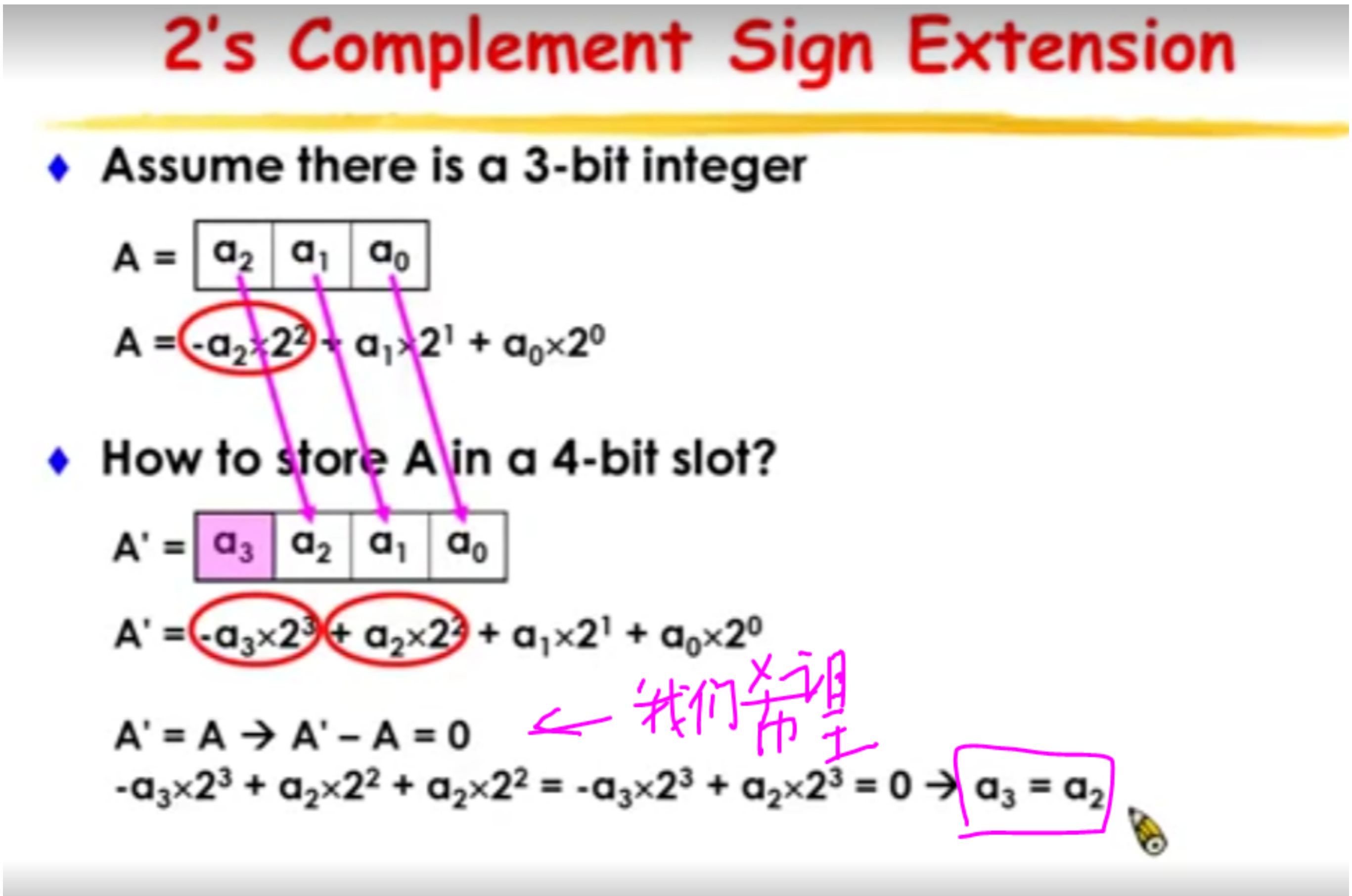
7个1

- For n-bit number
  - $-2^{n-1} \le B \le 2^{n-1}-1$

2's Complement

- 2's complement = 1's complement + 1

NO.	Binary	Unsigned	Signed Magnitude	1's	2's
7	111	+7	-3	-0	-1
6	110	+6	-2	-1	-2
5	101	+5	-1	-2	-3
4	100	+4	-0	-3	-4
3	011	+3	+3	+3	+3
2	010	+2	+2	+2	+2
1	001	+1	+1	+1	+1
0	000	+0	+0	+0	0


2's Complement Sign Extension
Assume there is a 3-bit integer
A = | a2 | a1 | a0 |
A = -a2×2^2 + a1×2^1 + a0×2^0
How to store A in a 4-bit slot?
A' = | a3 | a2 | a1 | a0 |
A' = -a3×2^3 + a2×2^2 + a1×2^1 + a0×2^0
A' = A → A' – A = 0
← 我们希望
-a3×2^3 + a2×2^2 + a2×2^2 = -a3×2^3 + a2×2^3 = 0 → a3 = a2