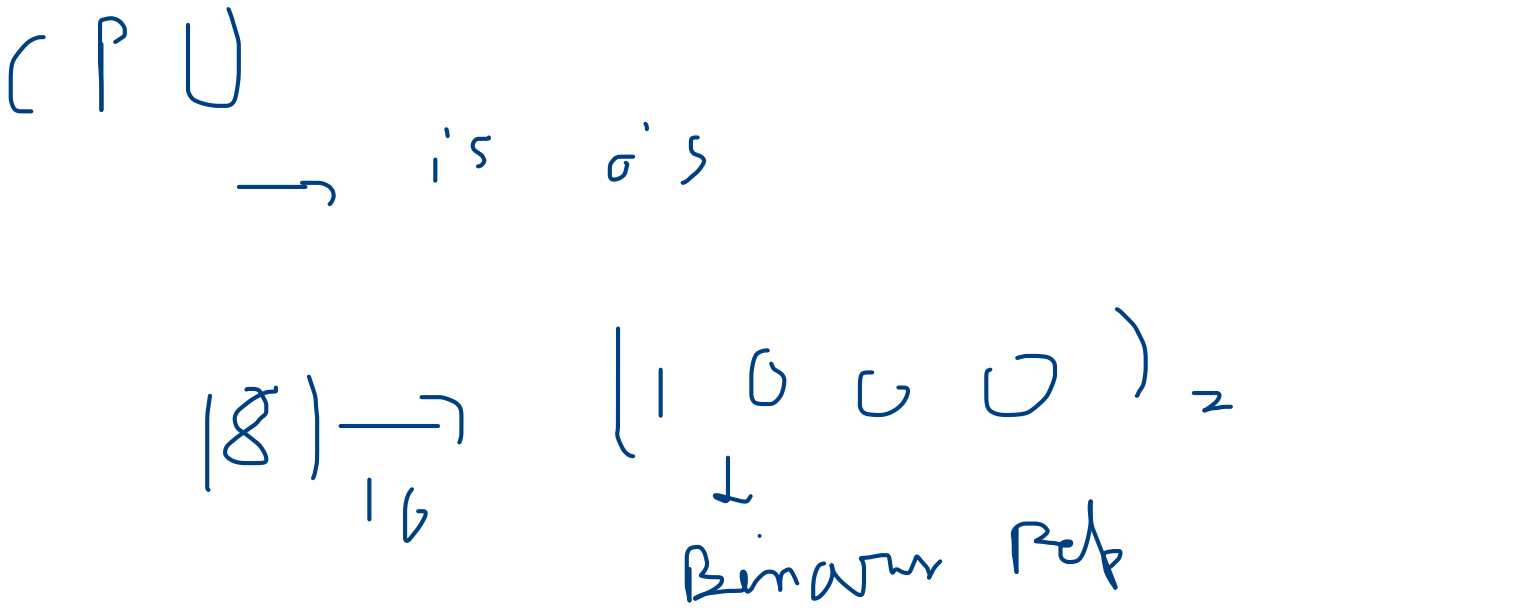
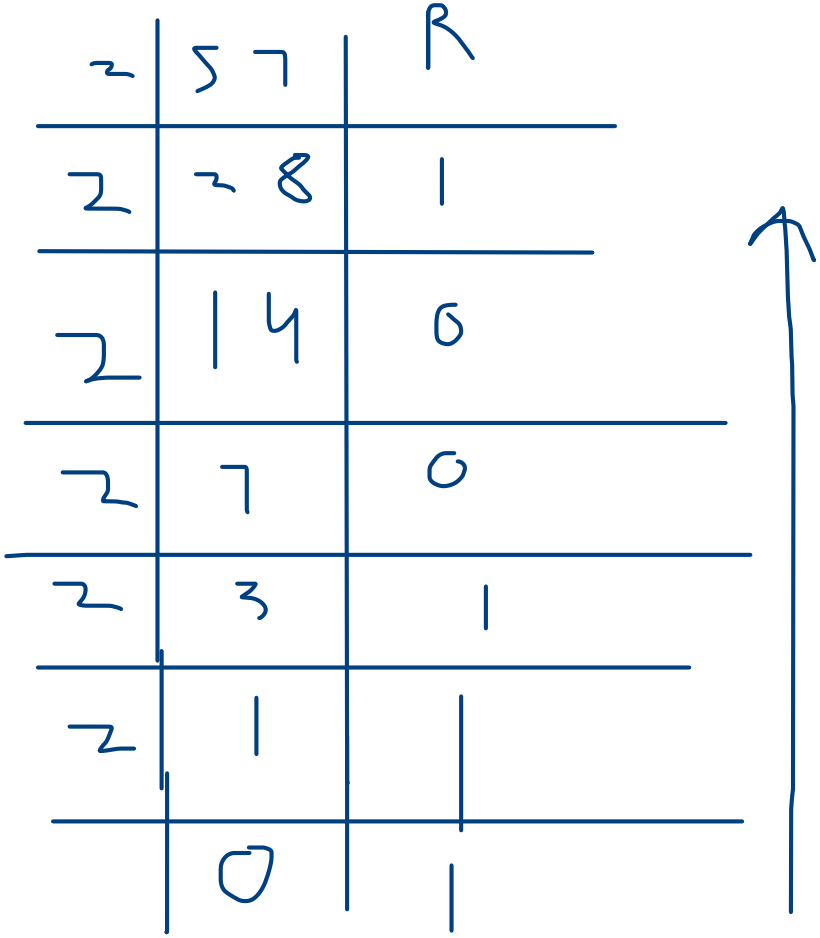
Bit Manipulation



$57_{10} \longrightarrow$



$(57)_{10} \longrightarrow (1\ 1\ 1\ 0\ 0\ 1)_2$

Integer
Family

	byte	short	int	long
4 bits	8 bits	16	32	64
2^4	2^8	2^{16}	2^{32}	2^{64}
16 numbers				

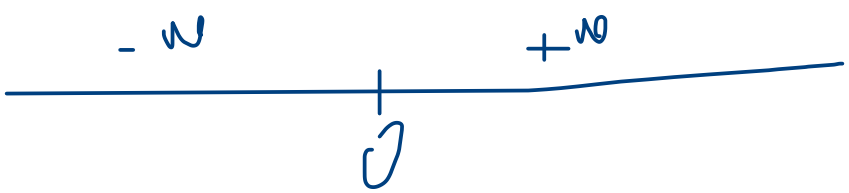
nibble

2^4

= 16

B.	D
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	10
1 0 1 1	11
1 1 0 0	12
1 1 0 1	13
1 1 1 0	14
1 1 1 1	15

-W ?



nibble

2^4

= 16

B.	D	
0 0 0 0	0	+ve
0 0 0 1	1	
0 0 1 0	2	
0 0 1 1	3	
0 1 0 0	4	
0 1 0 1	5	
0 1 1 0	6	
0 1 1 1	7	
1 0 0 0	−0	−ve
1 0 0 1	−1	
1 0 1 0	−2	
1 0 1 1	−3	
1 1 0 0	−4	
1 1 0 1	−5	
1 1 1 0	−6	
1 1 1 1	−7	

A2 → + ±0

1. MSB = 0 --> +ve number
2. MSB = 1 --> -ve number

remaining 3 bits used to make number in decimal

A3

nibble

2^4

= 16

B.	D
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	−8
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

+ve

−ve

MSB 0 1

+ve −ve

look for last 3 bit

2's compliment of last 3 bit

2's = 1's + 1

Comp Comp

1 s comp = toggle bits

$(1000) = (1000)' + 0001$

$= 0111 + 0001 \rightarrow +\ 0001$

$1000 = -8$

A3 ✓

nibble

2^4

= 16

B	D
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	−8
1 0 0 1	−7
1 0 1 0	−6
1 0 1 1	−5
1 1 0 0	−4
1 1 0 1	−3
1 1 1 0	−2
1 1 1 1	−1

+ve

−ve

1010

0101

+ 0001

0110 = 6

$(1001)_{11} = (1001)_1 + 0001$

0110 + 0001 →

0110

0001

0111 = 7

A3 ✓

nibble

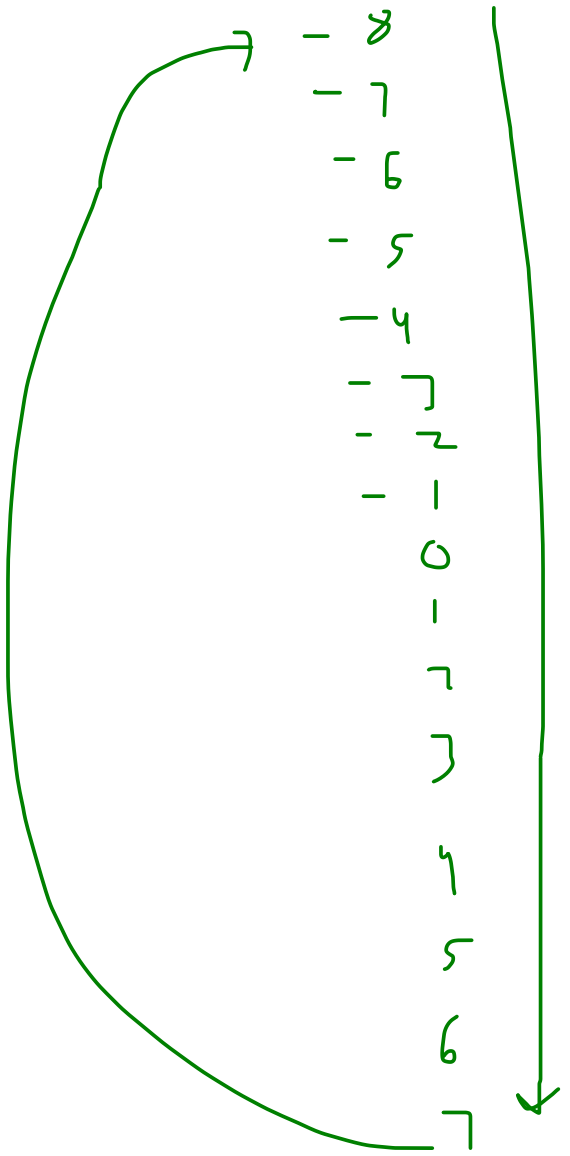
2^4

= 16

B.	D
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	−8
1 0 0 1	−7
1 0 1 0	−6
1 0 1 1	−5
1 1 0 0	−4
1 1 0 1	−3
1 1 1 0	−2
1 1 1 1	−1

+ve

−ve



MSB

0

MSB

1

Correct approach to fin range

1. Convert in decimal
2. Add +ve sign

1. Take 2s compliment of binary number

2. Add -ve sign

nibble → 4 bits → 2^4 → -2^3 to $2^3 - 1$

16 number → -8 to 7

byte → 8 → 2^8 → -2^7 to $2^7 - 1$

short → 16 → 2^{16} → -2^{15} to $2^{15} - 1$

int → 32 → 2^{32} → -2^{31} to $2^{31} - 1$

long → 64 → 2^{64} → -2^{63} to $2^{63} - 1$

Nibble 0 = 5

out of range ?

Nibble 9 = 12

not → 4

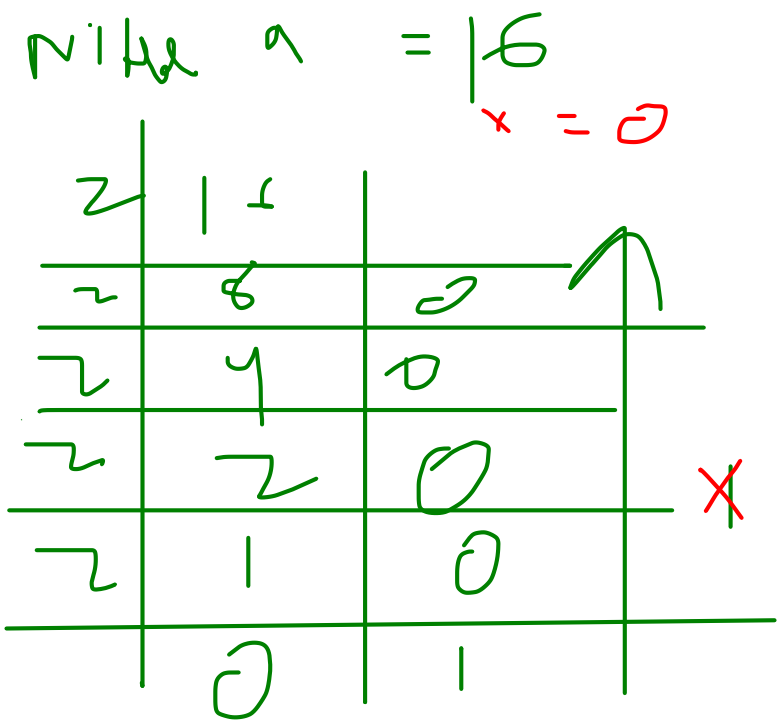
12

2	12	
2	6	0
2	3	0
2	1	1
	0	1

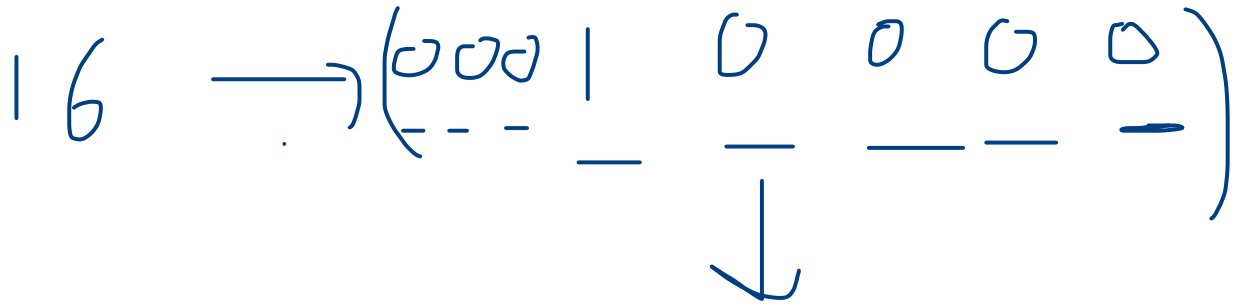
1 1 0 0

1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

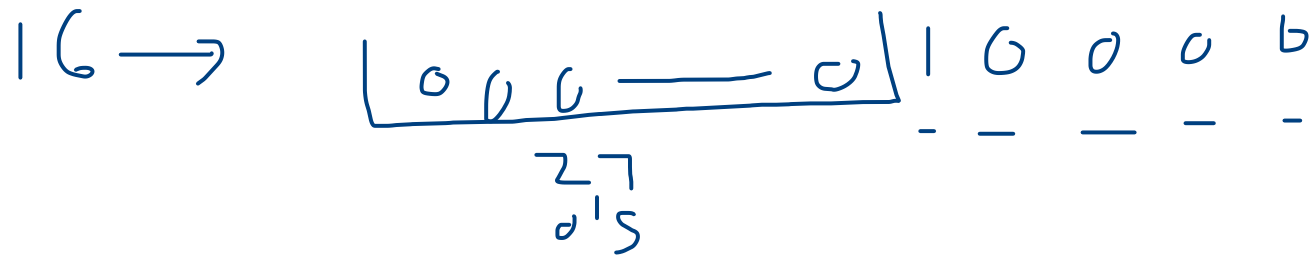
out of range ?



1 0 0 0	-8
1 0 0 1	-7
1 0 1 0	-6
1 0 1 1	-5
1 1 0 0	-4
1 1 0 1	-3
1 1 1 0	-2
1 1 1 1	-1
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7



int



B 2 D

msb
0

1. Convert in decimal
2. Add +ve sign

msb
1

1. Take 2s compliment of binary number
2. Add -ve sign

D 2 B

+ N

1. convert in binary
2. Fit in bits

$(16)_{10} = 10000$

(16) → 0000 a = 0

↓

Nibble

-N

1. Ignore sign
2. Convert in Binary
3. Fit in Bits
4. 2s comp

-7 → 2 | 7

2 | 3 | 1

2 | 1 | 1

0 | 1

0111

↓

1000

0001

(1001) ✓

Basic operators of Bit Manipulation

1. OR --> |

a	b	Res
0	0	0
0	1	1
1	0	1
1	1	1

0 | 0 = 0

1 | 0 = 1

2. AND --> &

0

a	b	Res
0	0	0
0	1	0
1	0	0
1	1	1

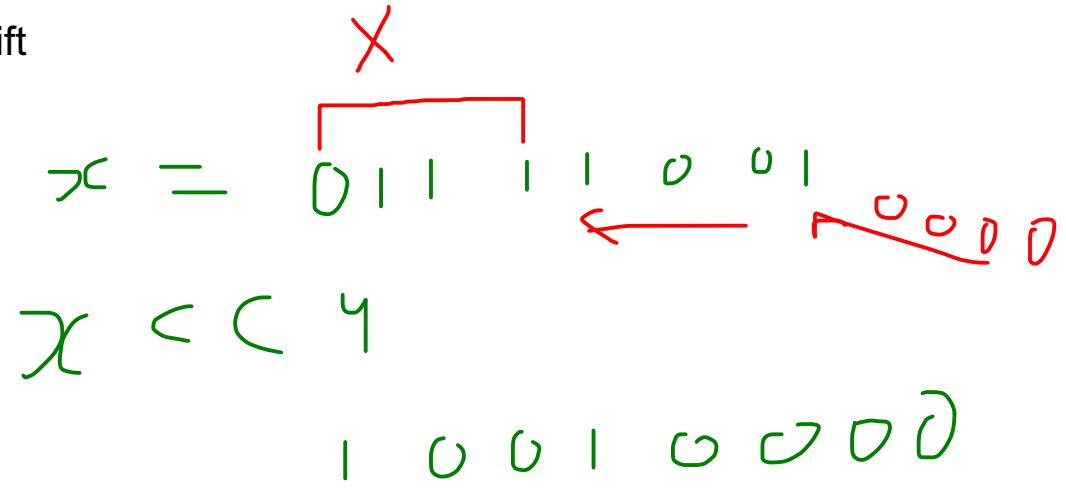
0 & 1 = 0

1 & 1 = 1

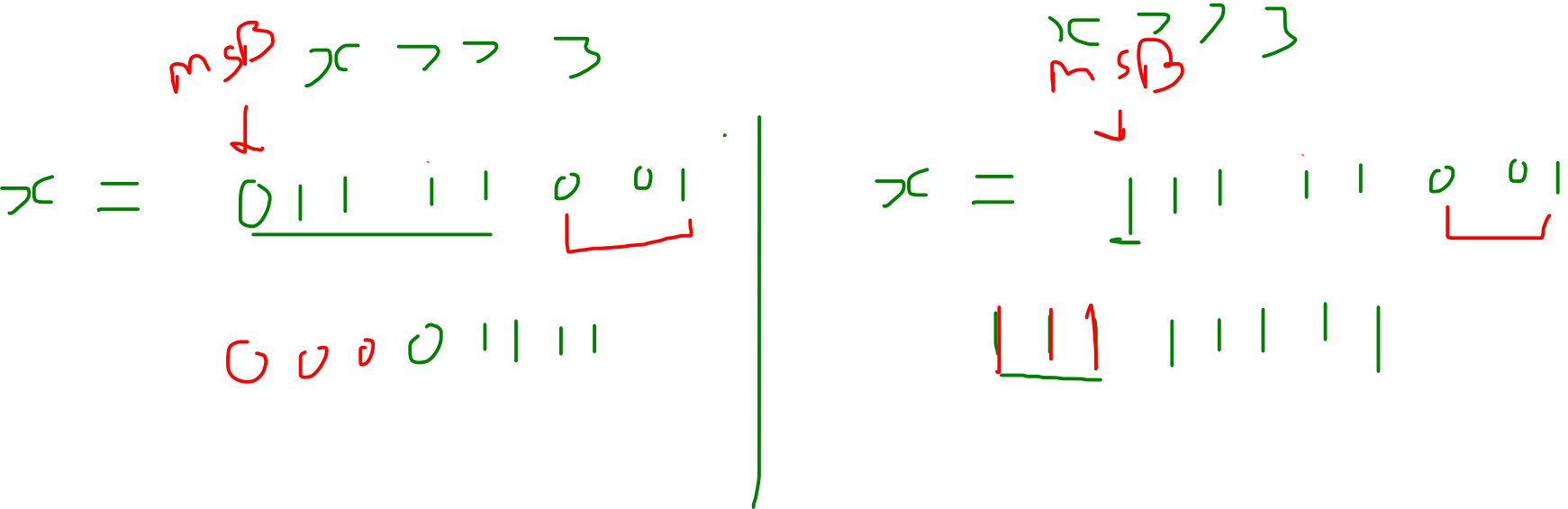
3 XOR

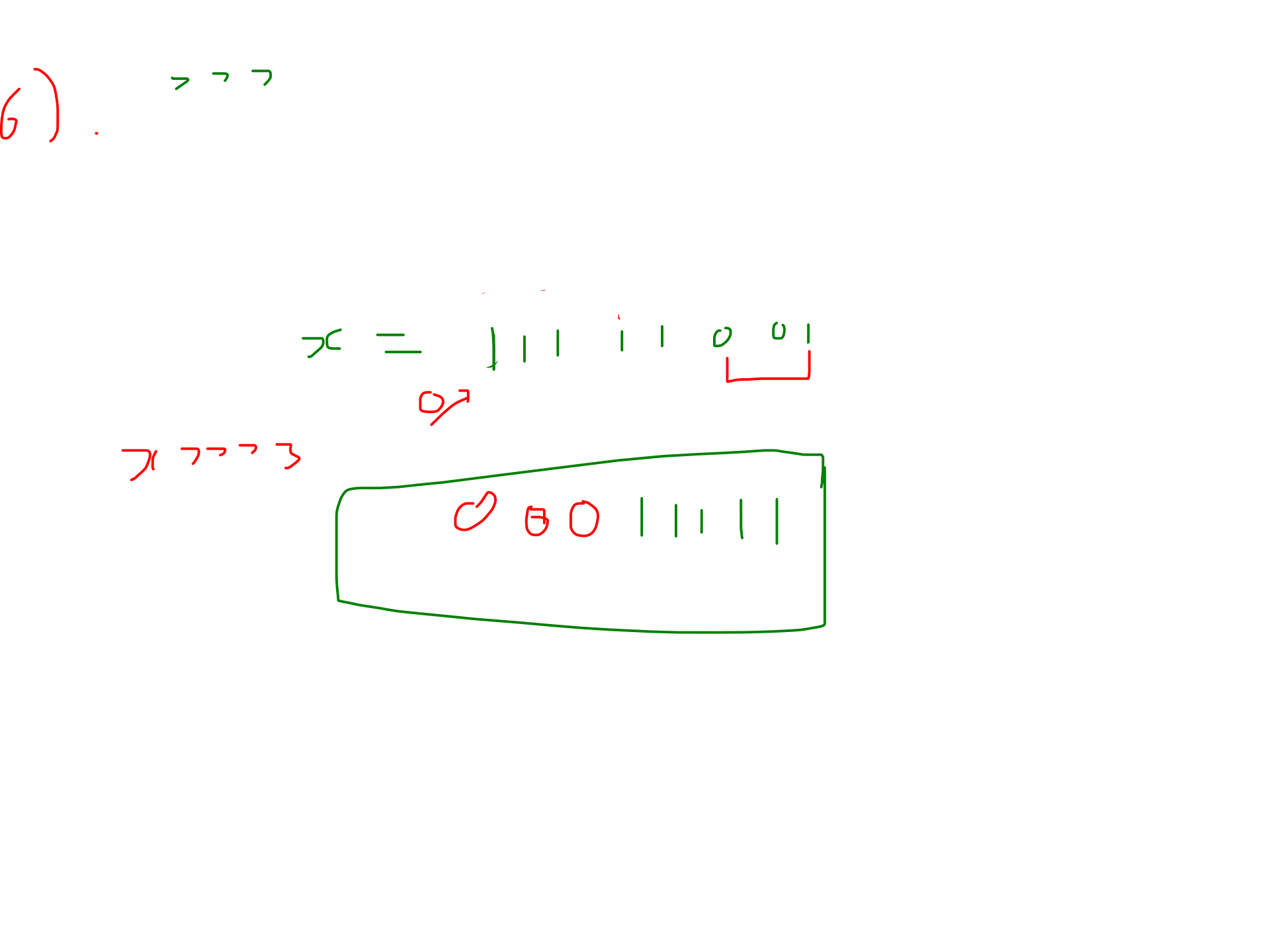
a	b	Res	
0	0	0	a
0	1	1	~a
1	0	1	a
1	1	0	~a

4. << left shift

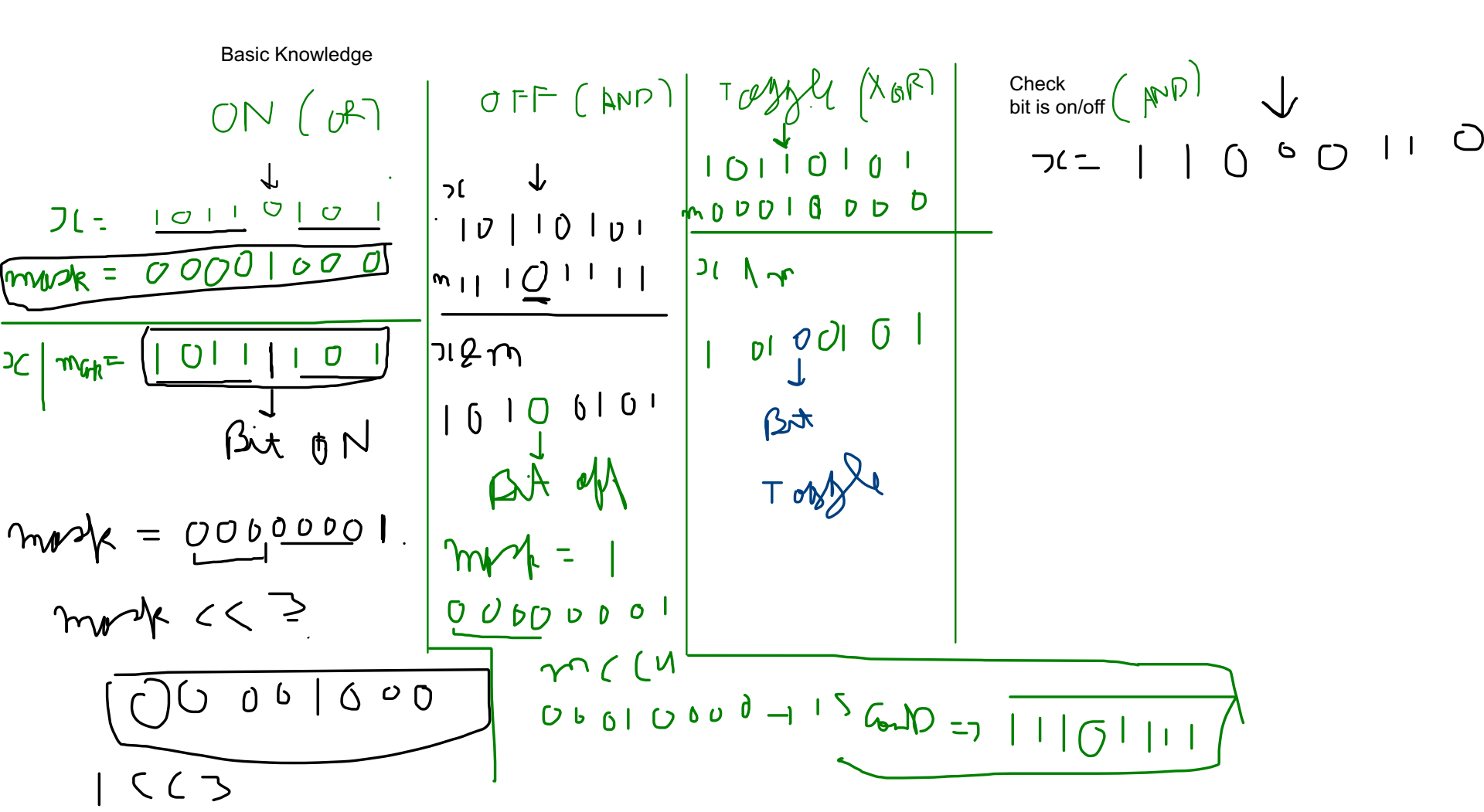


5. >> right shift

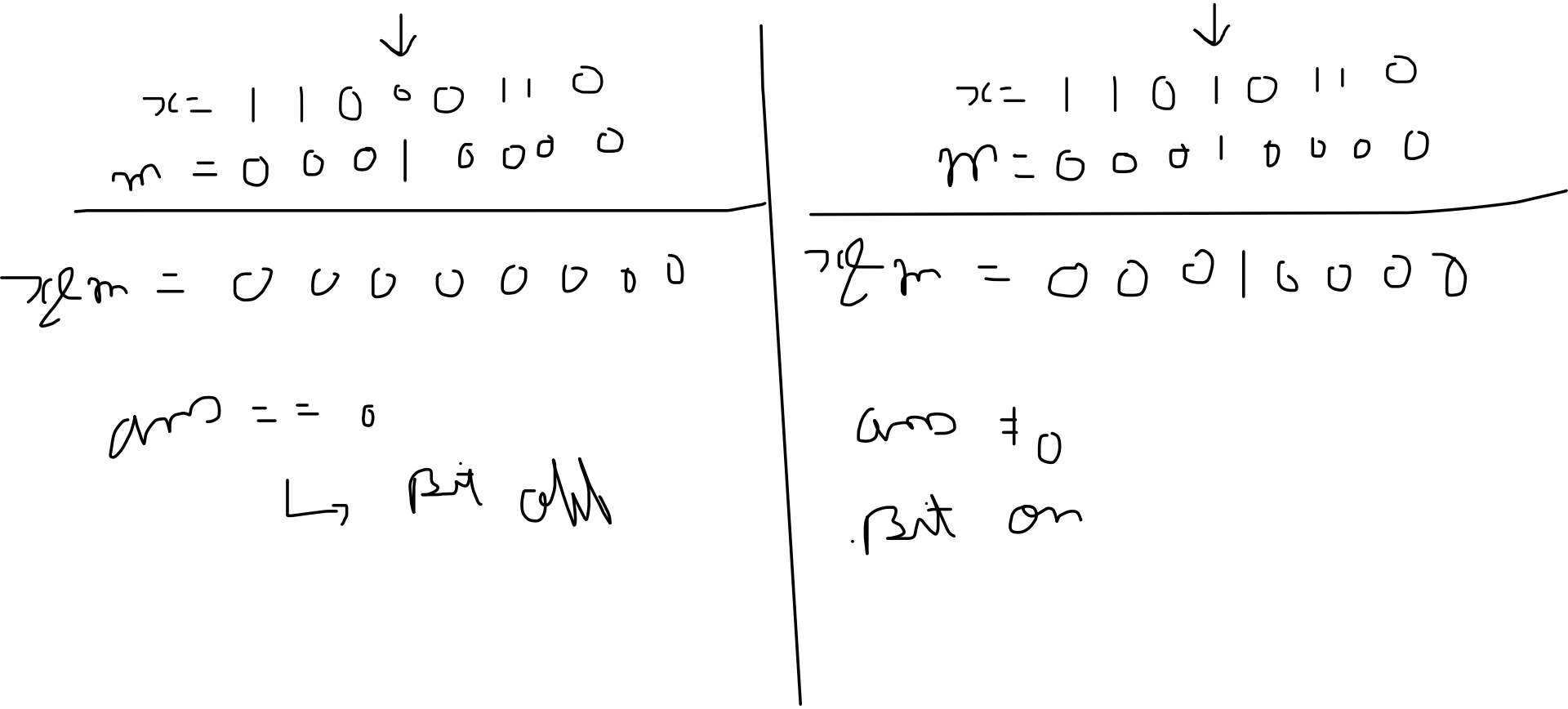




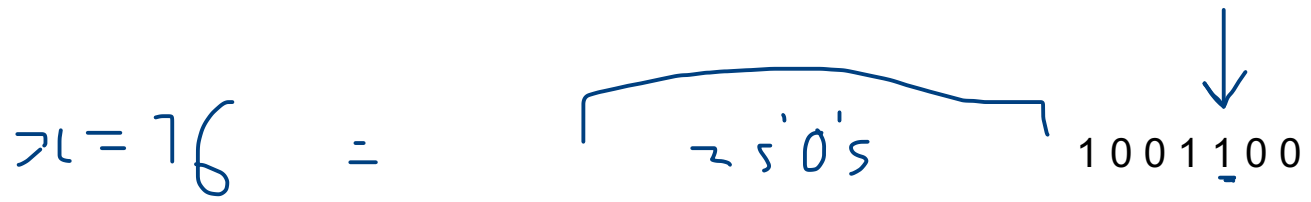
Basic Knowledge



Check
bit is on/off (AND)



Right Set Bit Mask



$$rsbm = x \,|\, n$$

$$= x \,|\, -n$$

x = A 1's & 0's | B 0's

x' = A 0's & 1's 0 B 1's

$x'' = x' + 1$ = A 0's & 1's | B 0's

$x \& x'' =$ A 0's | B 0's

RSBM