

\* Bitwise Operators :  $\rightarrow$

Bitwise  
AND  
OR  
XOR  
Right Shift  
Left Shift  
NOT

Symbols

&  
|  
^  
>>  
<<  
~

Names

No conversions needed  
Ampersand  
Pipe Symbol  
Caret  
Angular Brackets  
Tilde / Negation

operations  
8 1000  
4 0100  
814 1100 = (12)

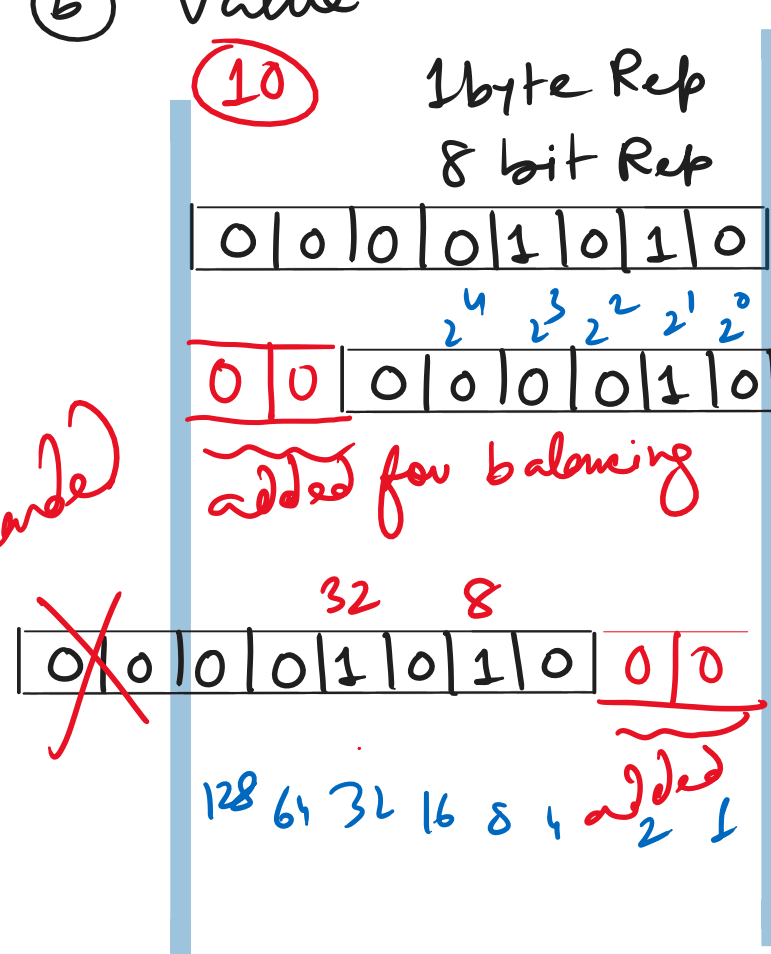
6 0110  
9 1001  
619 1111 = (15)

7 0111  
5 0101  
7 & 5 = (5)  
int int (BIN) int

\* Bitwise Shift Operations :  $\rightarrow$

(a) Unit/Step (b) Value

10 << 2  
40



a = 10

a >> 2 a << 2

{ Bit marking  
or  
Bit manipulation }

a >> 2  
10 >> 2 = 2

Not Truth Table :  $\rightarrow$

T	F
F	T

1	0
0	1

5  $\rightarrow$  0101  
~ 5  $\rightarrow$  1010 = (10)

(PV) (-6)

abs(-6) = 6 = 0110

Bitwise NOT

Double Negation or  
Negation of 2  
or 2's Complement

Formula:

n  $\rightarrow$  ~n = -n - 1

5 = -5 - 1 = -6

(-499) = -(-499) - 1  
= 498

\* Conditional Statements :  $\rightarrow$  (Decision Making)

(i) Simple if  $\rightarrow$  Exactly One Condition

(ii) if else  $\rightarrow$  Exactly Two Conditions

(iii) if else if else ladder  $\rightarrow$  More than 2

(iv) nested if else  $\rightarrow$  Condition Inside Condition

(v) Switch case  $\rightarrow$  More than 4 Conditions to make the code efficient.

(vi) Ternary Operator  $\rightarrow$  Shorthand if else operator.

(Even/Odd) (n % 2 == 0)  $\rightarrow$  Even  
 $\rightarrow$  Odd

2  $\rightarrow$  0010  
3  $\rightarrow$  0011  
4  $\rightarrow$  0100  
5  $\rightarrow$  0101  
6  $\rightarrow$  0110

(n & 1) == 0 even  
1 odd

0100  
0001  
0100  
0001  
0001  $\rightarrow$  odd

\* Power of 2 :  $\rightarrow$

(n & (n-1)) == 0

7 = x  
0111  
& 0110  
0110

6 = x  
0110  
& 0101  
0100

4 = 2<sup>2</sup>  
0100  
0011  
0000

2 = 2<sup>1</sup>  
0010  
0001  
0000

8 = 2<sup>3</sup>  
1000  
0111  
0000