Week 2

Hari Sethuraman

By the end of Today

You should have a solid understanding of

- How computers store information
- The memory
- Datatypes

How do we write numbers?

- We write numbers in a form called ‘base ten’

2 1 3

‘Hundreds’ place = 10^2

‘Tens’ place = 10^1

‘Ones’ place = 10^0

Examples

- 5 -> (5×10^0)
- 65 -> $(6 \times 10^1) + (5 \times 10^0)$
- 125 -> $(1 \times 10^2) + (2 \times 10^1) + (5 \times 10^0)$

Click to add text

- 213 -> $(2 \times 10^2) + (1 \times 10^1) + (3 \times 10^0)$

Questions:

- 159?
- 200?

Answers

- 159 -> $(1 \times 10^2) + (5 \times 10^1) + (9 \times 10^0)$
- 200 -> $(2 \times 10^2) + (0 \times 10^1) + (0 \times 10^0)$

How do computers ‘write’ numbers?

- Computers use something called ‘Base two’
- Instead of each digit being a power of ten
- Computers use Powers of two
  - Each digit can only have two values: 0, 1
  - This is because transistors in computers can only be
    - ‘Off’ = 0
    - Or ‘On’ = 1

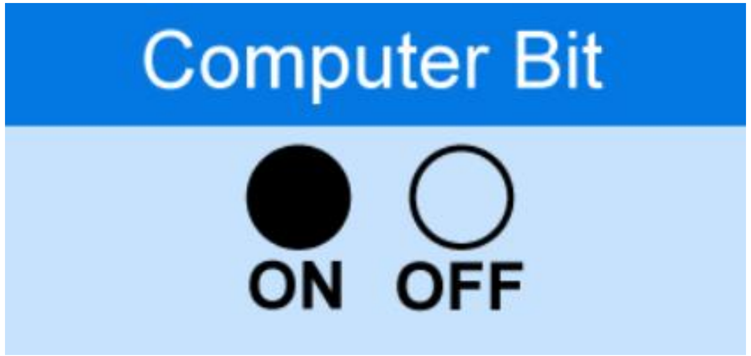
Continuation

1 0 1

‘Fours’ place = 2^2

‘Twos’ place = 2^1

‘Ones’ place = 2^0



Examples

- 1 -> (1×2^0)
    - 1
- 1 0 -> $(1 \times 2^1) + (0 \times 2^0)$
  - 2 + 0
    - 2
- 1 1 0 -> $(1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$
  - 4 + 1 + 0
    - 5
- 1 1 0 1 1 -> $(1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
  - 16 + 8 + 0 + 2 + 1
    - 27

Problems

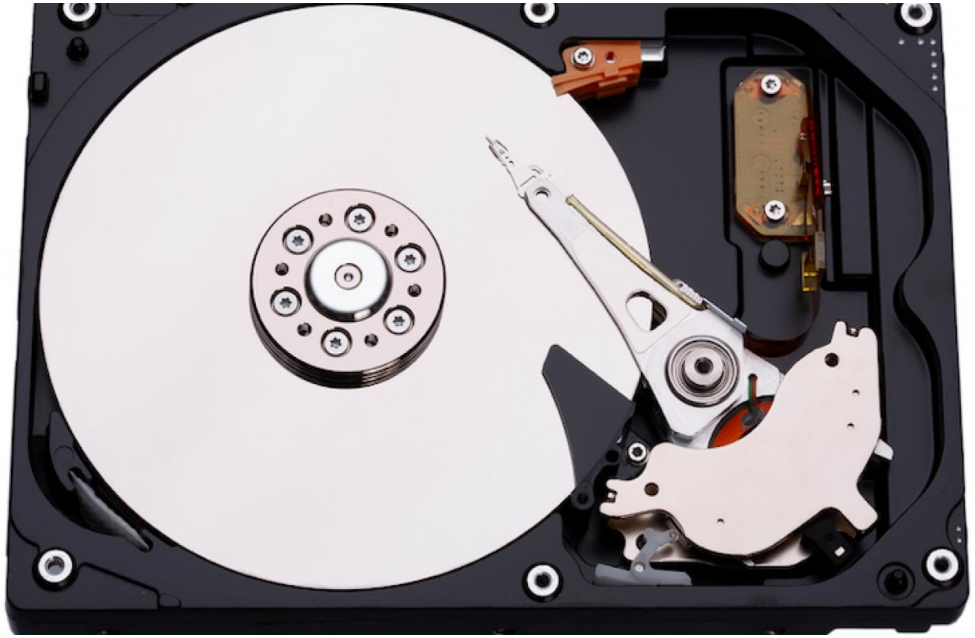
- 1 1 1 1 -> ?
- 1 0 1 0 1 -> ?
- 14 -> 8 + 4 + 2 + 0?
- 8 -> ?

Answers

- 1 1 1 1 -> $(1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
  - 8 + 4 + 2 + 1
    - 15
- 1 0 1 0 1 -> $(1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
  - 16 + 0 + 4 + 0 + 1
    - 21
- 14 -> 8 + 4 + 2 + 0
  - $(1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$
    - 1 1 1 0
- 8 -> 8 + 0 + 0 + 0
  - $(1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$
    - 1 0 0 0

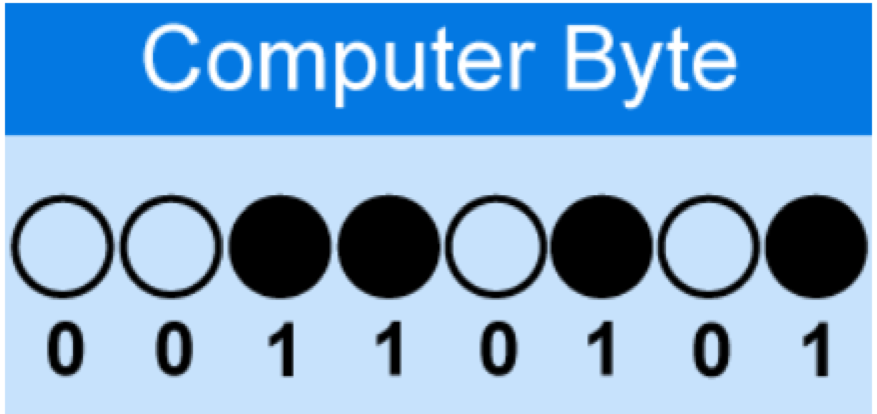
Bits and Bytes

- Each ‘0’ or ‘1’ is equal to **one Bit**
  - 1 1 1 1 -> Four Bits
  - 1 0 1 0 1 -> Five Bits
- Computers group Bits into **Bytes**
  - Eight Bits make **One Byte**
    - 1024 Bytes Make one Kilobyte
    - 1024 Kilobytes make one Megabyte
    - 1024 Megabytes make one Gigabyte
    - …



What is the maximum Value you can store in a Byte?

- (A Byte is 8 bits) -> 1 1 1 1 1 1 1 1 -> 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

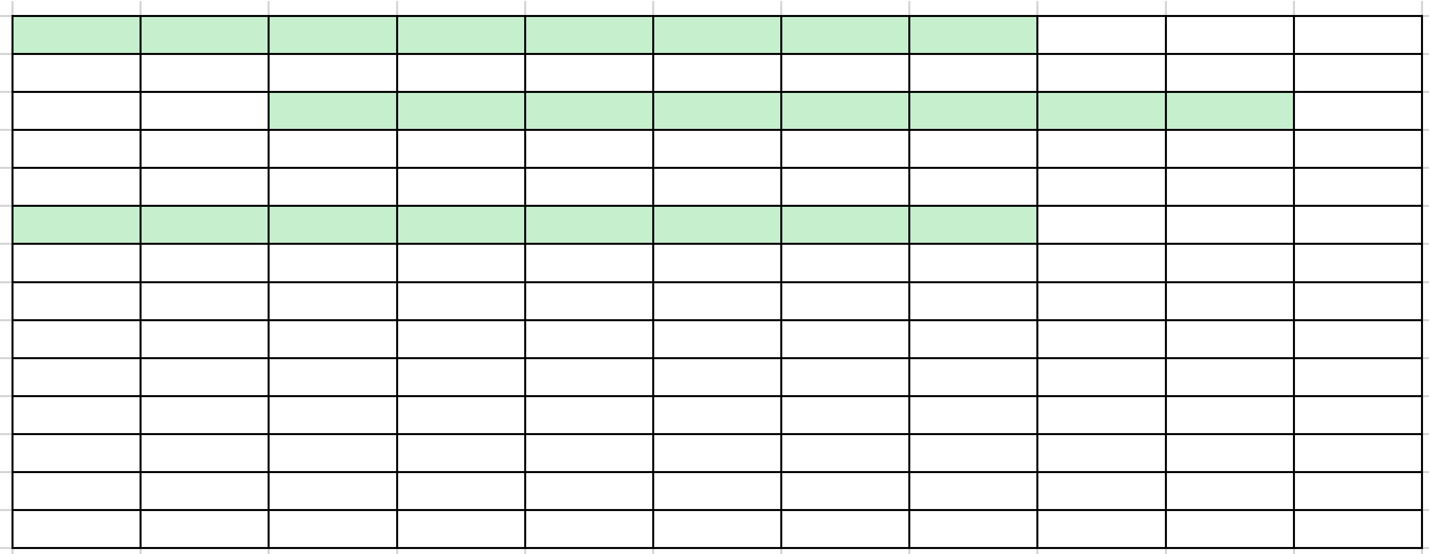


Answer

- The maximum number you can represent is 255
- Why?
- 1 1 1 1 1 1 1 1 ->
  - $(1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
  - 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255
- This means you can represent 256 different values using just one Byte!
  - (256 because you include Zero 0 -> 0 0 0 0 0 0 0 0)

Memory

- The memory is a very big grid
- Each Cell is one bit.
- Each green Group is one Byte



Data types 1

- Char
  - Character: there are 256 different Characters. Represented using ‘ASCII’

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0			NULL	32			space	64			@	96			`
1			SOH	33			!	65			A	97			a
2			STX	34			"	66			B	98			b
3			ETX	35			#	67			C	99			c
4			EOT	36			$	68			D	100			d
5			ENQ	37			%	69			E	101			e
6			ACK	38			&	70			F	102			f
7			BEL	39			'	71			G	103			g
8			BS	40			(	72			H	104			h
9			TAB	41			)	73			I	105			i
10			LF	42			*	74			J	106			j
11			VT	43			+	75			K	107			k
12			FF	44			,	76			L	108			l
13			CR	45			-	77			M	109			m
14			SO	46			.	78			N	110			n
15			SI	47			/	79			O	111			o
16			DLE	48			0	80			P	112			p
17			DC1	49			1	81			Q	113			q
18			DC2	50			2	82			R	114			r
19			DC3	51			3	83			S	115			s
20			DC4	52			4	84			T	116			t
21			NAK	53			5	85			U	117			u
22			SYN	54			6	86			V	118			v
23			ETB	55			7	87			W	119			w
24			CAN	56			8	88			X	120			x
25			EM	57			9	89			Y	121			y
26			SUB	58			:	90			Z	122			z
27			ESC	59			;	91			[	123			{
28			FS	60			<	92			\	124			|
29			GS	61			=	93			]	125			}
30			RS	62			>	94			^	126			~
31			US	63			?	95			_	127			DEL

‘A’ -> 65 -> 0 1 0 0 0 0 0 1

‘!’ -> 33 -> 0 0 1 0 0 0 0 1

‘B’ -> ?

‘$’ ->

Answers

- ‘B’ -> 66 -> 0 1 0 0 0 0 1 0
- ‘$’ -> 36 -> 0 0 1 0 0 1 0 0

Data types 2

- Int
  - Integer: Any while number between -32,768 and 32,767
  - Uses two Bytes or 16 bits of memory
  - Integers are numbers you can add, divide, subtract and do other mathematical notations.
- Bool
  - Boolean: Can have two values: True, False
  - Uses 1 byte or 8 bits of memory
  - Why not 1 bit? Because the computer groups memory in bytes, we can’t use an individual bit. So, we use a full byte.

Data types 3

- Float
  - Decimals: up to 4 numbers after decimal point
  - 4 bytes usually

- Double
  - Decimal: up to **8** numbers after decimal point
  - 8 bytes usually