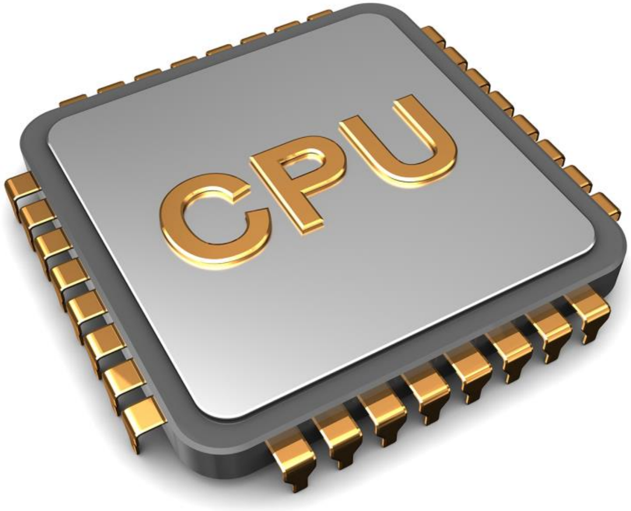
NUMBERS AND INSTRUCTIONS

By Olmo Gordon and Dario von Muenchhausen

INSTRUCTION SET OF A CPU

- Each processor type has a limit of how many instructions it can carry out at once (instruction set)
- Instructions are:
  - Changing data
  - Moving data
  - Controlling data flow

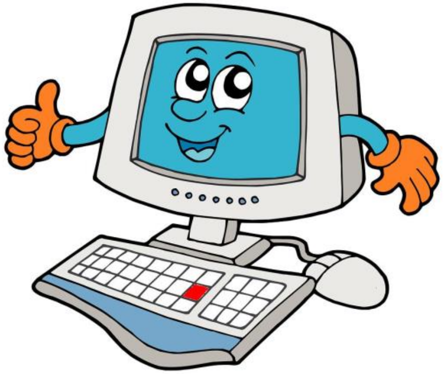


TYPES OF INSTRUCTIONS

Arithmetic operations	Logical operations
Plus	Comparisons
Minus	AND, OR, NOT (bitwise operators)
Multiply	They Change the bits in a byte
Divide	

PROCESS OF INSTRUCTIONS

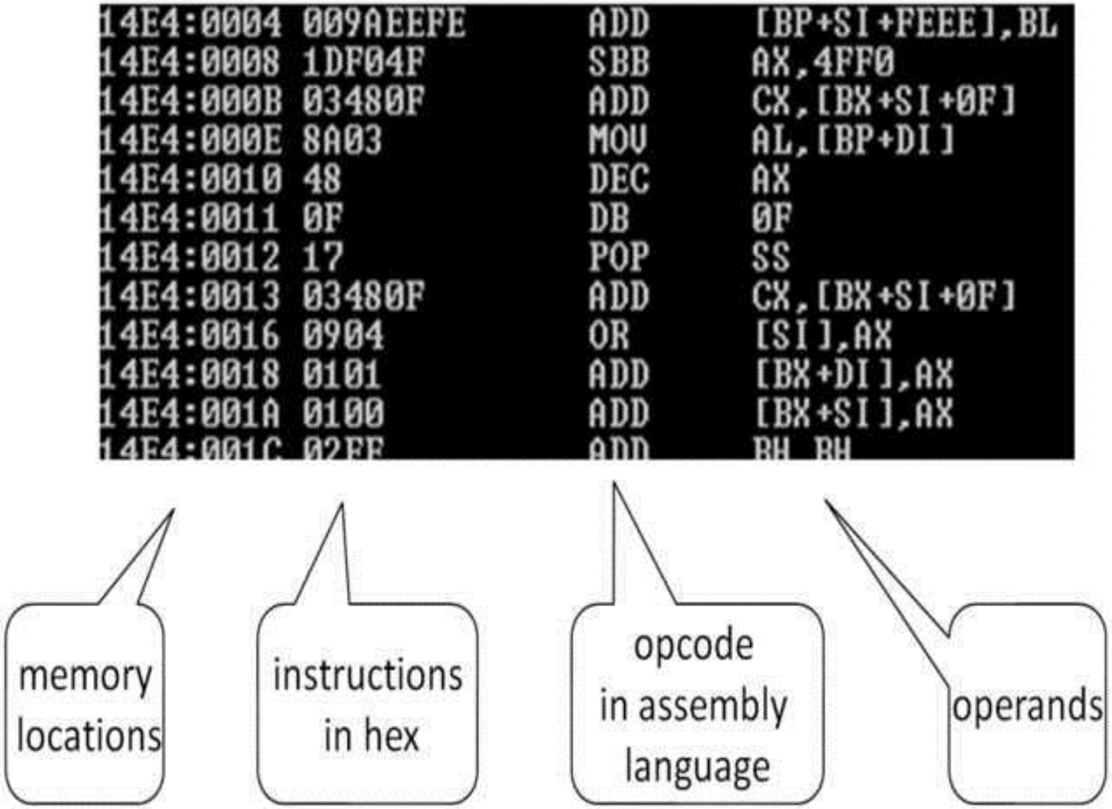
- Computer gets instructed to run a program
- Directed to the start address of the data and instructions
- CPU fetches the first instruction from the start location
- Then it decodes it to find out what to do next



OPERATOR AND OPERAND

Operator	Operand
Instruction part	Data part
ADD to the accumulator	The contents of memory location
First 4 numbers	Last 4 numbers

HOW INSTRUCTIONS ARE SET



```
14E4:0004 009AEEFE      ADD     [BP+SI+FEEE],BL
14E4:0008 1DF04F        SBB     AX,4FF0
14E4:000B 03480F        ADD     CX,[BX+SI+0F]
14E4:000E 8A03          MOV     AL,[BP+DI]
14E4:0010 48            DEC     AX
14E4:0011 0F            DB      0F
14E4:0012 17            POP     SS
14E4:0013 03480F        ADD     CX,[BX+SI+0F]
14E4:0016 0904          OR      [SI],AX
14E4:0018 0101          ADD     [BX+DI],AX
14E4:001A 0100          ADD     [BX+SI],AX
14E4:001C 02FF          ADD     BH,BH
```

memory locations

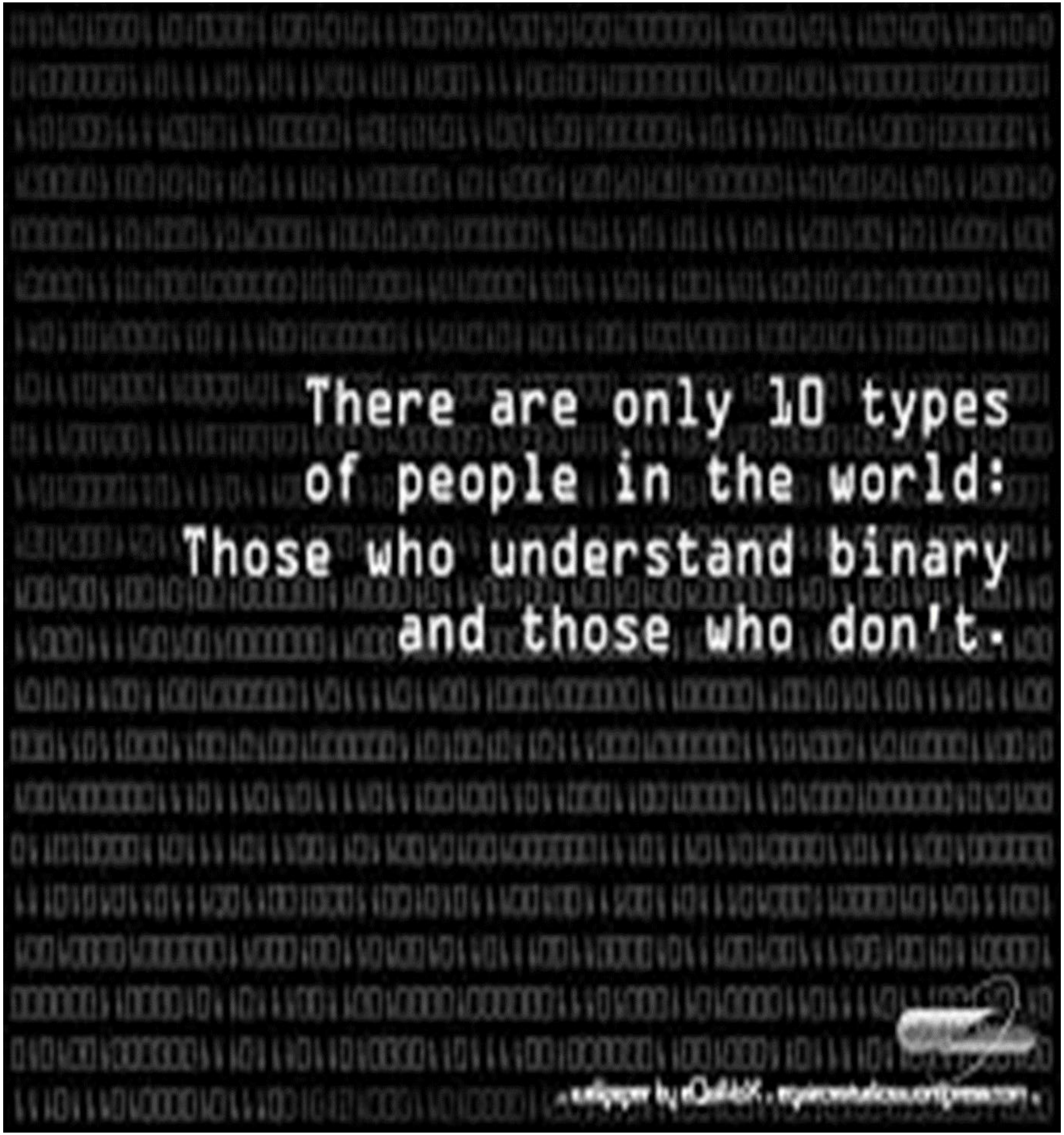
instructions in hex

opcode in assembly language

operands

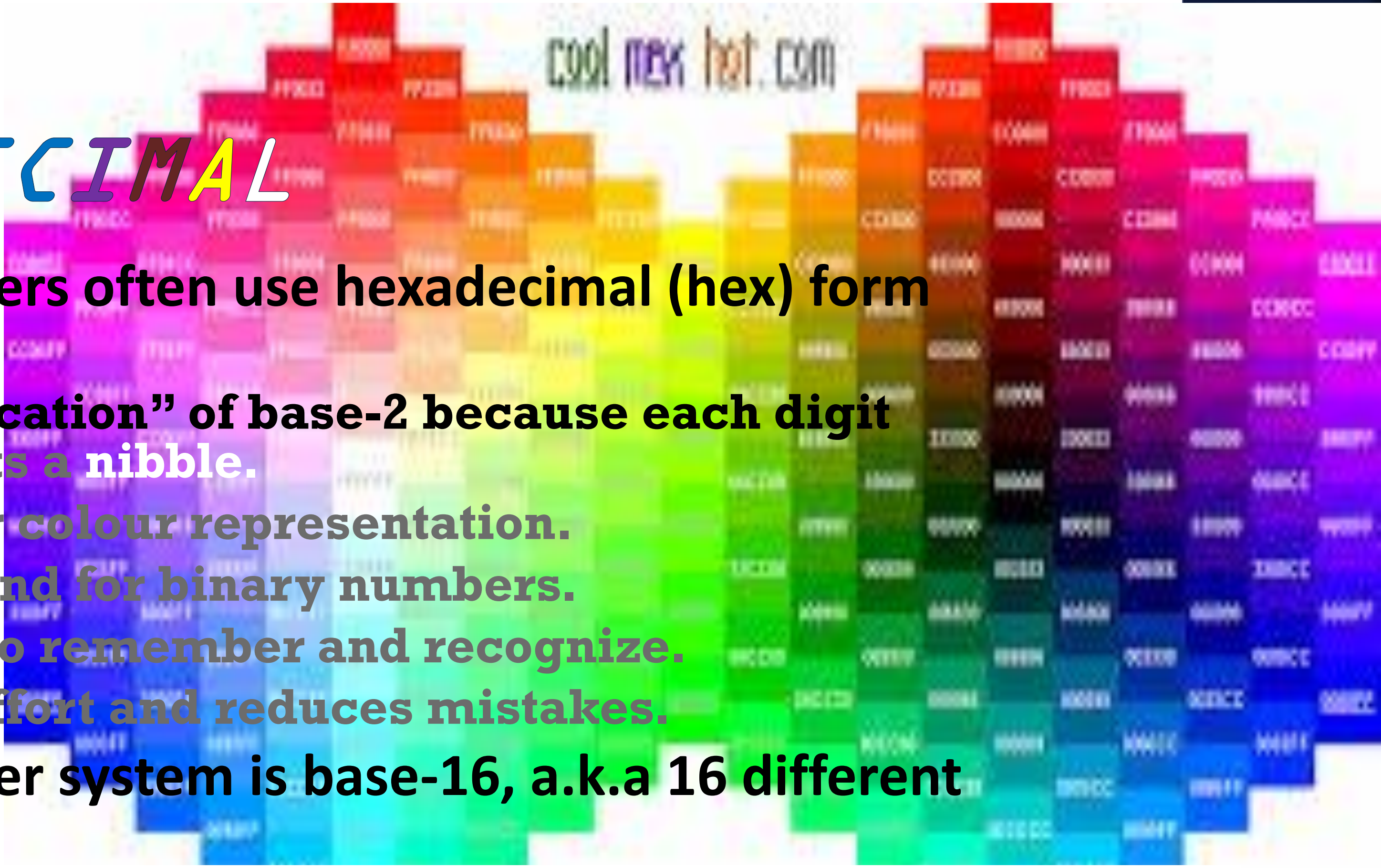
BINARY

- The binary system (a.k.a ***base-2*** system) uses a ***bit*** (most basic unit of information in a computer) that:
  - Can either be **"on" (1)** or **"off" (0)**.
  - Represent **high** or **low** voltage in a circuit.
  - Form ***bytes*** (group of eight bits, formed by two **nibbles**). It is the smallest possible *addressable* unit of computer storage.
- A ***word*** is a group of bytes.
  - Words size for a processor is a multiple of bytes (commonly 16, 32, or 64 bits).



There are only 10 types of people in the world: Those who understand binary and those who don't.

HEXADECIMAL



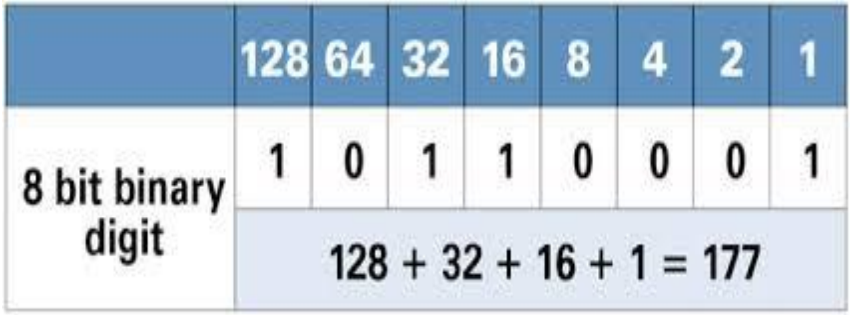
- Programmers often use hexadecimal (hex) form because:
  - "Simplification" of base-2 because each digit represents a nibble.
  - Useful for colour representation.
    - Shorthand for binary numbers.
    - Easier to remember and recognize.
    - Saves effort and reduces mistakes.
- This number system is base-16, a.k.a 16 different digits:

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- Each column is worth 16 times the one on its right

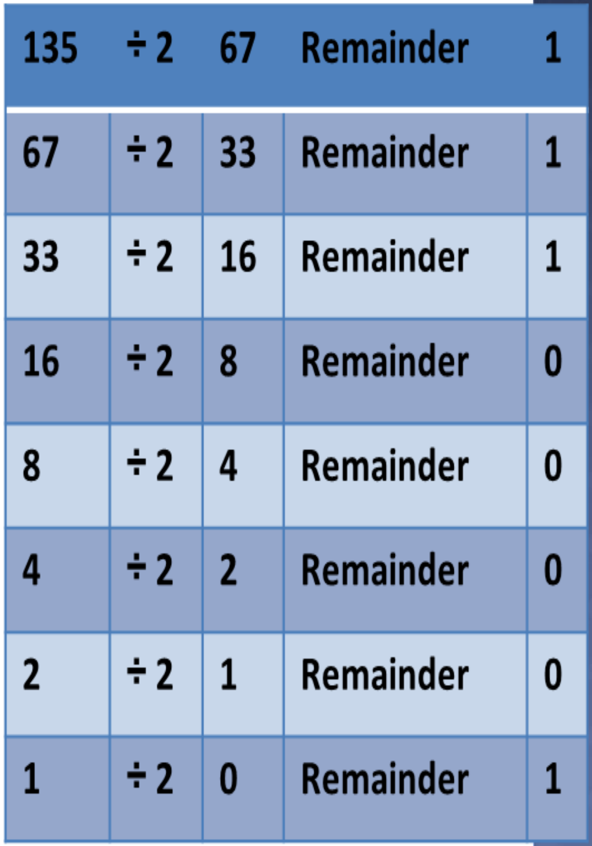
256	16	1

CONVERSION

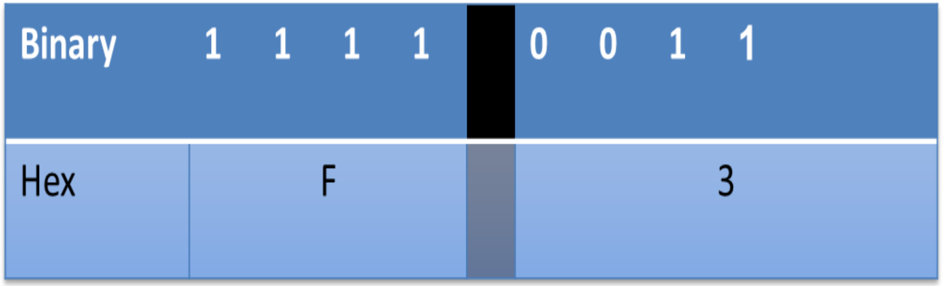


	128	64	32	16	8	4	2	1
8 bit binary digit	1	0	1	1	0	0	0	1
	128 + 32 + 16 + 1 = 177							

- Convert and interchange between al three systems.
- Different methods:
  - Repeated division.
  - Multiplication and addition.
  - Nibble trick.



135	÷2	67	Remainder	1
67	÷2	33	Remainder	1
33	÷2	16	Remainder	1
16	÷2	8	Remainder	0
8	÷2	4	Remainder	0
4	÷2	2	Remainder	0
2	÷2	1	Remainder	0
1	÷2	0	Remainder	1



Binary	1	1	1	1		0	0	1	1
Hex	F					3			

Thank you for your attention!