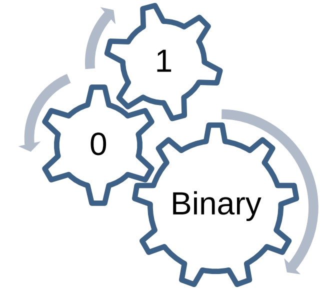
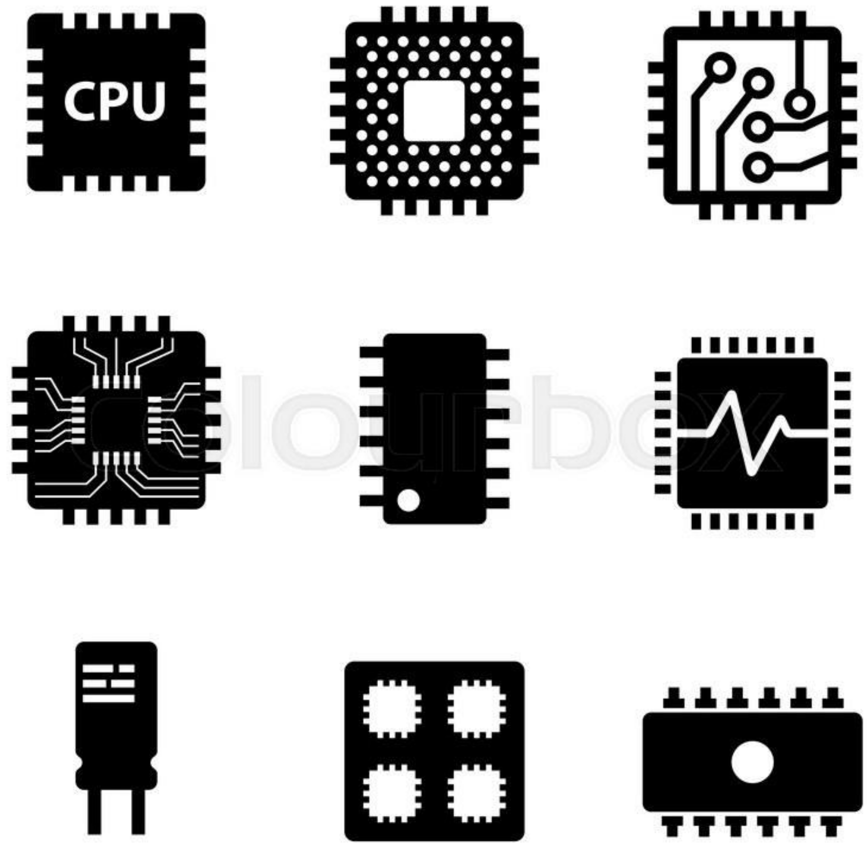


«Computer Systems»

CSIT502

The course is about computer hardware …



… and basic hardware-based computer software (assembly language)

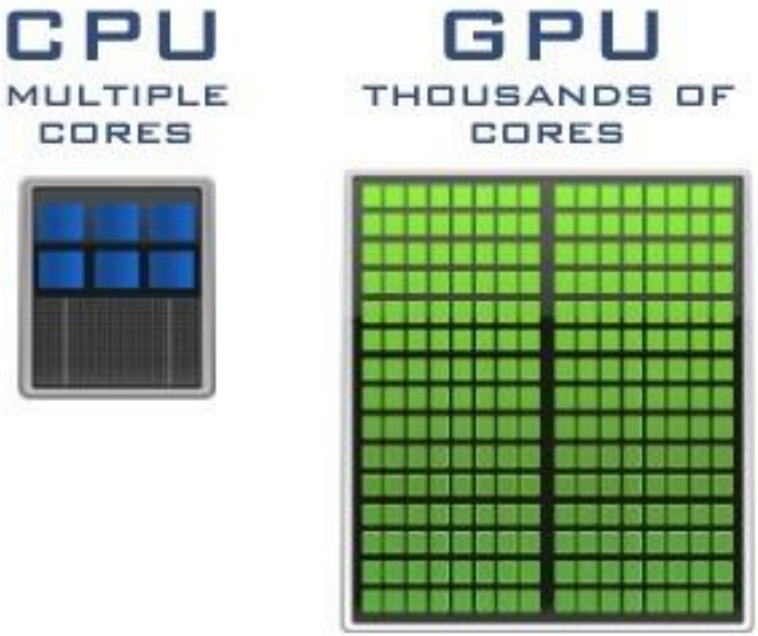
[CPU][DSP][GPU][MEM]…[…]

- Computers consist of many units …

CPU: Central Processing Unit
DSP: Digital Signal Processing
GPU: Graphics Processing Unit
MEM: Memory

https://www.qualcomm.com/

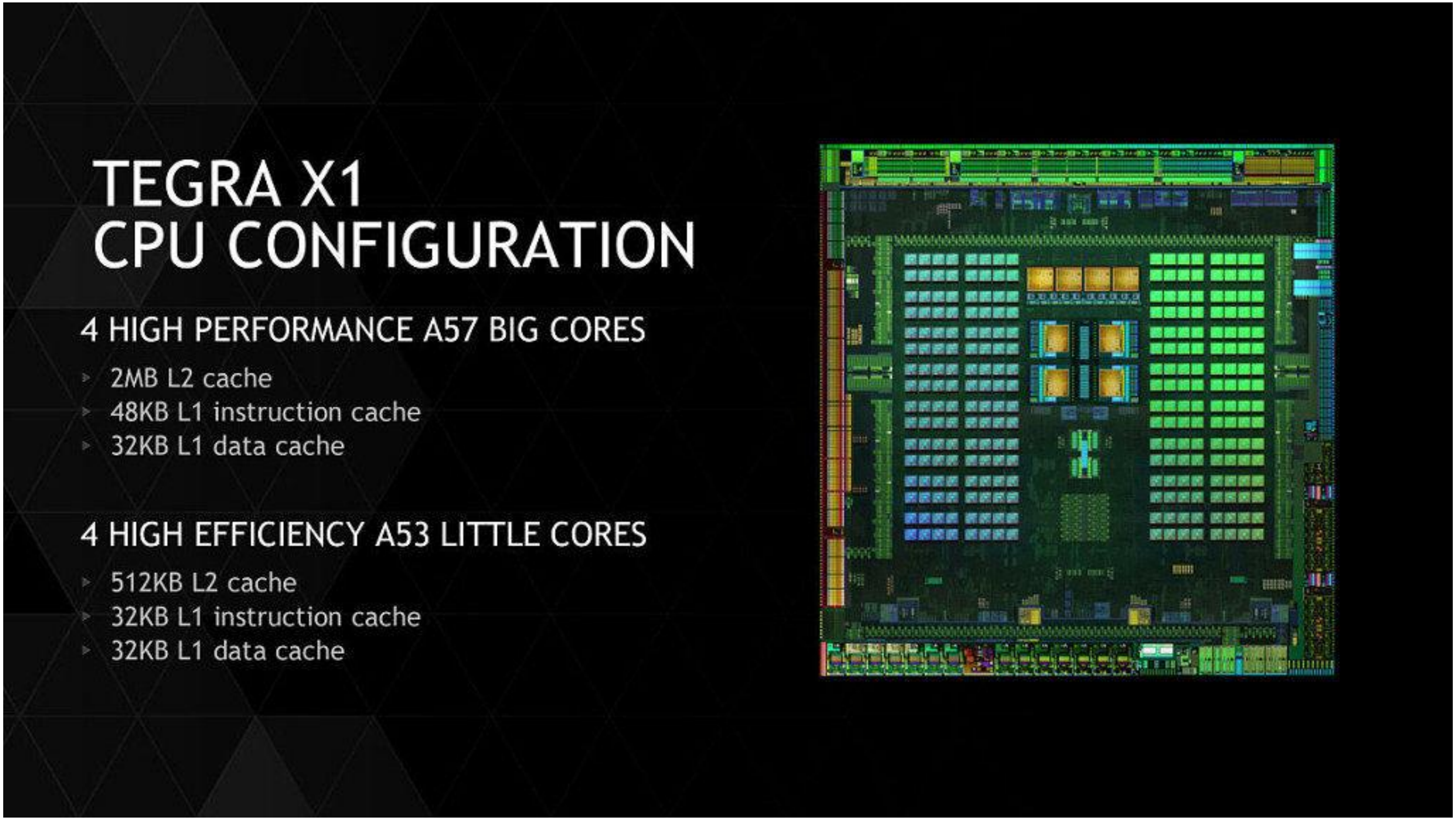
CPU vs GPU



TEGRA X1 PROCESSOR SPECIFICATIONS

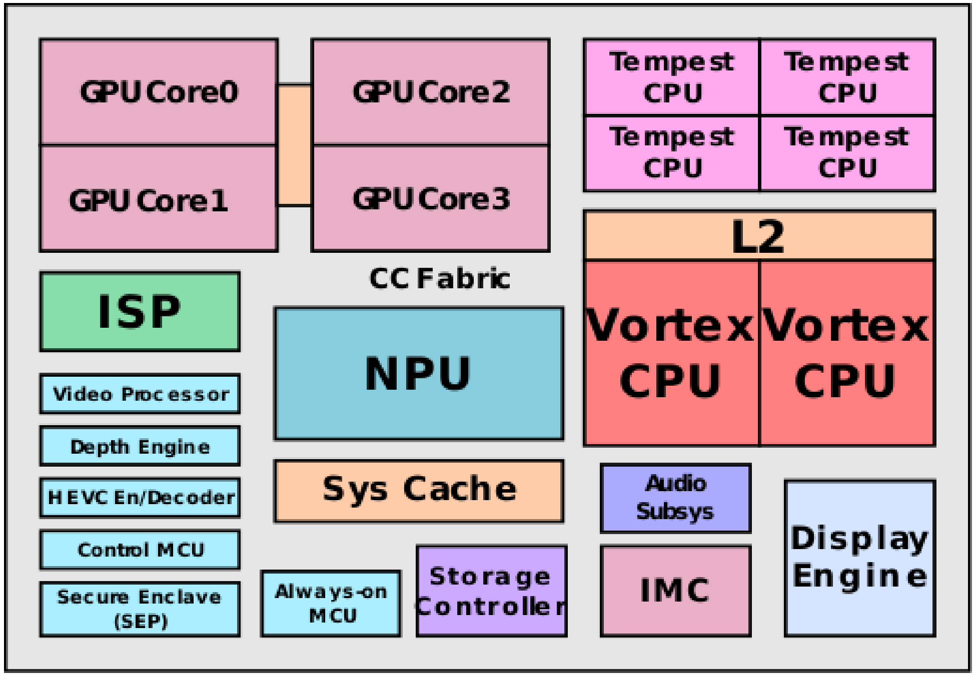
	TEGRA X1
GPU	NVIDIA Maxwell 256-core GPU DX-12, OpenGL 4.5, NVIDIA CUDA®, OpenGL ES 3.1, AEP, and Vulkan
CPU	4 CPU-cores, 64-bit ARM® CPU 4x A57 2MB L2

TEGRA X1


TEGRA X1
CPU CONFIGURATION
4 HIGH PERFORMANCE A57 BIG CORES
2MB L2 cache
48KB L1 instruction cache
32KB L1 data cache
4 HIGH EFFICIENCY A53 LITTLE CORES
512KB L2 cache
32KB L1 instruction cache
32KB L1 data cache

Android: “Snapdragon 865” SoC (2020)

iPhone: RISC ARMv8-A (XS, XS Max, XR)



CMOS Process Technology: **7nm**

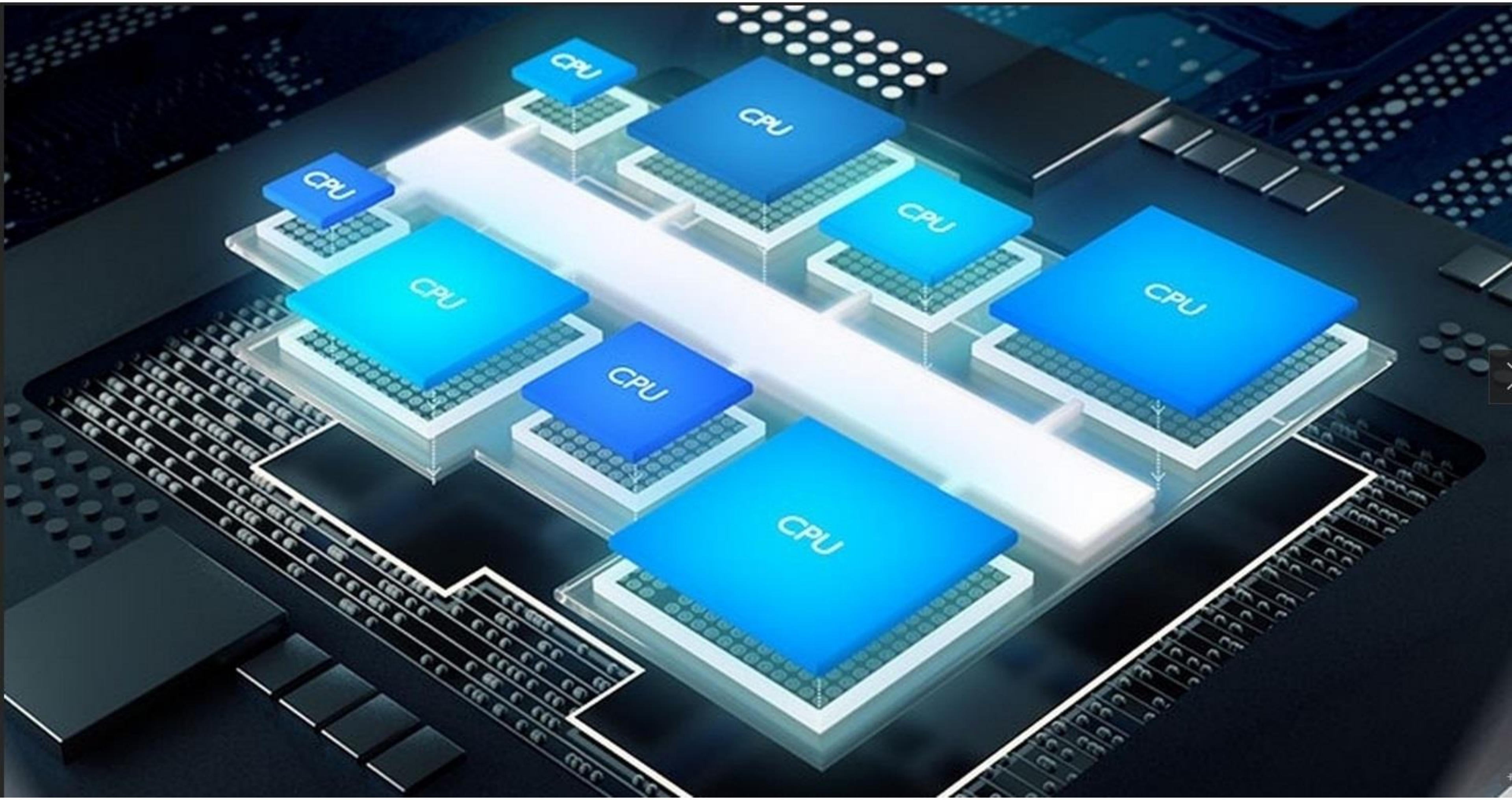
The Apple A12 is a 64-bit system-on-chip (SoC), designed by Apple Inc.

iPhone 11: RISC ARMv8.5-A

Hexa-core (2 high performance Lightning + 4 high efficiency Thunder)

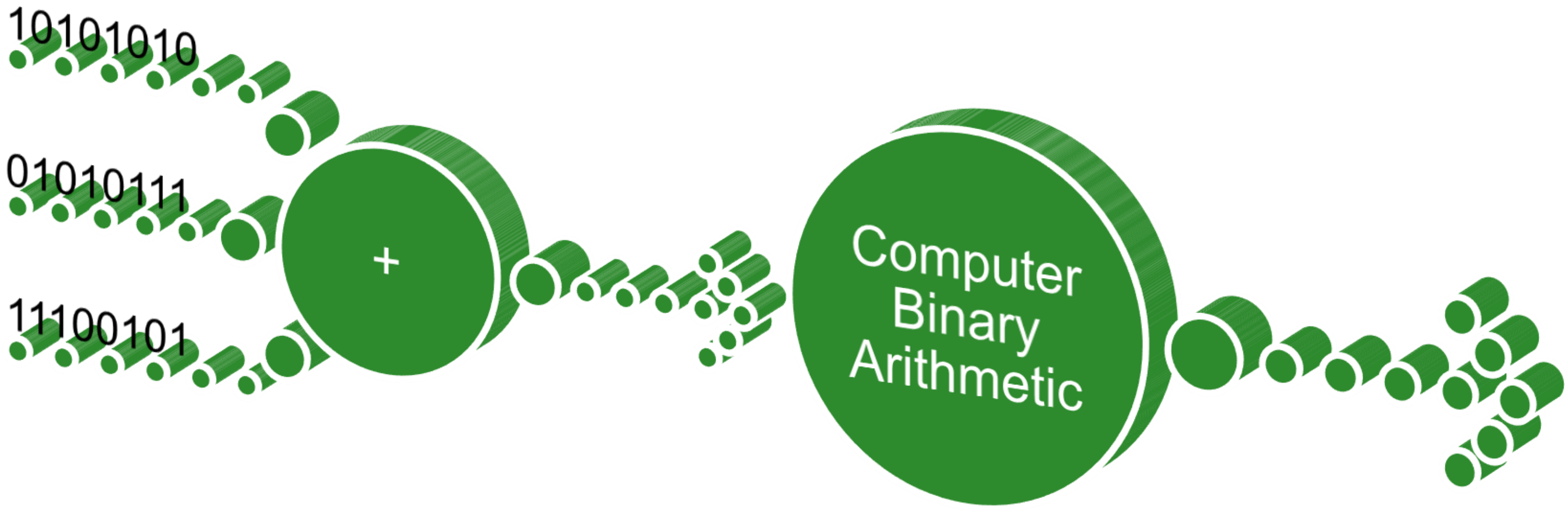
CMOS Process Technology: **7nm**

The Apple A13 is a 64-bit system-on-chip (SoC), designed by Apple Inc.

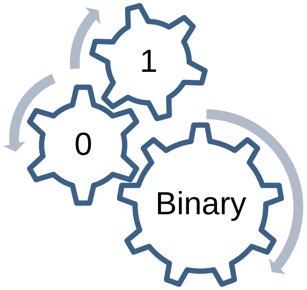


Octa-Core CPU

«Computer Systems», use binary arithmetic



… and **Bits**



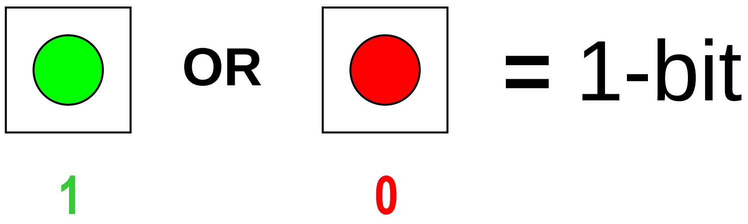
Bit ...

Bit

- Binary digit (Bit)
- Binary system of arithmetic
- digit (unit) … “The digit or finger is an ancient and obsolete non-SI unit of measurement of length. It was originally based on the breadth of a human finger. It was a fundamental unit of length in the Ancient Egyptian, Mesopotamian, Hebrew, Ancient Greek and Roman systems of measurement”. (Wikipedia)

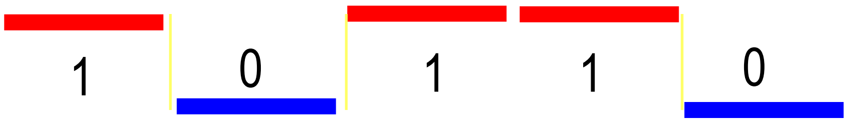
Digital computers use two logic states

- 0 and 1 = Binary System
- Bi in Latin is 2



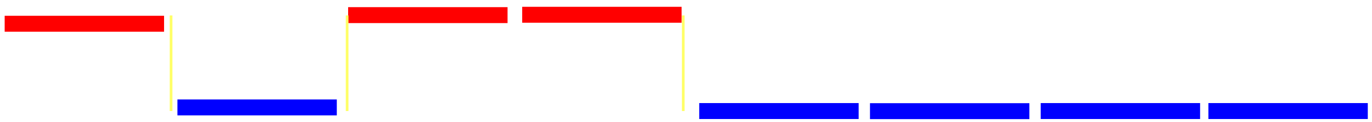
Binary system

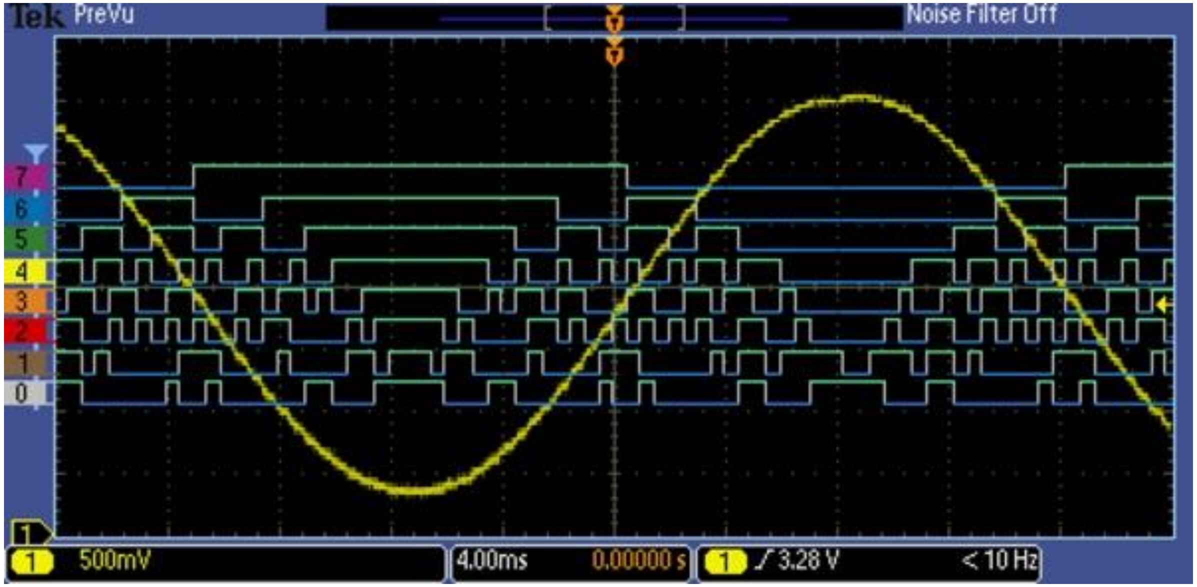
- 0 and 1 = Binary System
- 0 = OFF, 1 = ON
- 0 = Down, 1 = Up



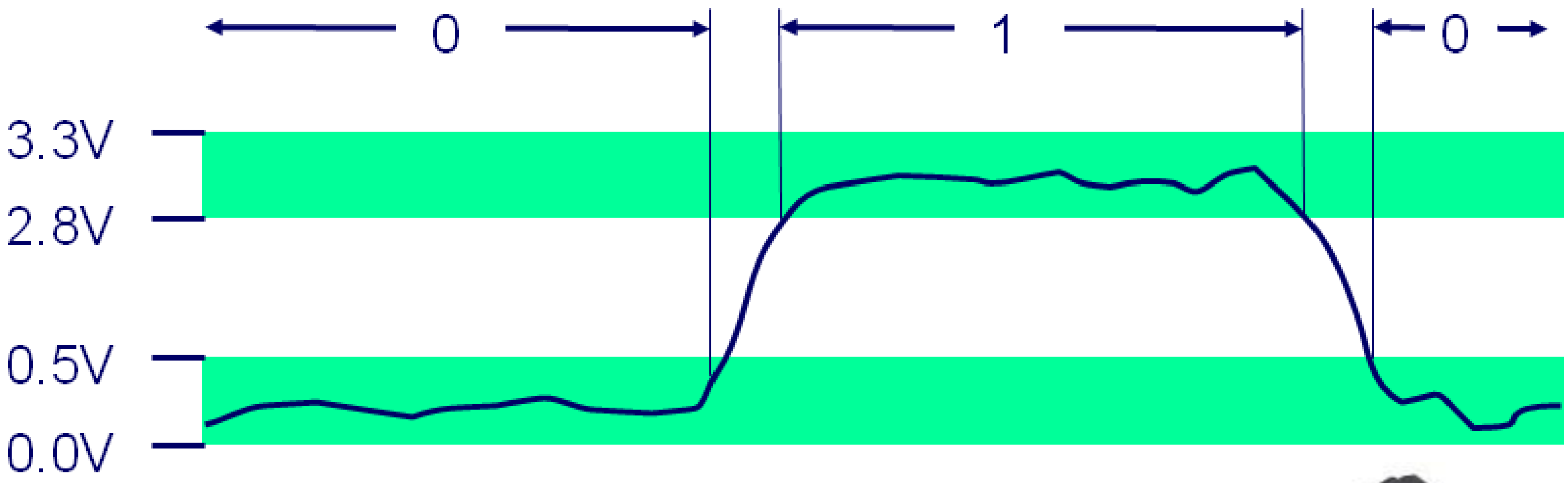
- ✓ 0 = (0.0-0.5 volts)
- ✓ 1 = (2.8-3.7 volts)

Actual binary signal or sequence





It is all voltage … electricity



ON=1 and OFF=0

ON-OFF: Two logic states like a home two position electrical light switch

Digital (Two logic states: ON and OFF)

Switch

Binary

Digital (Binary) & Analog

Binary Analog

Digital Computer Arithmetic

Arithmetic-Number Systems

- (10) Decimal: 0,1,2,3,4,5,6,7,8,9.
- (2) Binary
- (8) Octal
- (16) Hexadecimal

Computer used arithmetic systems

- Binary numbers: $base_2$

  [0, 1]

- Octal numbers: $base_8$

  [0 1 2 3 4 5 6 7]

- Hexadecimal numbers: $base_{16}$

  [0 1 2 3 4 5 6 7 8 9 A B C D E F]

1-Bit Binary Number (Binary System)

1-Bit Binary Numbers
0
1

2-Bit Binary Number (Binary System)

1-Bit Binary Numbers	2-Bit Binary Numbers
0	00
1	01
	10
	11

3-Bit Binary Numbers (Octal System)

1-Bit Binary Numbers	2-Bit Binary Numbers	3-Bit Binary Numbers
0	00	000
1	01	001
	10	010
	11	011
		100
		101
		110
		111

4-Bit Binary Numbers (Hexadecimal System)

1-Bit Binary Numbers	2-Bit Binary Numbers	3-Bit Binary Numbers	4-Bit Binary Numbers	Decimal Equivalents
0	00	000	0000	0
1	01	001	0001	1
	10	010	0010	2
	11	011	0011	3
		100	0100	4
		101	0101	5
		110	0110	6
		111	0111	7
			1000	8
			1001	9
			1010	10
			1011	11
			1100	12
			1101	13
			1110	14
			1111	15

Number conversions

- Our aim is to learn how to convert:
- Decimal numbers to Binary numbers and back
- $(xxxx)_{10} \Leftrightarrow (yyyy)_2$

Number with Base-10 (Decimal)

- $NumberWithBase_{10}$
  - $(\ 9 \quad 5 \quad 3\ )_{10}$

    $= 900 \quad +50 \quad +3$

    $= 9*100 \quad +5*10 \quad +3*1$

    $= 9*10^2 \quad +5*10^1 \quad +3*10^0$

Multiply by the proper power of 10

- NumberWithBase$_{10}$
  - $(9 \quad 5 \quad 3)_{10}$

$= 900 \quad +50 \quad +3$

$= 9*100 \quad +5*10 \quad +3*1$

$= 9*10^2 \quad +5*10^1 \quad +3*10^0$

We multiply each digit by the appropriate power of 10

Convert binary number to decimal number

- NumberWithBase$_{10}$
  - $(953)_{10}$

    $= 9 * 10^2 + 5 * 10^1 + 3 * 10^0$

- Convert: From base-2 to base-10
  - $(1\ 1\ 1\ 0)_2 = (\ ?\ )_{10}$

Convert binary number to decimal number

- NumberWithBase$_{10}$
  - $(953)_{10}$

$$= 9 * 10^2 + 5 * 10^1 + 3 * 10^0$$

- Convert: From base-2 to base-10
  - $(1\ 1\ 1\ 0)_2 = (\ ?\ )_{10}$

$$= 1 * 2^3 \;+\; 1 * 2^2 \;+\; 1 * 2^1 \;+\; 0 * 2^0$$

We multiply each binary digit by the appropriate power of 2

Perform the simple arithmetic operations

- NumberWithBase$_{10}$
  - $(953)_{10}$

    $= 9 * 10^2 + 5 * 10^1 + 3 * 10^0$

- Convert: From base-2 to base-10
  - $(1\ 1\ 1\ 0)_2 = (\ ?\ )_{10}$

$$= 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$$

$$= 8 + 4 + 2 + 0$$

$$= 12 + 2$$

$$= (14)_{10}$$

The result

Therefore…

- $(1\ 1\ 1\ 0)_2 = (14)_{10}$

Another view

$$(1\ 1\ 1\ 0)_2$$

$$0 * 2^0 = 0$$

$$1 * 2^1 = 2$$

$$1 * 2^2 = 4$$

$$1 * 2^3 = 8$$

$$(14)_{10}$$

Another example

Convert: From $base_2$ to $base_{10}$

- $(1\ 1\ 0\ 1\ 1)_2$

  $= 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$

  $= 16 + 8 + 0 + 2 + 1$

  $= (27)_{10}$

- Therefore the result is: $(1\ 1\ 0\ 1\ 1)_2 = (27)_{10}$

One more example

- $(\ 1\ 0\ 1\ 1\ )_2 = (\ ?\ )_{10}$

  $= 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$

  $= 8 + 0 + 2 + 1$

  $= (11)_{10}$

- Therefore the result is: $(\ 1\ 0\ 1\ 1\ )_2 = (\ 11\ )_{10}$

Convert: From base-10 ⇨ base-2

The inverse …

Binary to Decimal, use: Multiplications
Decimal to Binary, use: Divisions

Convert: From base-10 ⇨ base-2

- $(2\ 7)_{10} = (?)_2$
- Divide decimal 27 by 2
  - 27 / 2 = 13 remainder 1
  - 13 / 2 = 6 remainder 1
  - 6 / 2 = 3 remainder 0
  - 3 / 2 = 1 remainder 1
  - 1 / 2 = 0 remainder 1

When the result of the division is ZERO. We stop

The result …

- $(2\ 7)_{10} = (?)_2$
- Divide decimal 27 by 2
  - 27 / 2 = 13 remainder 1
  - 13 / 2 = 6 remainder 1
  - 6 / 2 = 3 remainder 0
  - 3 / 2 = 1 remainder 1
  - 1 / 2 = 0 remainder 1

Write remainders of division (bottom to top)

The result …

Write remainders of division (bottom to top)

- $(2\ 7)_{10} = (?)_2$
- Divide decimal 27 by 2
  - 27 / 2 = 13 remainder 1
  - 13 / 2 = 6 remainder 1
  - 6 / 2 = 3 remainder 0
  - 3 / 2 = 1 remainder 1
  - 1 / 2 = 0 remainder 1

1 1 0 1 1

Therefore…

- $(2\ 7)_{10} = (11011)_2$

Another example

- $(2\ 8)_{10} = (?)_2$
- Divide decimal 28 by 2
  - 28 / 2 = 14 remainder 0
  - 14 / 2 = 7 remainder 0
  - 7 / 2 = 3 remainder 1
  - 3 / 2 = 1 remainder 1
  - 1 / 2 = 0 remainder 1

When the result of the division is ZERO. We stop

Another example …

- $(2\ 8)_{10} = (?)_2$

Write remainders of division (bottom to top)

- Divide decimal 28 by 2
  - 28 / 2 = 14 remainder 0
  - 14 / 2 = 7 remainder 0
  - 7 / 2 = 3 remainder 1
  - 3 / 2 = 1 remainder 1
  - 1 / 2 = 0 remainder 1

1 1 1 0 0

The result

- $(2\ 8)_{10} = (11100)_2$

Example: $(1101)_2$, using weights

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	**<< weights**
256	128	64	32	16	8	4	2	1	
					*	*	*	*	
					1	1	0	1	
					=	=	=	=	
					8	4	0	1	

Add: 8 +4 +0 +1 = 13

Therefore: $(1101)_2 \rightarrow (13)_{10}$

Example: $(100111001)_2$, using weights

- Weights:
- 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
- 256 128 64 32 16 8 4 2 1

256	128	64	32	16	8	4	2	1
1	0	0	1	1	1	0	0	1
256			+32	+16	+8			+1

Add: 256+32+16+8+1 = 313

Therefore: $(100111001)_2 \rightarrow (313)_{10}$

Conclusion: Binary and Decimal conversion

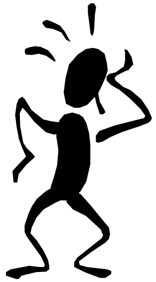
1. Converting a binary (base-2) to decimal (base-10), form the appropriate sum of powers in base "2"

2. Converting a decimal integer (base-10) to binary (base-2), divide by "2", use the remainders as coefficients. Collect the coefficients bottom-top.

Conversion from different bases

Decimal (Base-10) to Octal (Base-8)

- $Base_{10}$ → $Base_8$
- $(\ 1\ 5\ 3\ )_{10} = (?)_8$

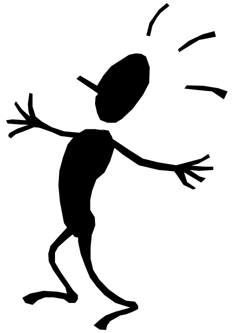


Base-10 to Base-8

- $\text{Base}_{10} \rightarrow \text{Base}_8$
- $(1\ 5\ 3)_{10} = (?)_8$

  - $153 / 8 = 19\ \mathbf{1}/8$
  - $19 / 8 = 2\ \mathbf{3}/8$
  - $2 / 8 = 0\ \mathbf{2}/8$

  $= (\mathbf{231})_8$



Base-10 to Base-16

- $\text{Base}_{10} \Rightarrow \text{Base}_{16}$
- $(2\ 4\ 5)_{10} = (\ ?\ )_{16}$

Base-10 to Base-8

- $\text{Base}_{10} \rightarrow \text{Base}_{16}$
- $(2\ 4\ 5)_{10} = (\ ?\ )_{16}$

  - 245 / 16 = 15   **5**/16
  - 15 / 16 = 0   **15**/16

  $= (\mathbf{F5})_{16}$

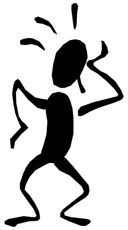
Decimal (with fraction part) to Binary

Base-10 to Base-2

- $(0\ .\ 6\ 8\ 7\ 5)_{10} = (?)_2$

- Lets try to solve this problem on your own …

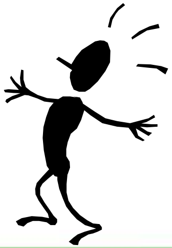
- **You have … 3 minutes**



Base-10 to Base-2

- $(0.6875)_{10} = \mathbf{(?)}_2$
  - $0.6875 * 2 = 1.3750$
  - $0.3750 * 2 = 0.7500$
  - $0.7500 * 2 = 1.5000$
  - $0.5000 * 2 = 1.0000$
  - $0.0000 * 2 = 0.0000$

  $= (0.10110)_2$



1. Multiply by 2 the fraction
2. Keep the leftmost digit
3. Take the fraction, as long as is not all zeros, and multiply it again by 2.
4. Continue the multiplication-by-2 process until you either get an all zero fraction part, or a repeating pattern [choose when to stop]
5. Start from the top and select all leftmost digits. This is the answer.

Decimal numbers with … integer and fraction parts

Conversion to binary

Two steps: Decimal (integer+fraction) to Binary

- The conversion of decimal numbers with integer and fraction parts is done by converting the integer and the fraction and then combining the two answers.

- Example:
  - $(41.6875)_{10} \Rightarrow (?)_2$

$(41)_{10} = (101001)_2$ (Step-1)

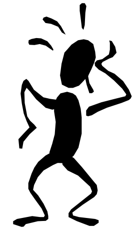
$(0.6875)_{10} = (0.10110)_2$ (Step-2)

Result: $(101001.10110)_2$

Binary-Octal-Hex

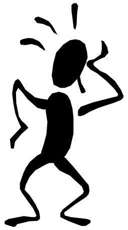
Conversion from binary to octal

$(010110001101011.111100000010)_2 = (?)_8$



Groups of 3

$(010110001101011.111100000010)_2 = (?)_8$



Conversion from binary to octal

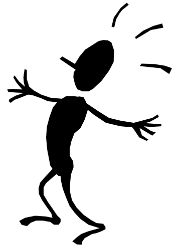
A

$$\mathbf{(010\ 110\ \ 001\ 101\ \ 011\ .\ 111\ 100\ 000\ 010)_2 = (?)_8}$$

- Step 1: Start from A, go left ⇨ 3 digits
- Step 2: Start from A, go right ⇨ 3 digits

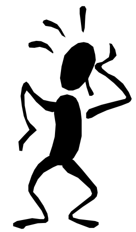
– 011 = 3	111 = 7
– 101 = 5	100 = 4
– 001 = 1	000 = 0
– 110 = 6	010 = 2
– 010 = 2	

$= (26153.7402)_8$



Conversion from binary to hex

(10110001101011 . 11110010)$_2$ = (?)$_{16}$



Groups of 4

$(101100011010 11 \,.\, 11110010)_2 = (?)_{16}$

Conversion from binary to hex

B

- **(10 1100 0110 1011 . 1111 0010)$_2$ = (?)$_{16}$**

  .

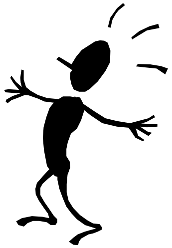
- Step 1: Start from B, go left ⇨ 4 digits
- Step 2: Start from B, go right ⇨ 4 digits

  - 1011 = B
  - 0110 = 6
  - 1100 = C
  - 0010 = 2

  1111 = F

  0010 = 2

  = (2C6B.F2)$_{16}$



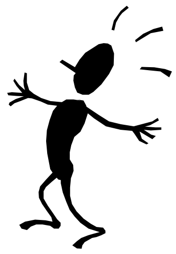
Conversion from octal to binary

- $(306.7)_8 = (?)_2$

Conversion from octal to binary

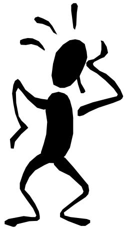
- $(306.7)_8 = (?)_2$

$= (011\ 000\ 110\ .\ 111)_2$



Conversion from hex to binary

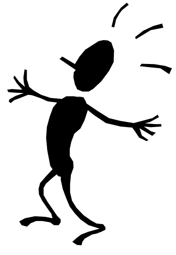
- $(306.D)_{16} = (?)_2$



Conversion from hex to binary

- $(306.D)_{16} = (?)_2$

$= (0011\ 0000\ 0110\ .\ 1101)_2$



Conclusion

- Converting from base “2” to base 10, form the appropriate sum of powers in base “2”
- Converting from a decimal integer to base “2”, divide by “2”, use the remainders as coefficients
- Converting a decimal fraction to base “2”, multiply by “2”, use the integers as coefficients
- Converting octal and hex to binary, form groups of 3 or 4 digits.


WELCOME TO MY CLASS
WELCOME TO MY CLASS
WELCOME TO MY CLASS
WELCOME TO MY CLASS