# INTRO TO BINARY SYSTEMS

# Main Numeral Systems

| | | | |
|---|---|---|---|
| Decimal | (Base 10) | Number System | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 |
| Binary | (Base 2) | Number System | 0, 1 |
| Ternary | (Base 3) | Number System | 0, 1, 2 |
| Quaternary | (Base 4) | Number System | 0, 1, 2, 3 |
| .......... . | | | |
| Hexadecimal | (Base 16) | Number System | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F |

# Converting Between Counting Systems

| Decimal | Hex | Binary |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| 10 | A | 1010 |
| 11 | B | 1011 |
| 12 | C | 1100 |
| 13 | D | 1101 |
| 14 | E | 1110 |
| 15 | F | 1111 |

$(10110)_b = 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 1\times 2^1 + 0\times 2^0 \rightarrow (22)_d$

$(10010)_b = 1\times 2^4 + 0\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 \rightarrow (18)_d$

$(AE3)_{hex} = A\times 16^2 + E\times 16^1 + 3\times 16^0 \rightarrow (2787)_d$

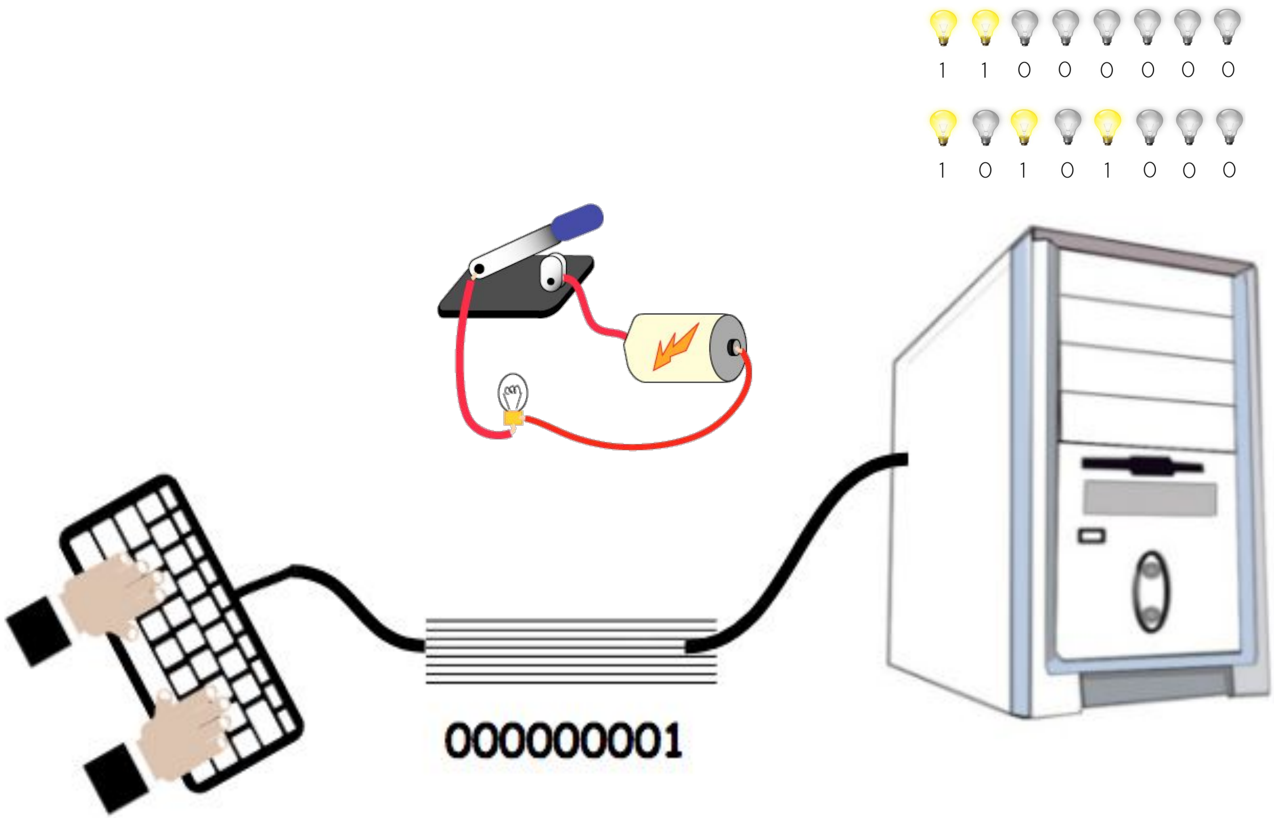
$(B2F)_{hex} = B\times 16^2 + 2\times 16^1 + F\times 16^0 \rightarrow (2863)_d$

The value of a number is often expressed in hex

The 16-bit number **FFFF** hex corresponds to ----> $(1111\ 1111\ 1111\ 1111)_b$

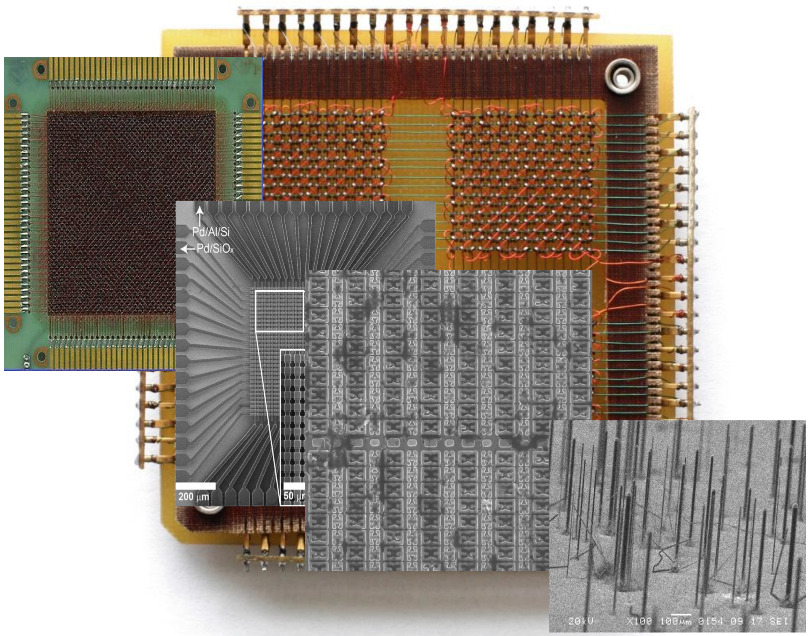
The 16-bit number **4AE3** hex corresponds to ----> $(0100\ 1010\ 1110\ 0011)_b$

# Computer Language



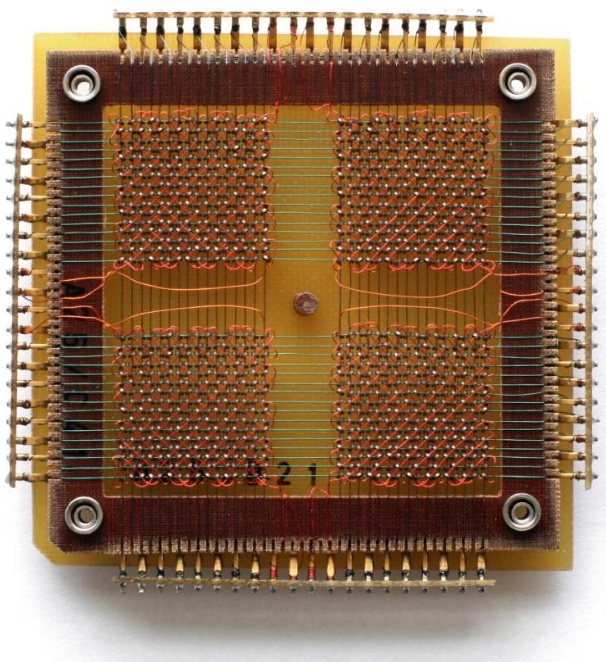
# Computer Memory

- <u>Memory</u> – A collection of storage cells together with the necessary circuits to transfer information to and from them.




| | | | |
|---|---|---|---|
| byte | b | 8 bits | 1 byte |
| kilobyte | Kb | 1024 bytes | 1 024 bytes |
| megabyte | MB | 1024 KB | 1 048 576 bytes |
| gigabyte | GB | 1024 MB | 1 073 741 824 bytes |
| terabyte | TB | 1024 GB | 1 099 511 627 776 bytes |

# Mem Content vs Mem Address



- n-bit memory location can store one of $2^n$ different numbers.
- A 32-bit variable can store any of $2^{32}$ different numbers.

- n-bit address can refer to any of $2^n$ different memory locations.
- A 16-bit address can refer to any of $2^{16}$ = 64K different memory locations.

# Size matters!

**Powers of two**

| | | |
|---|---|---|
| $2^0 = 1$ | $2^5 = 32$ | $2^{10} = 1024 = 1K$ (kilo) |
| $2^1 = 2$ | $2^6 = 64$ | $2^{20} = 1M$ (mega) |
| $2^2 = 4$ | $2^7 = 128$ | $2^{30} = 1G$ (giga) |
| $2^3 = 8$ | $2^8 = 256$ | $2^{40} = 1T$ (tera) |
| $2^4 = 16$ | $2^9 = 512$ | |

Rule: $2^{n+m} = 2^n \times 2^m$

Example: $2^{16} = 2^{10} \times 2^6 = 1k \times 64 = 64k$

**Logarithms (base 2)**

| | | |
|---|---|---|
| $\log_2 1 = 0$ | $\log_2 32 = 5$ | $\log_2 1024 = \log_2 1K = 10$ |
| $\log_2 2 = 1$ | $\log_2 64 = 6$ | $\log_2 1M = 20$ |
| $\log_2 4 = 2$ | $\log_2 128 = 7$ | $\log_2 1G = 30$ |
| $\log_2 8 = 3$ | $\log_2 256 = 8$ | $\log_2 1T = 40$ |
| $\log_2 16 = 4$ | $\log_2 512 = 9$ | |

Rule: $\log_2(n \times m) = \log_2 n + \log_2 m$

Example: $\log_2 64k = \log_2 1k + \log_2 64 = 10+6 = 16$

# Examples

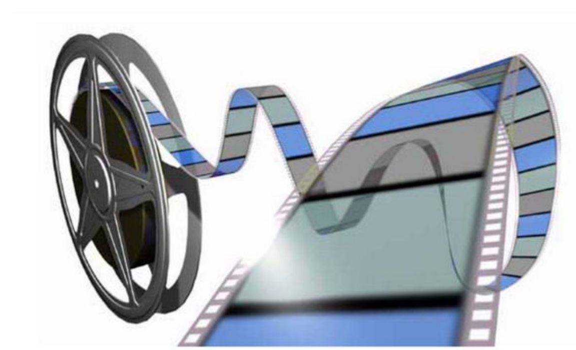
- What is the size the largest memory we can have with 45-bit address?

  $2^{45} = 2^{40} \times 2^{5} = 1T \times 32 = 32T$

- How many binary digits do we need for an address of 2G memory?

  $\log 2G = \log 2 + \mathrm{Log}\ G = 1 + 30 = 31$ bits

# Representation of Data

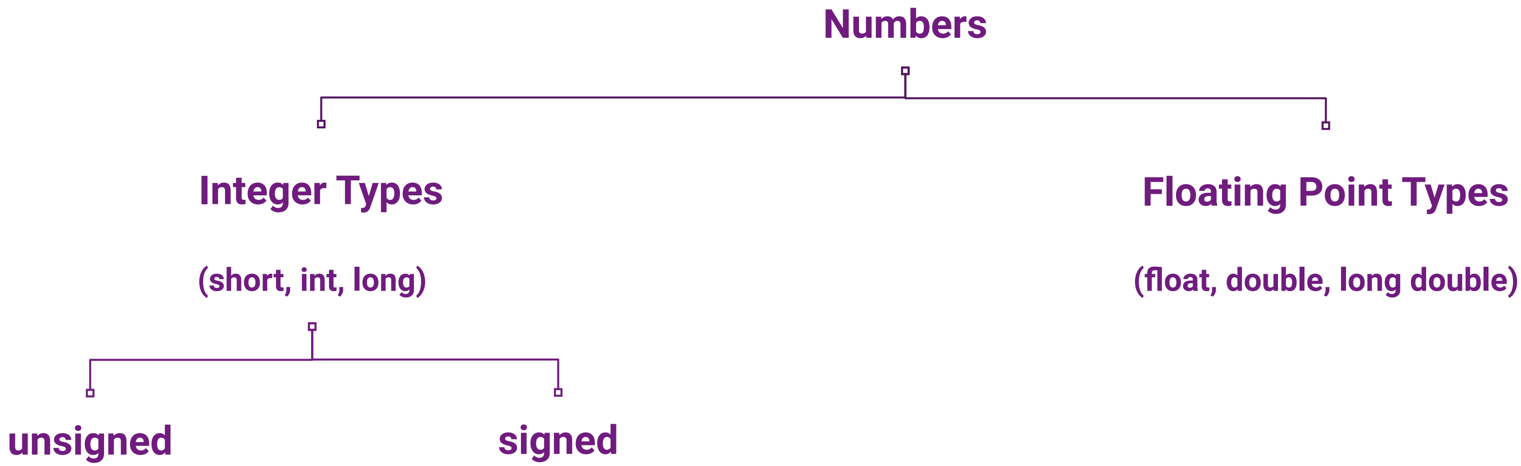



# Representation of Symbols and Text

- **Symbols and Text**
  - Each symbol can be assigned a numeric value.

**ASCII Code: Character to Binary**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0011 0000 | G | 0100 0111 | W | 0101 0111 | m | 0110 1101 | : | 0011 1010 |
| 1 | 0011 0001 | H | 0100 1000 | X | 0101 1000 | n | 0110 1110 | ; | 0011 1011 |
| 2 | 0011 0010 | I | 0100 1001 | Y | 0101 1001 | o | 0110 1111 | ? | 0011 1111 |
| 3 | 0011 0011 | J | 0100 1010 | Z | 0101 1010 | p | 0111 0000 | ! | 0010 0001 |
| 4 | 0011 0100 | K | 0100 1011 | a | 0110 0001 | q | 0111 0001 | ' | 0010 1100 |
| 5 | 0011 0101 | L | 0100 1100 | b | 0110 0010 | r | 0111 0010 | " | 0010 0010 |
| 6 | 0011 0110 | M | 0100 1101 | c | 0110 0011 | s | 0111 0011 | ( | 0010 1000 |
| 7 | 0011 0111 | N | 0100 1110 | d | 0110 0100 | t | 0111 0100 | ) | 0010 1001 |
| 8 | 0011 1000 | O | 0100 1111 | e | 0110 0101 | u | 0111 0101 | space | 0010 0000 |
| 9 | 0011 1001 | P | 0101 0000 | f | 0110 0110 | v | 0111 0110 | | |
| A | 0100 0001 | Q | 0101 0001 | g | 0110 0111 | w | 0111 0111 | | |
| B | 0100 0010 | R | 0101 0010 | h | 0110 1000 | x | 0111 1000 | | |
| C | 0100 0011 | S | 0101 0011 | I | 0110 1001 | y | 0111 1001 | | |
| D | 0100 0100 | T | 0101 0100 | j | 0110 1010 | z | 0111 1010 | | |
| E | 0100 0101 | U | 0101 0101 | k | 0110 1011 | . | 0010 1110 | | |
| F | 0100 0110 | V | 0101 0110 | l | 0110 1100 | , | 0010 0111 | | |


Numbers
Integer Types
(short, int, long)
unsigned
signed
Floating Point Types
(float, double, long double)