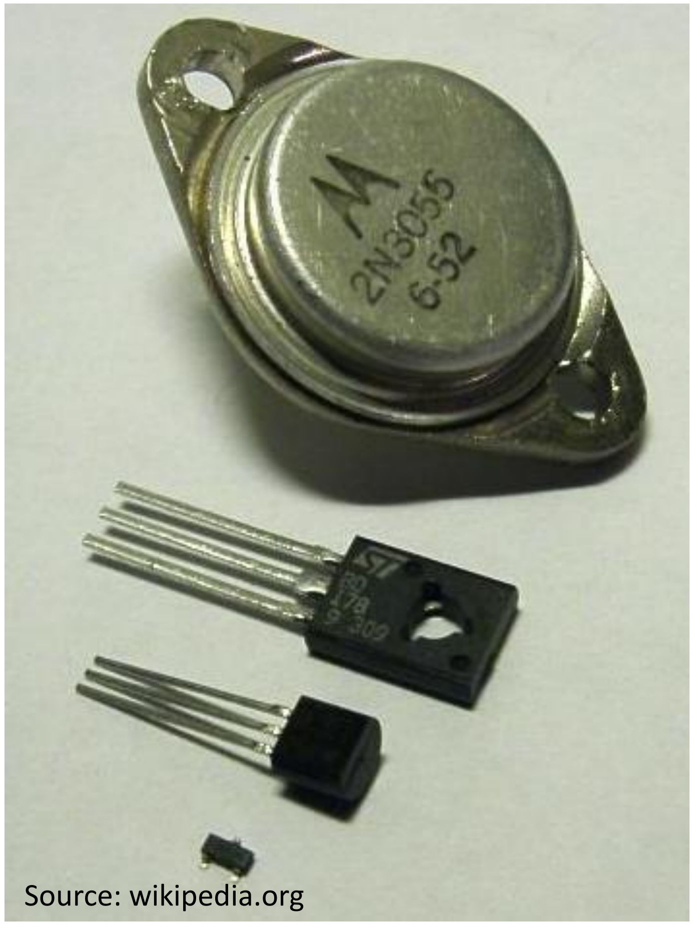
EEE243 – Applied Computer Programming

How Computers Work, Compilation and IDE

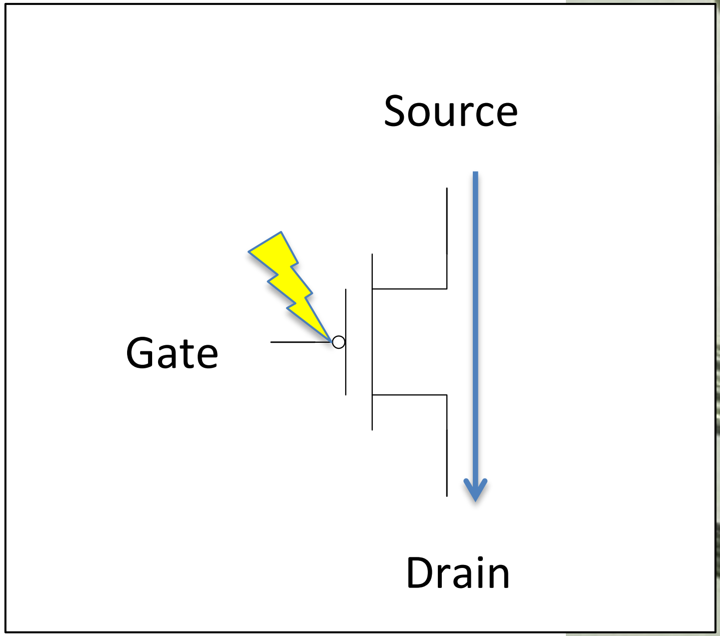
Encoding

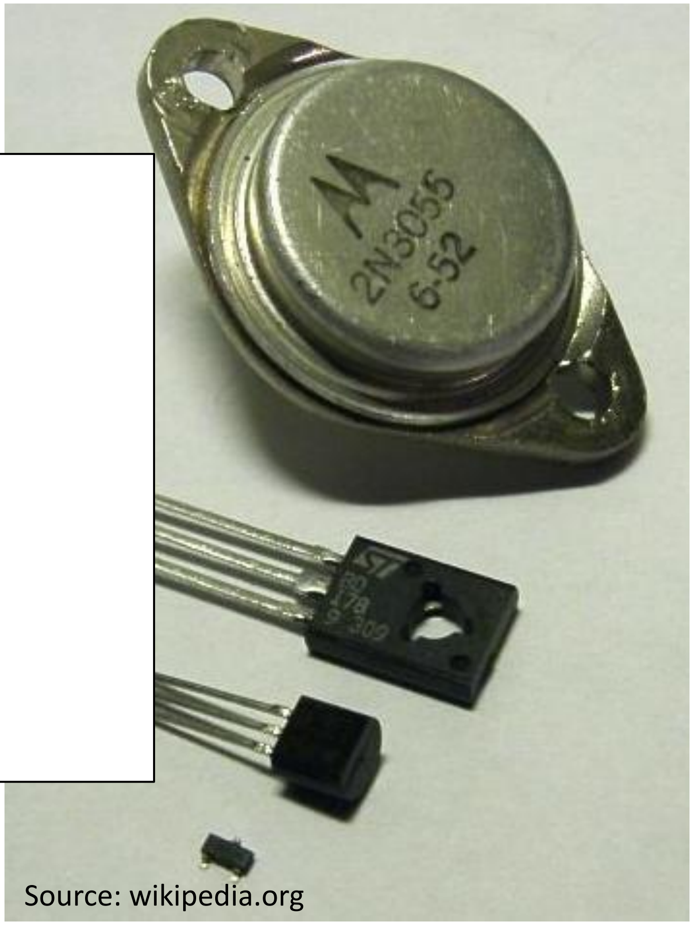
A Computer is a machine for manipulating symbols

Encoding



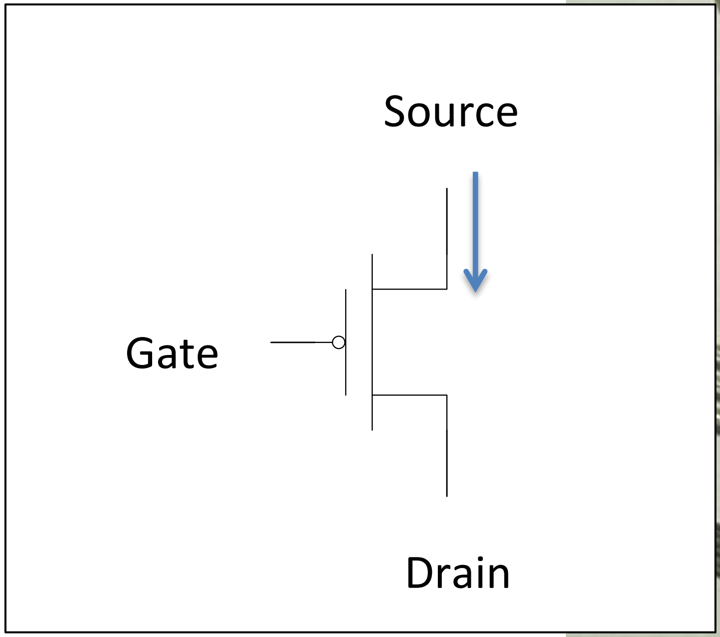
Encoding

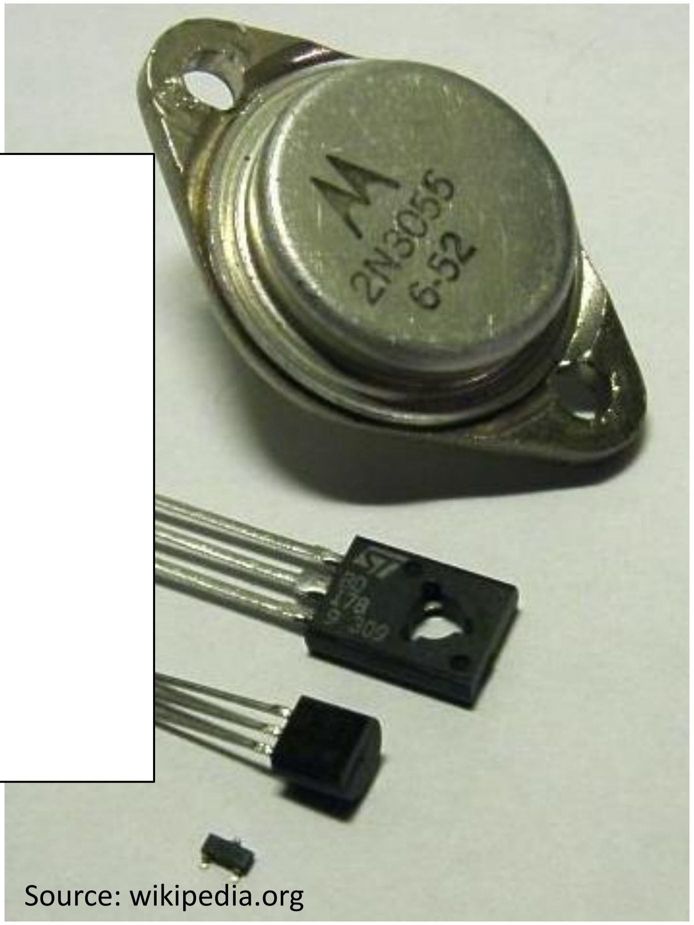




Source: wikipedia.org

Encoding



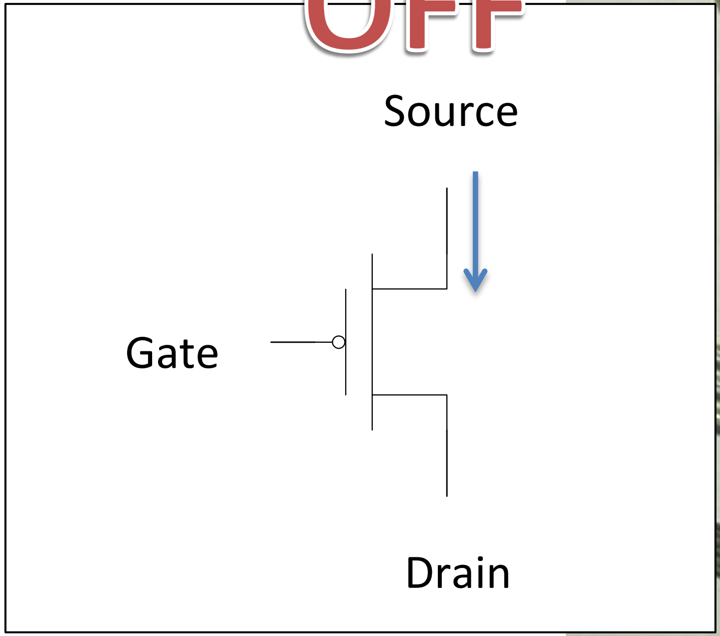


Source: wikipedia.org

Encoding

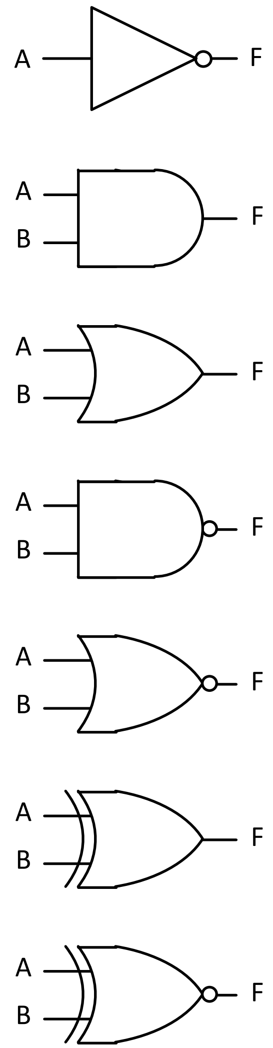
ON TRUE 0

OFF FALSE 1



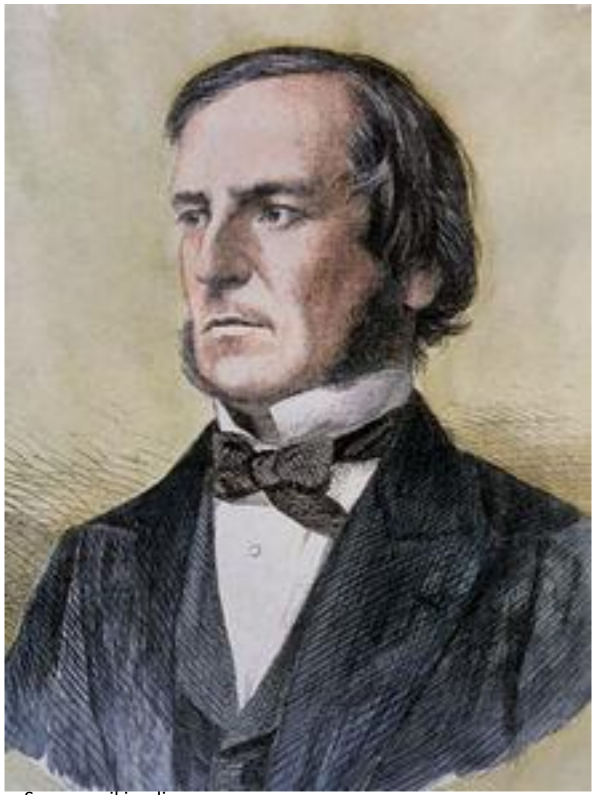
Source: electrical-online.com

Encoding

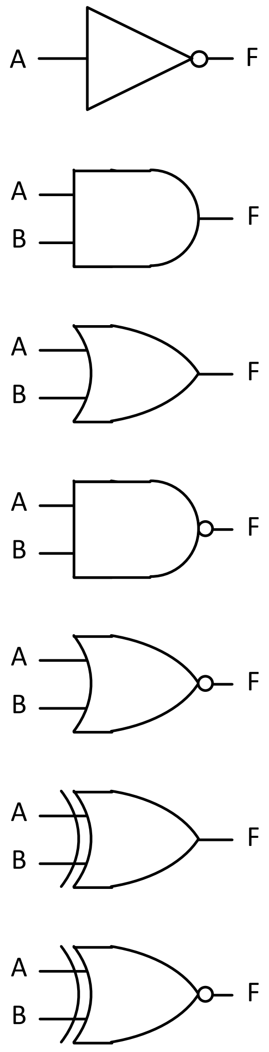


A = The door is closed
B = It is cold outside
F = It is cold inside

$$F = A'B$$


Source: wikipedia.org

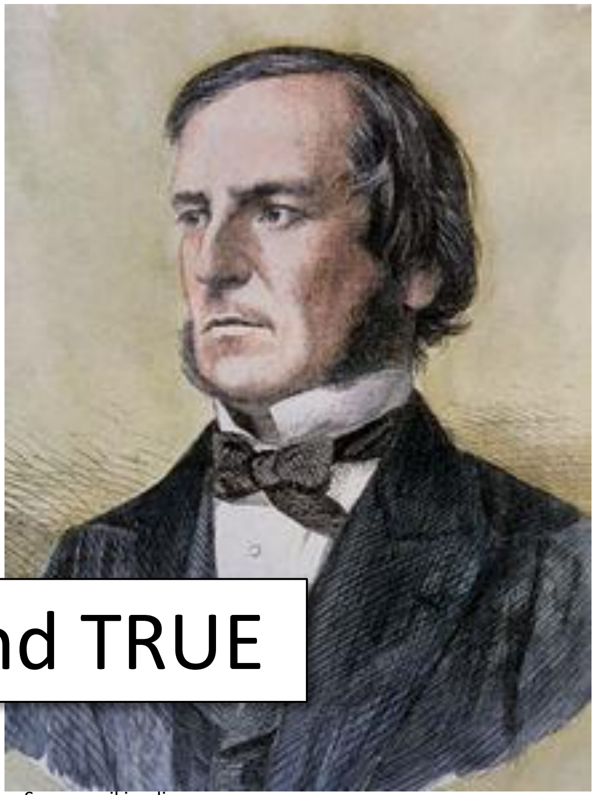
Encoding



A = The door is closed
B = It is cold outside
F = It is cold inside

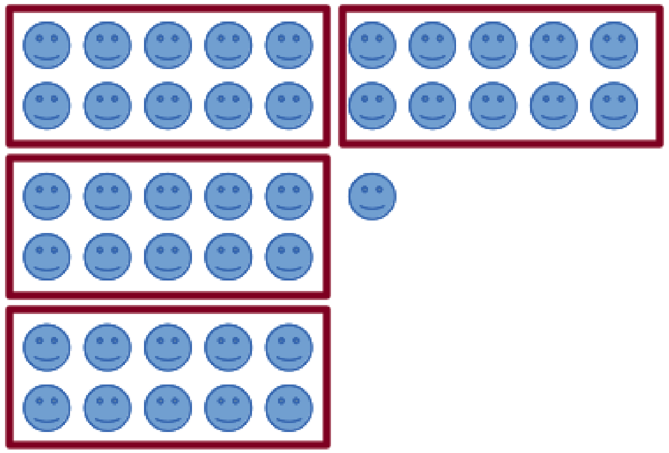
F = A’B

TRUE = (NOT FALSE) and TRUE


Source: wikipedia.org

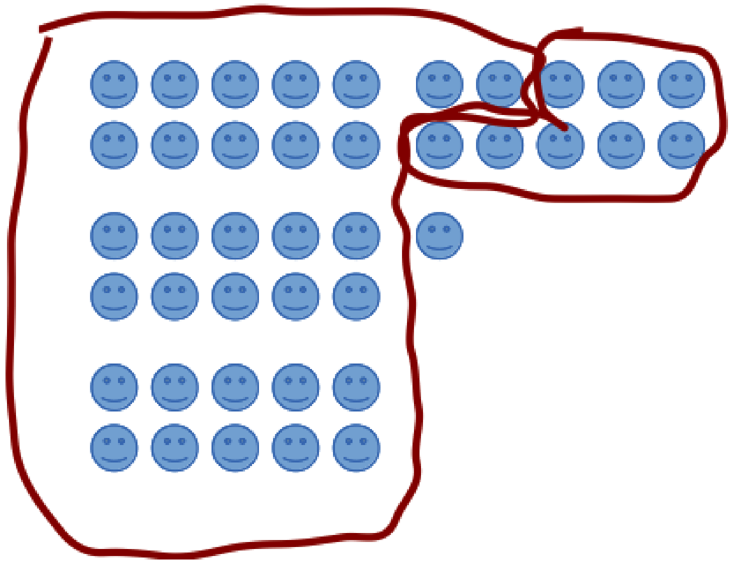
Encoding

41 students



Encoding

101001 students



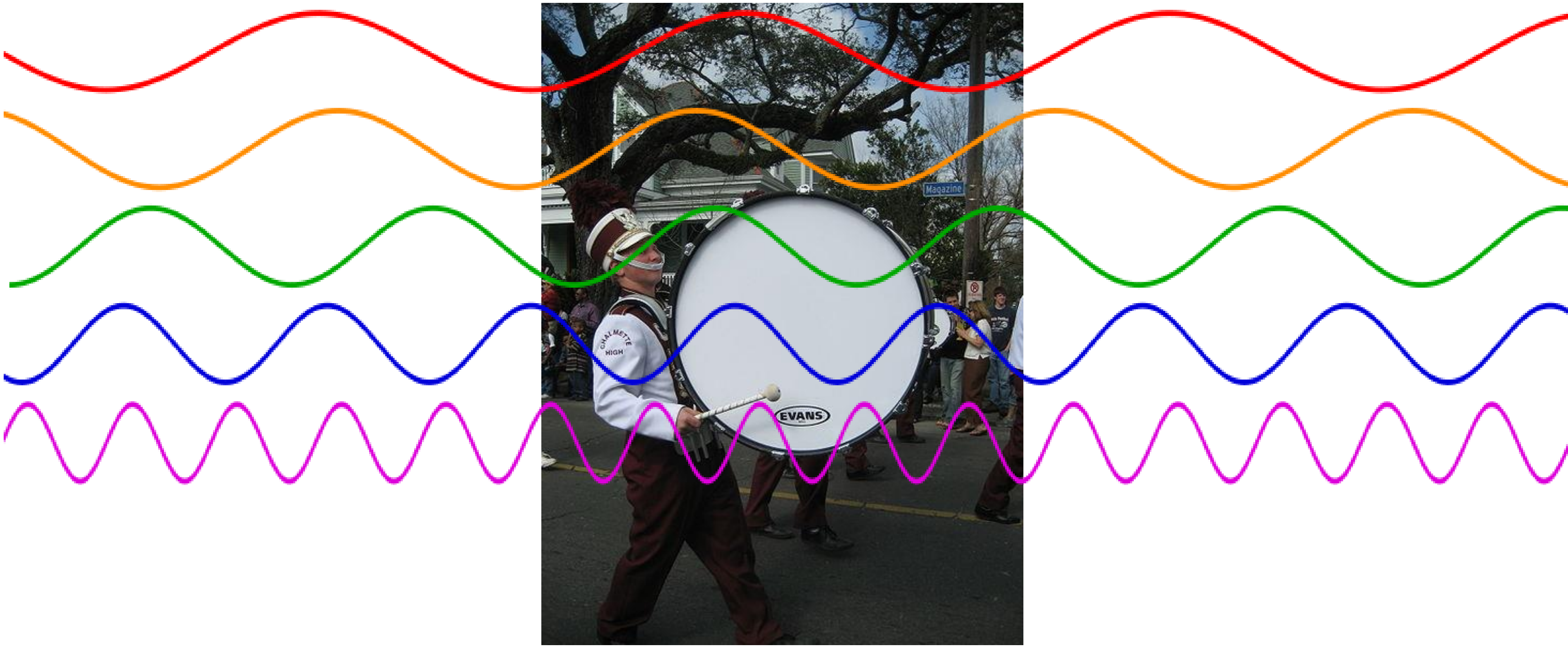
Encoding

Welcome to the Real World

Source: wikipedia.org

Encoding

Welcome to the Real World

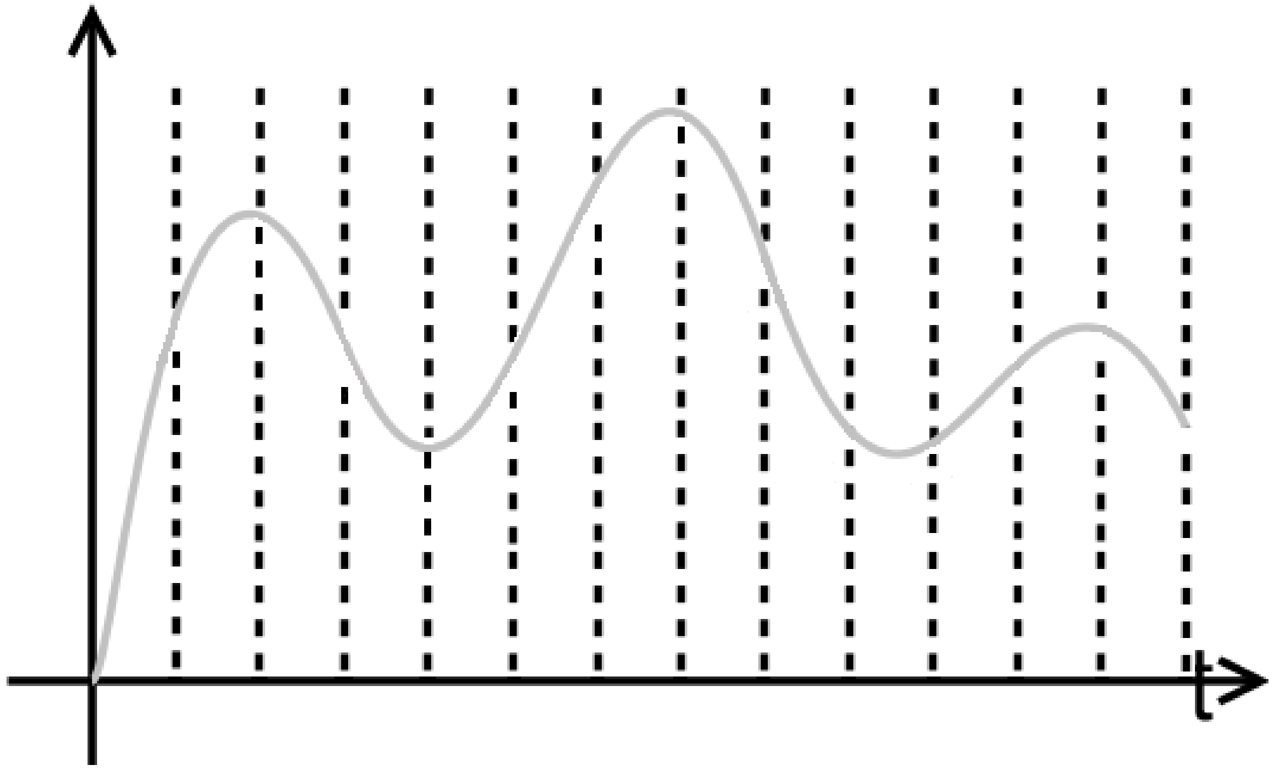


Source: wikipedia.org

Source: wikipedia.org

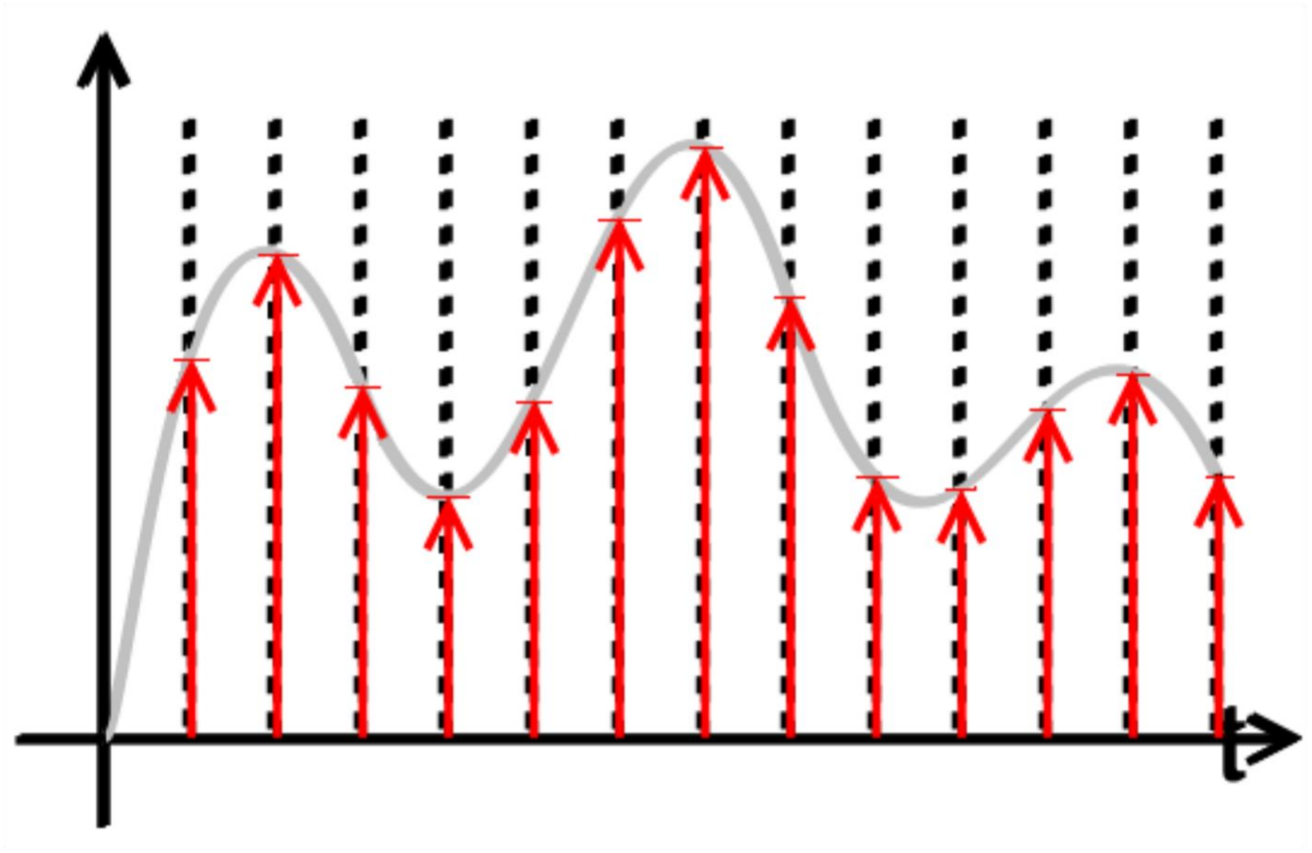
Encoding

Welcome to the <u>Real World</u>



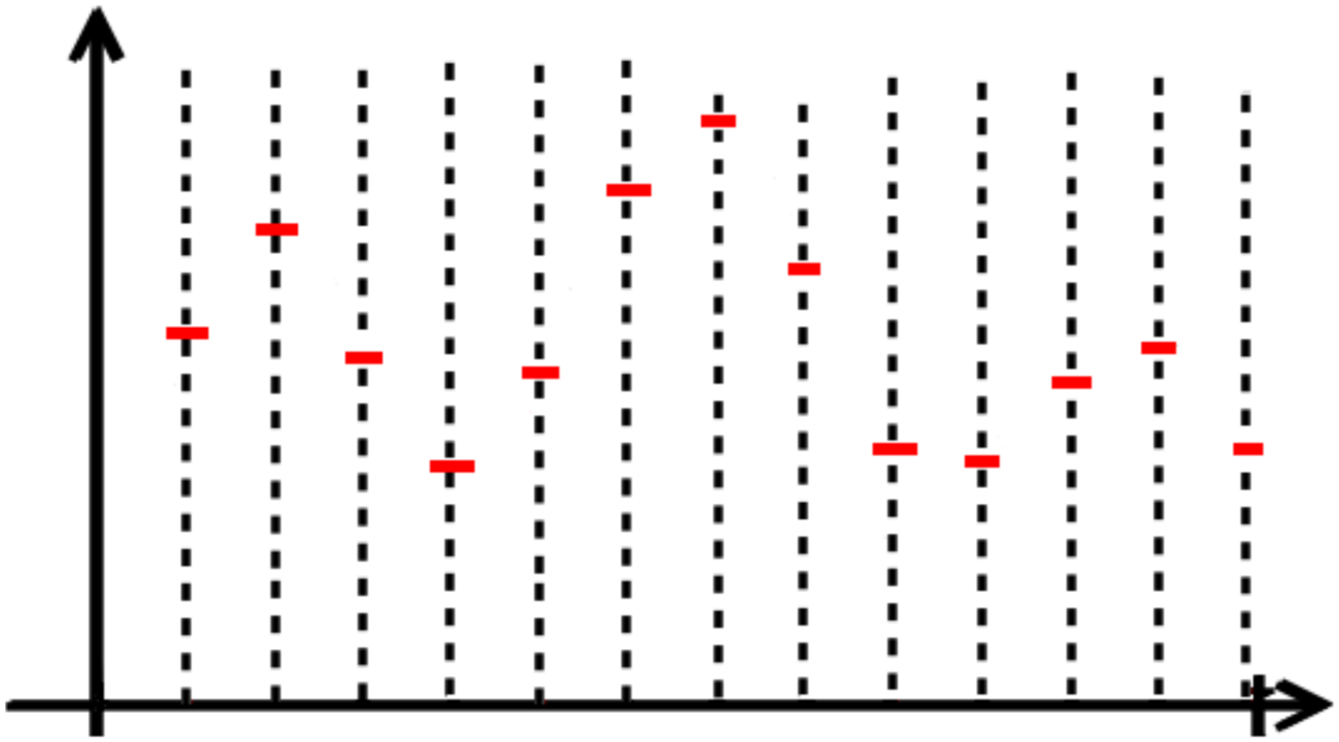
Encoding

Welcome to the Real World



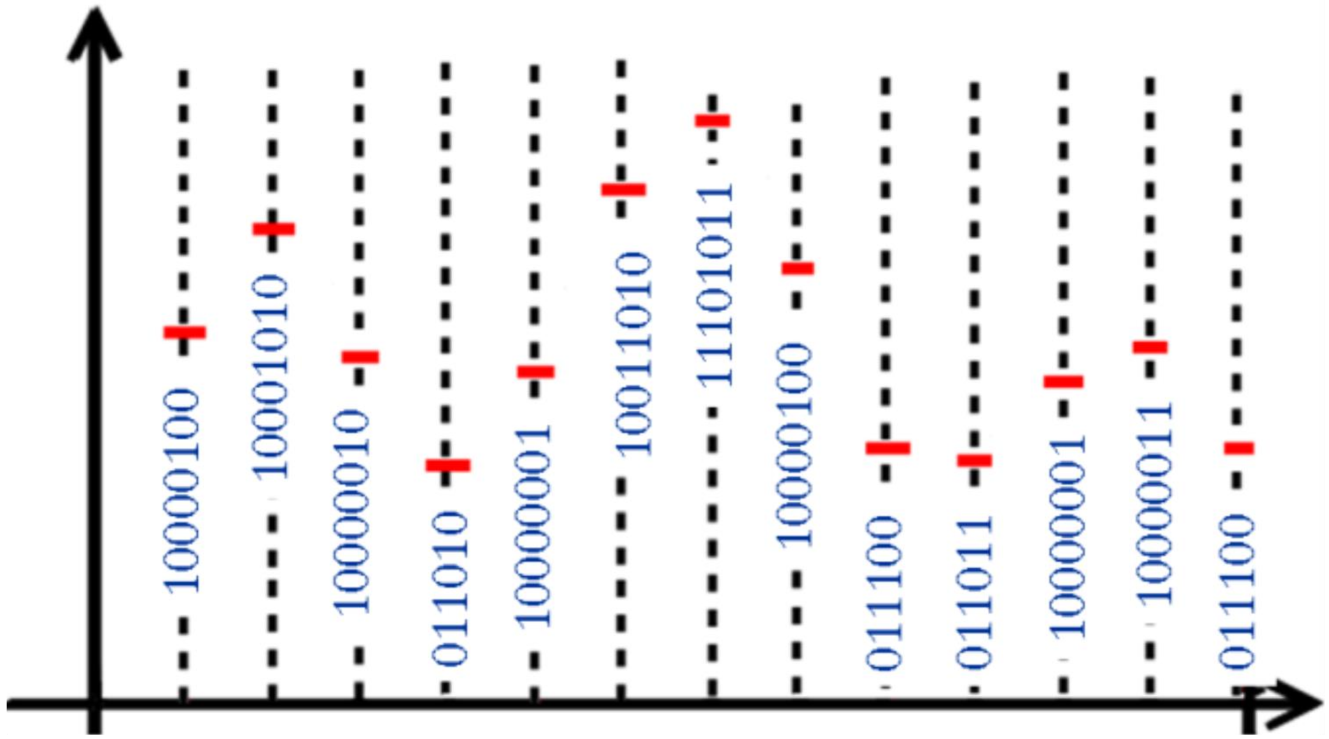
Encoding

Welcome to the <u>Real World</u>

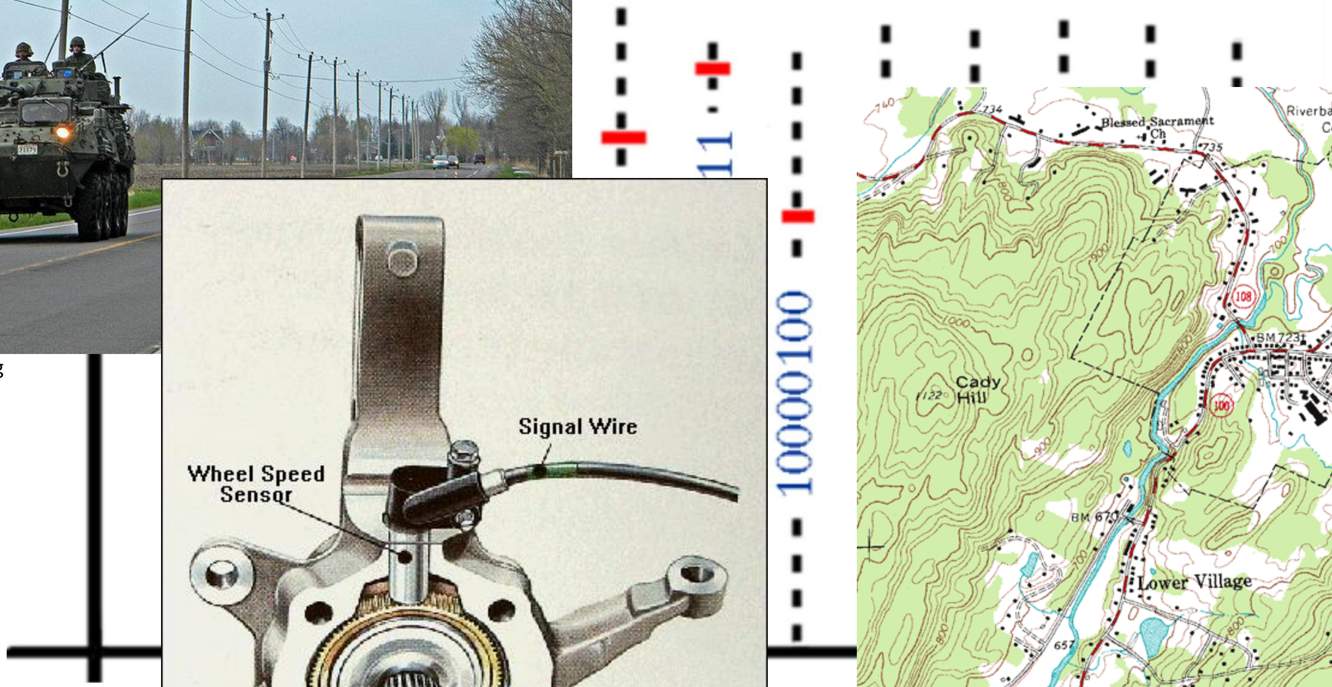


Encoding

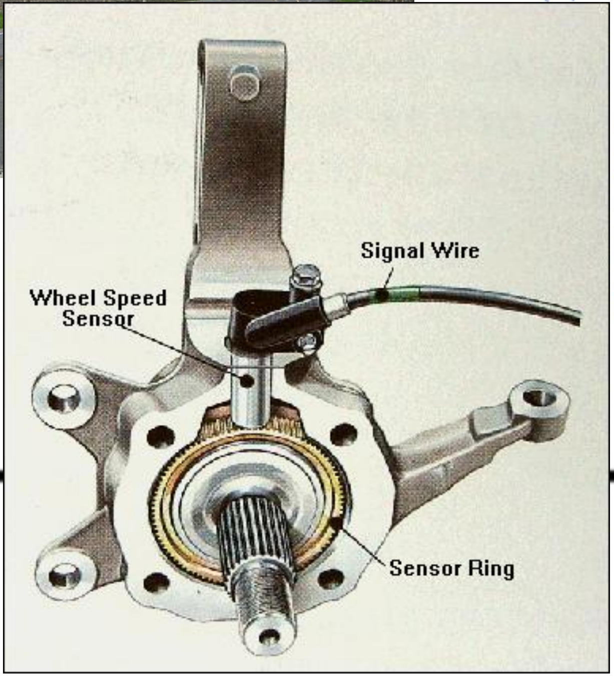
Welcome to the <u>Real World</u>



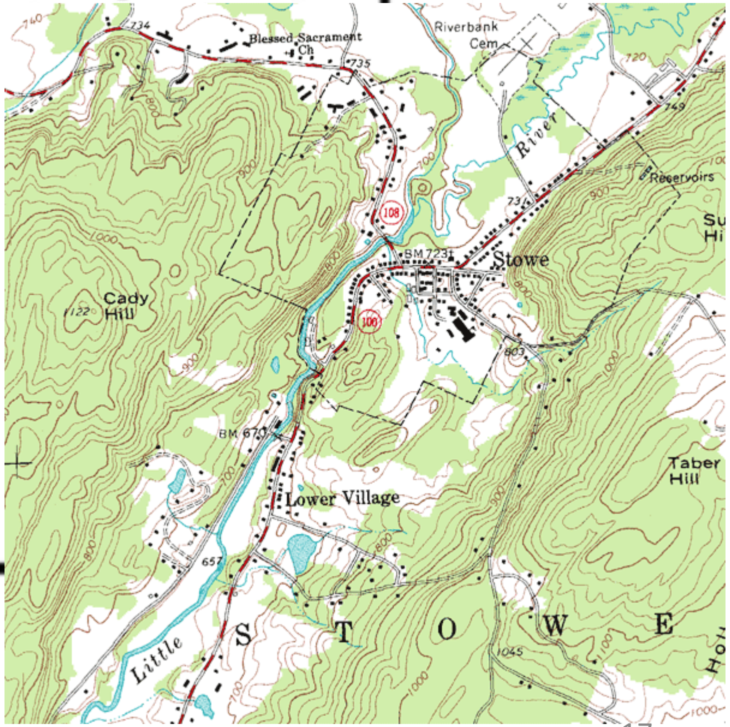
Encoding



Source: wikipedia.org

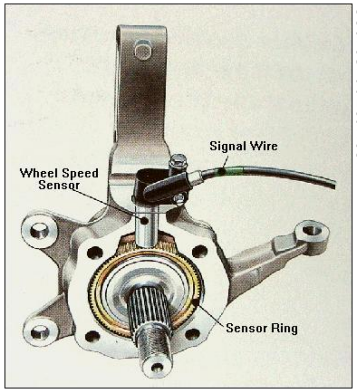


Source: aa1car.com

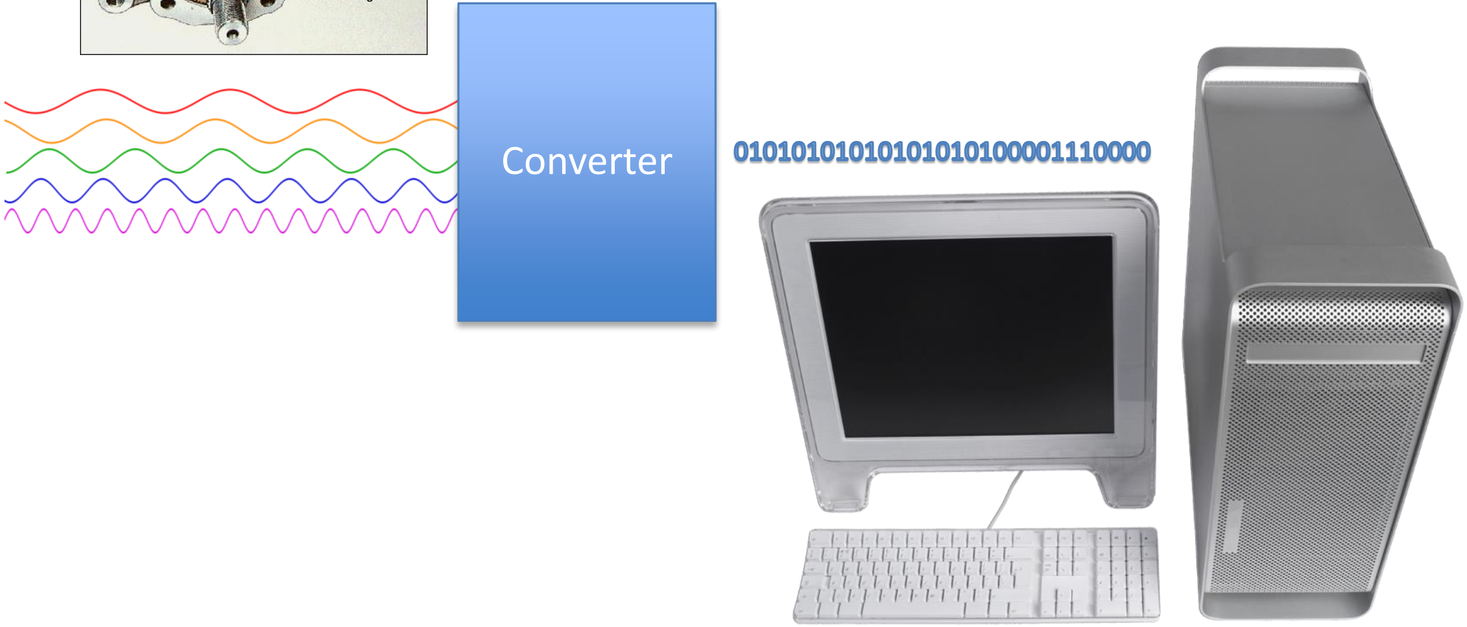


Source: wikipedia.org

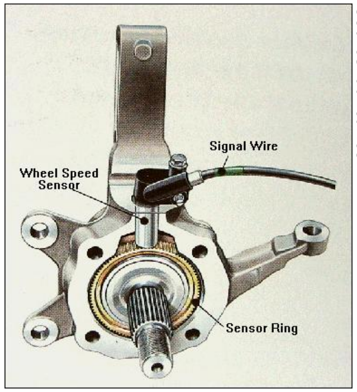
Encoding



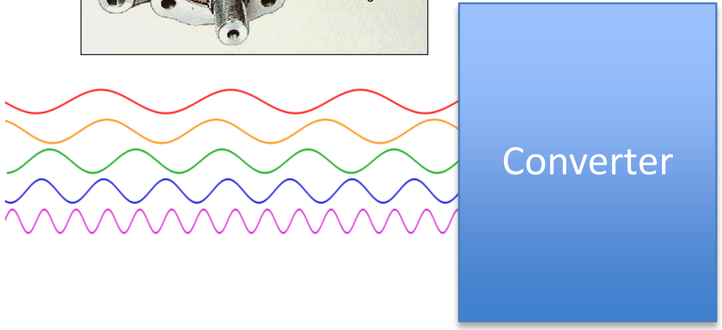
Source: aa1car.com



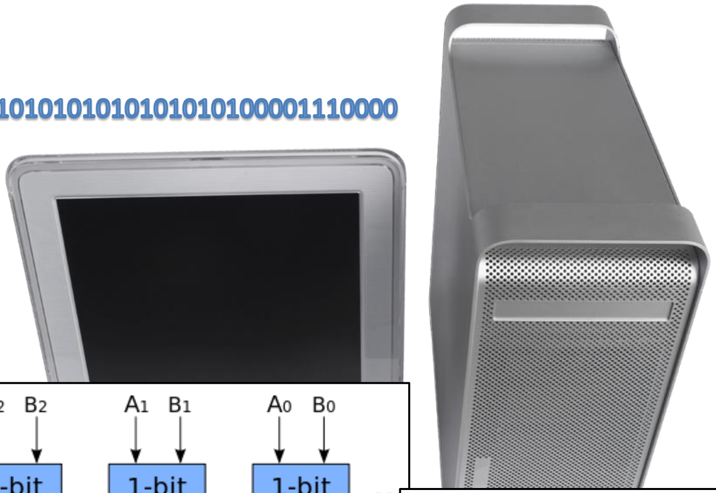
Encoding

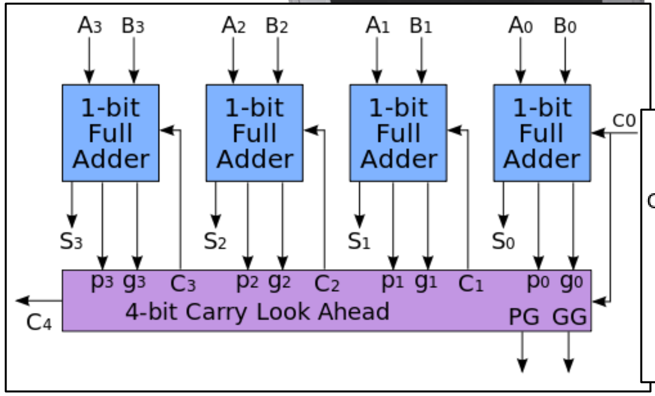


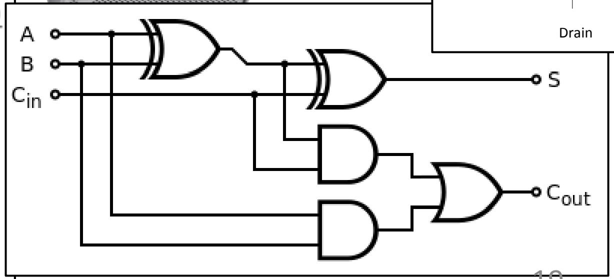
Source: aa1car.com

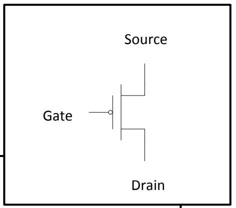


01010101010101010100001110000

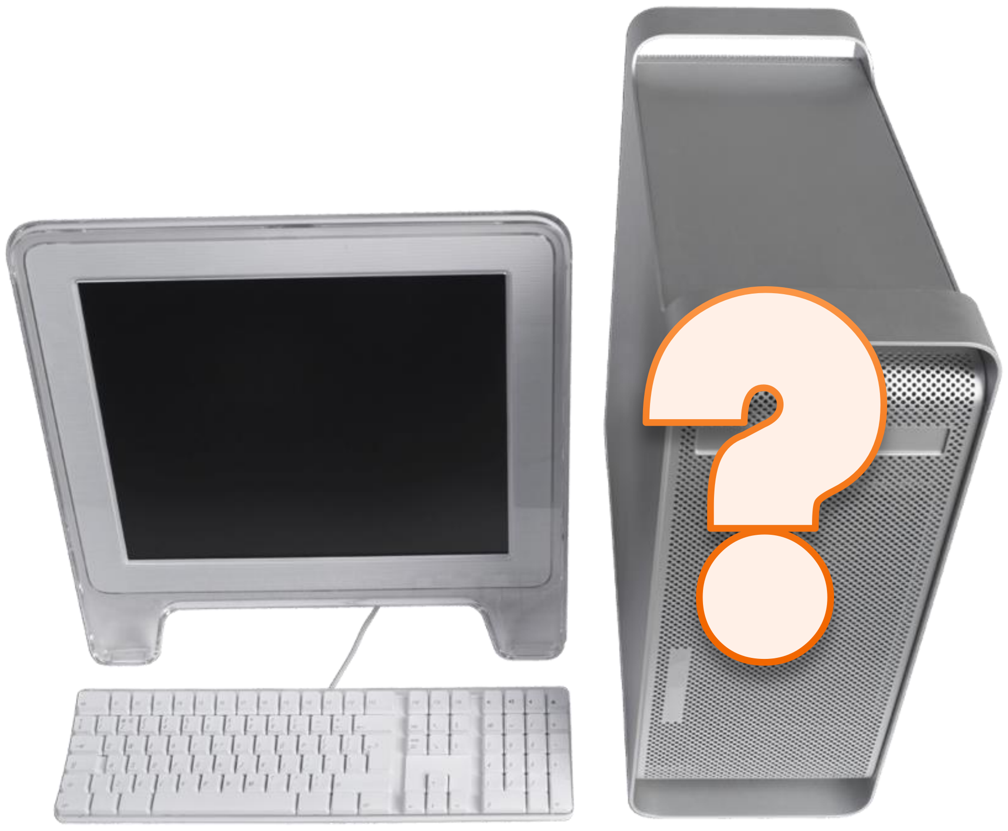






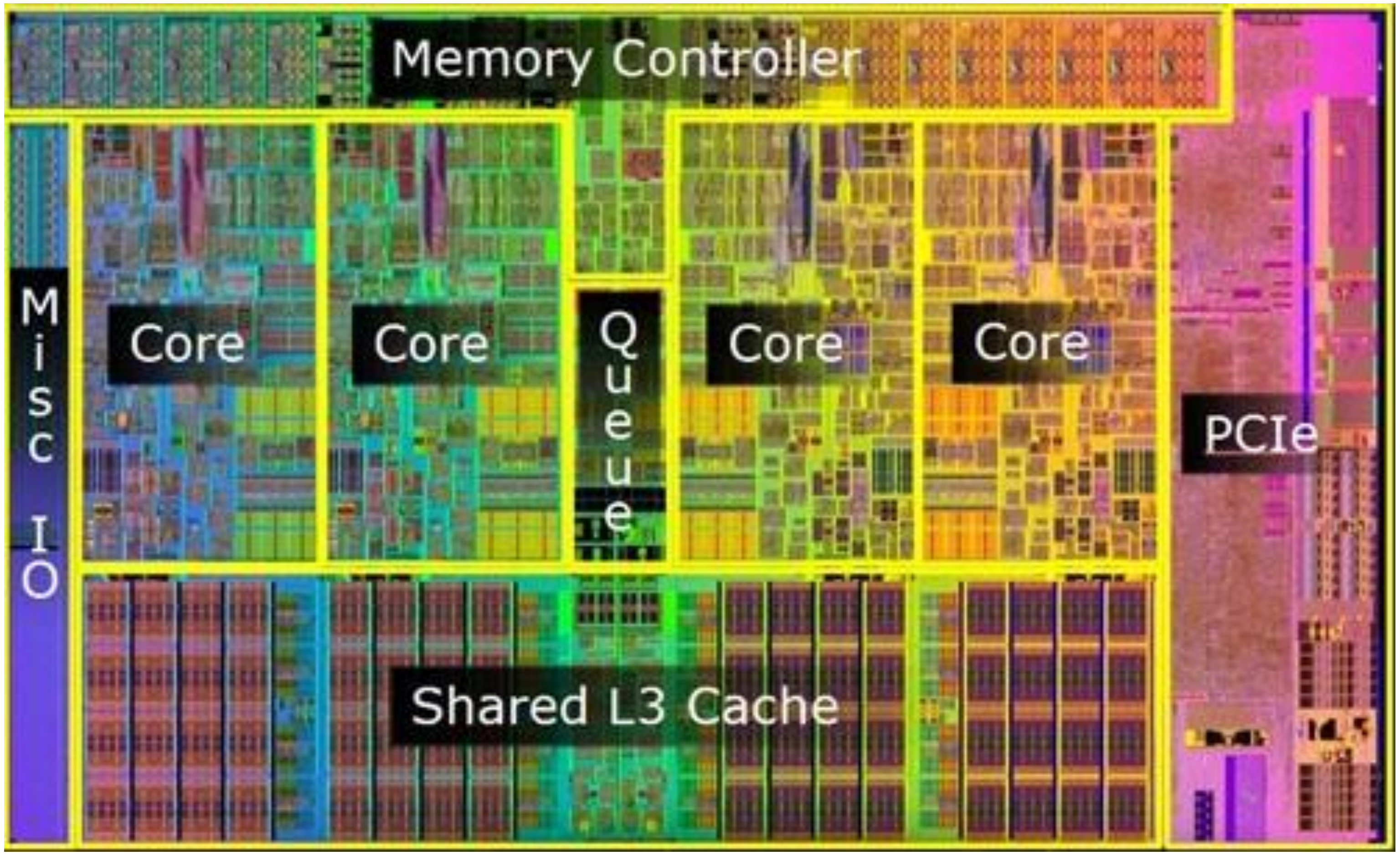


Computer Structure



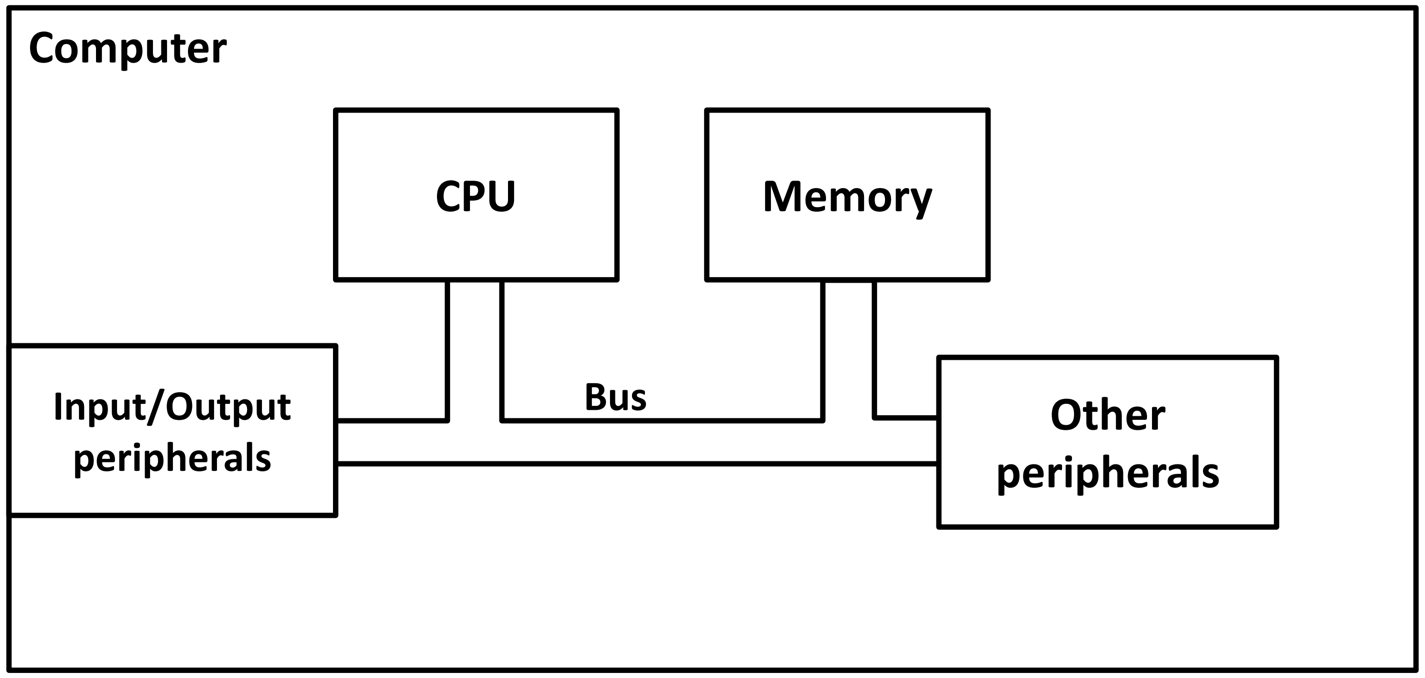
Computer Structure

A machine built out of integrated circuitry



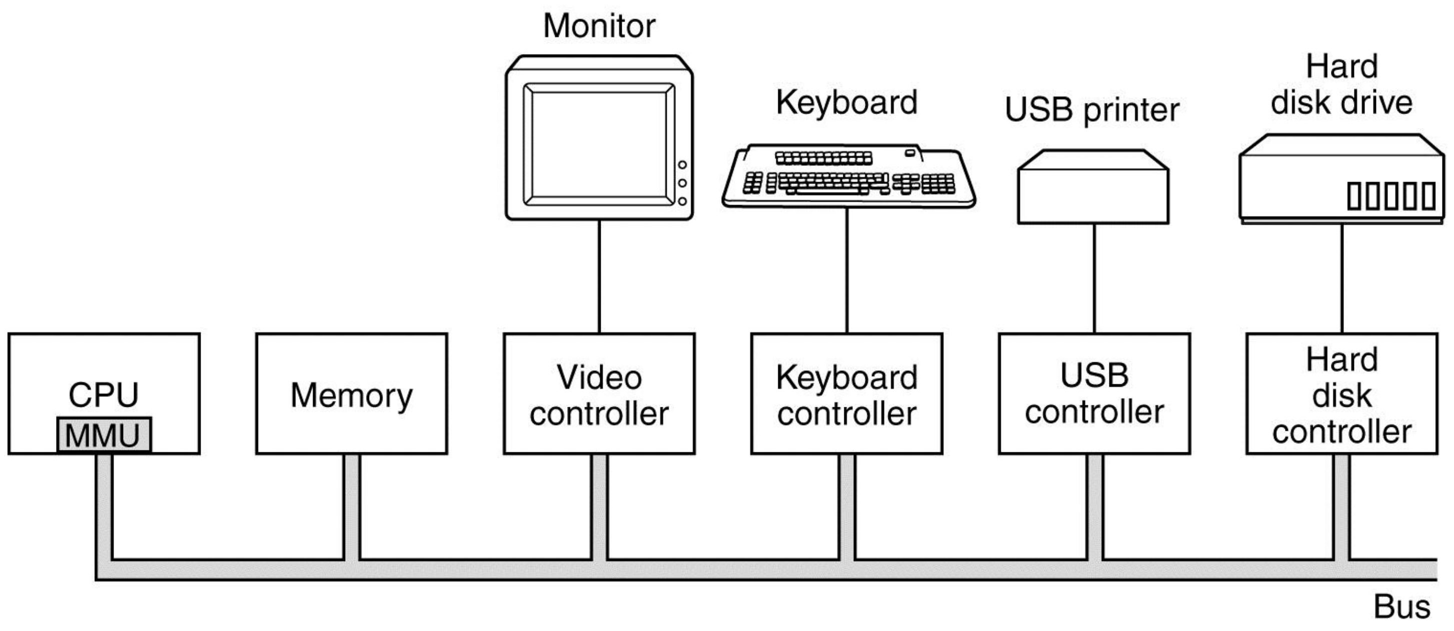
Computer Structure

Abstract view



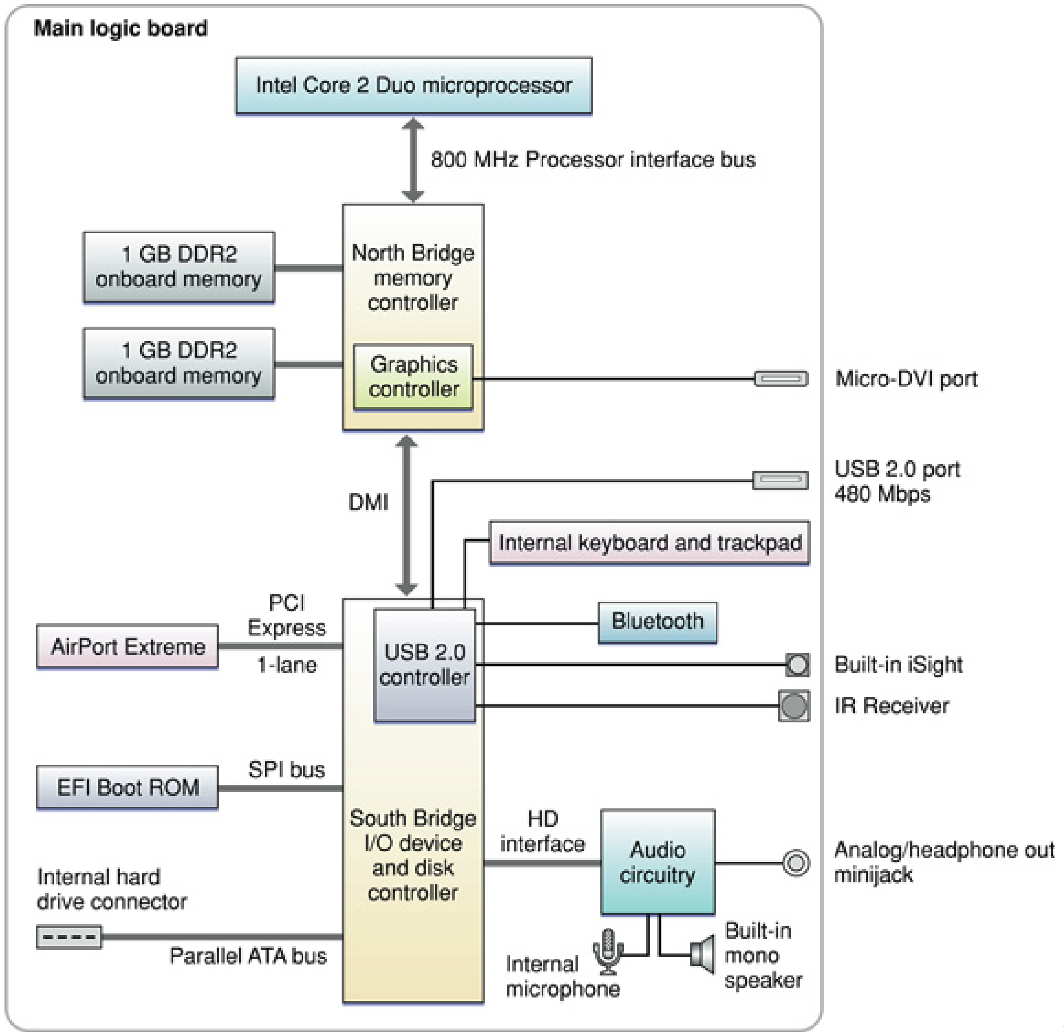
Computer Structure

A “typical” computer



Computer Structure

MacBook Air

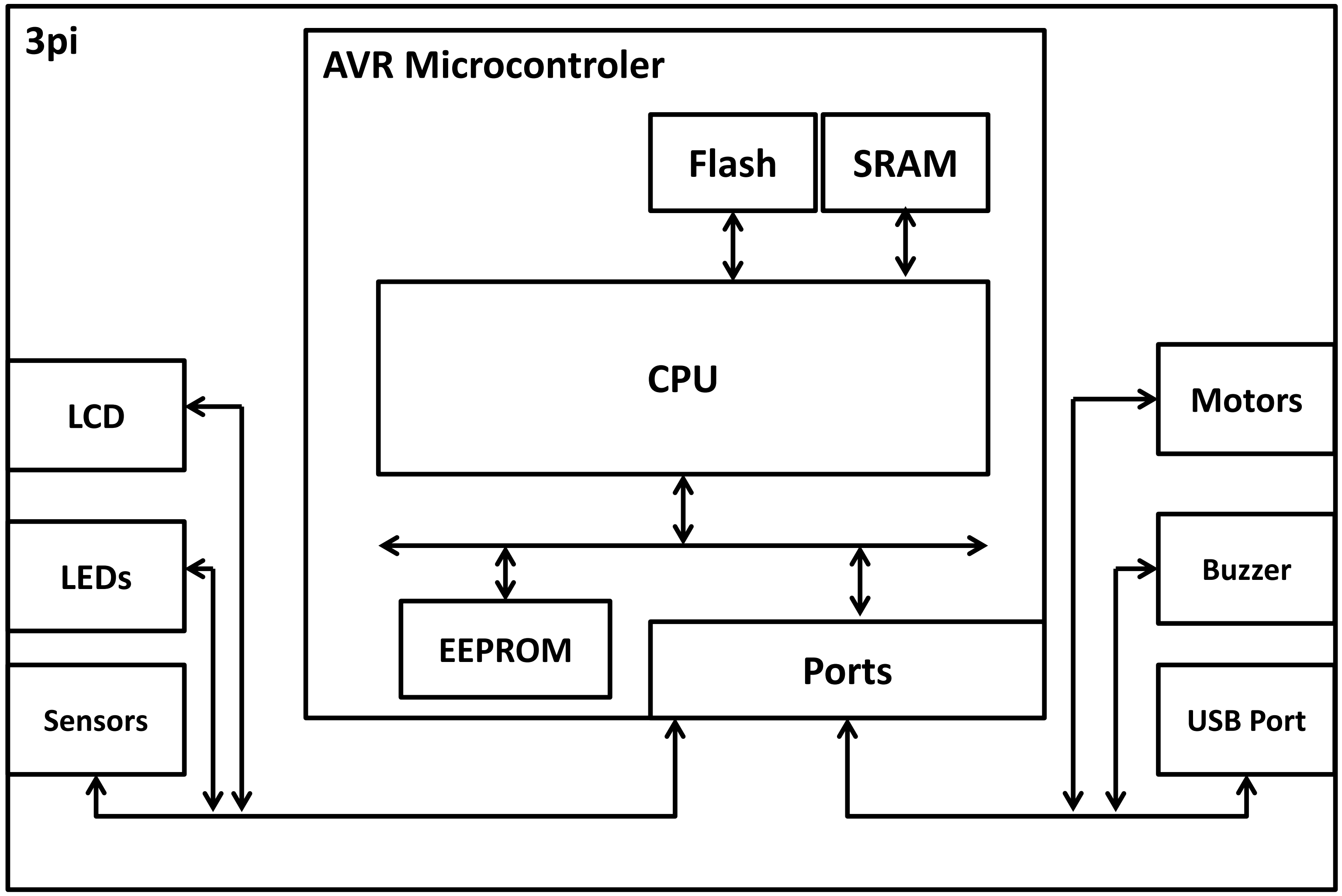


Source: www.urimagination.com/knowledge-sharing/tips-and-fixes/tips-and-fixes-for-macbook-air/macbook-air-architecture/

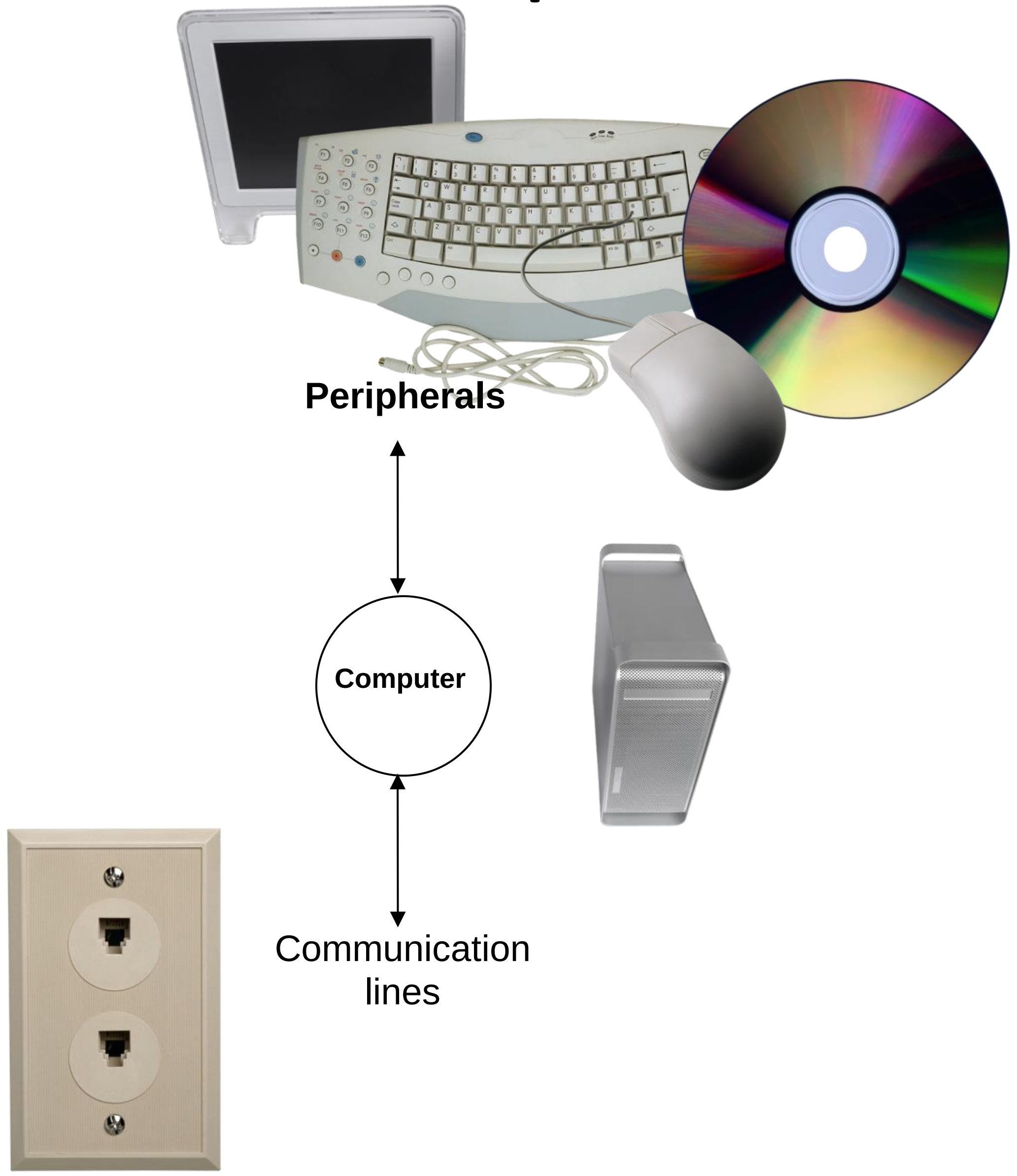
Computer Structure

Pololu 3pi

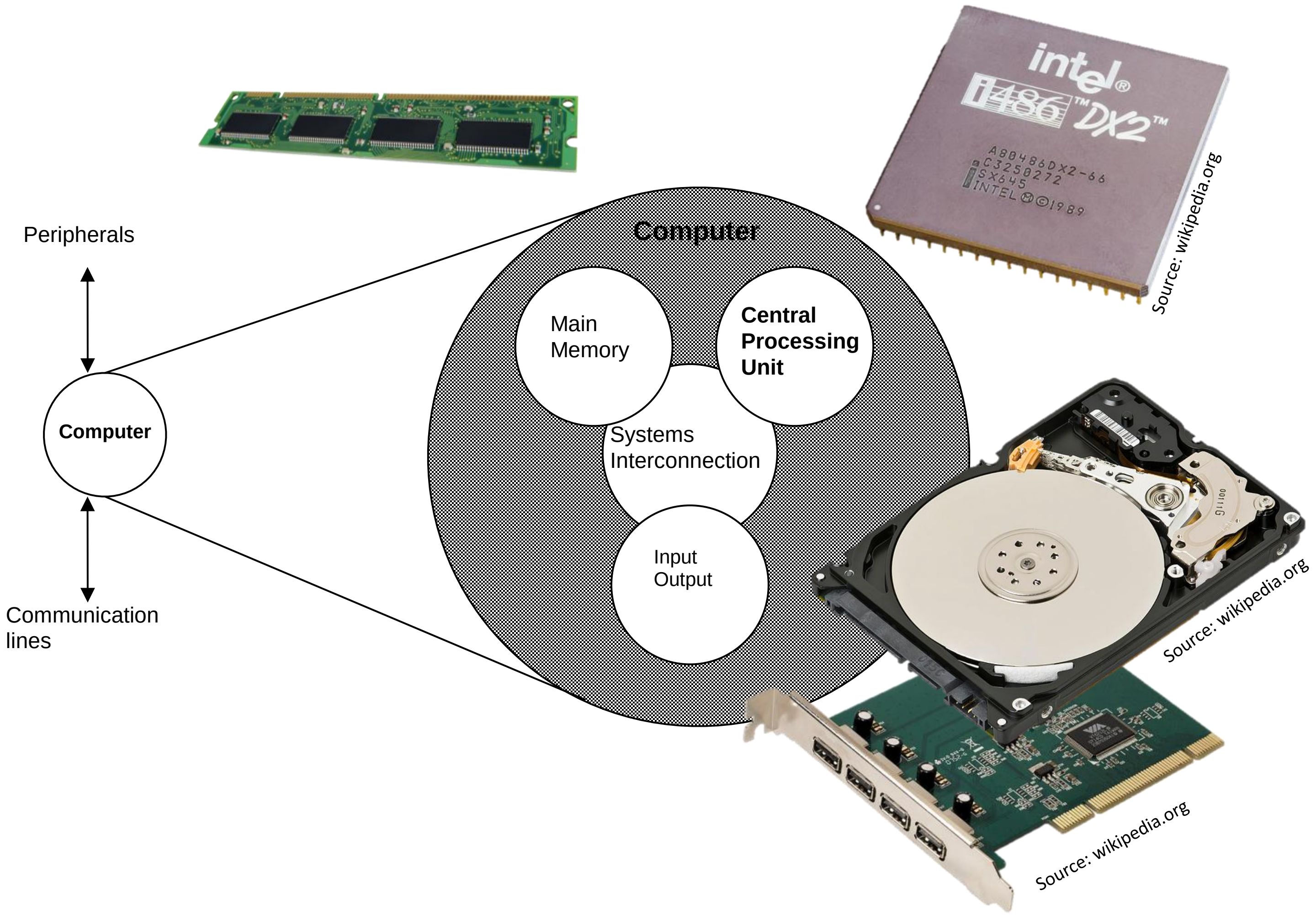
Computer Structure



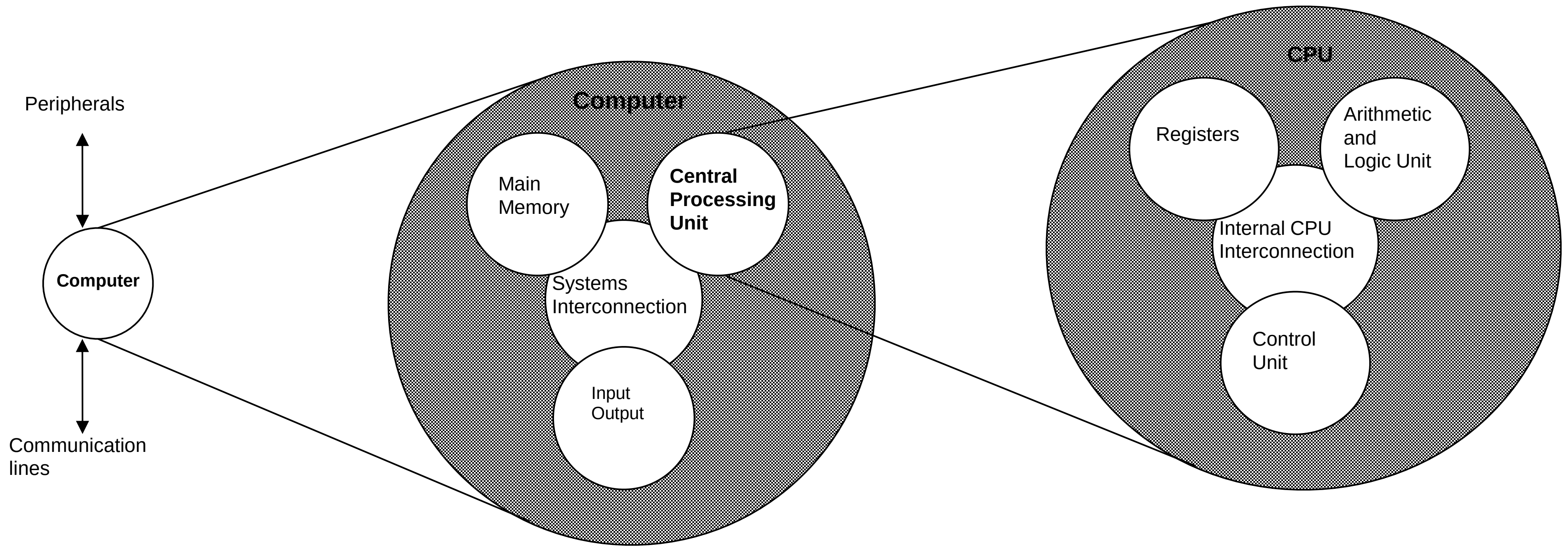
Computer Structure



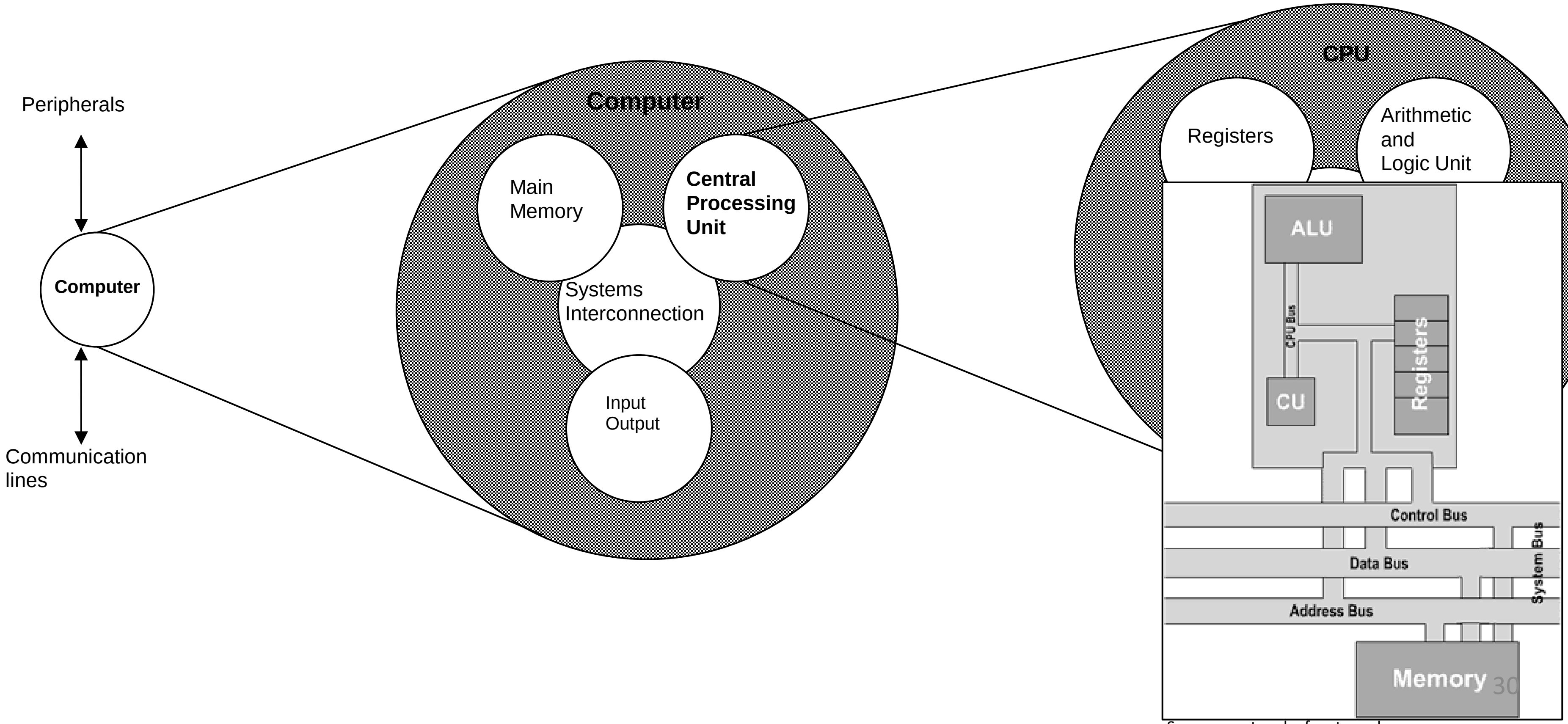
Computer Structure



Computer Structure



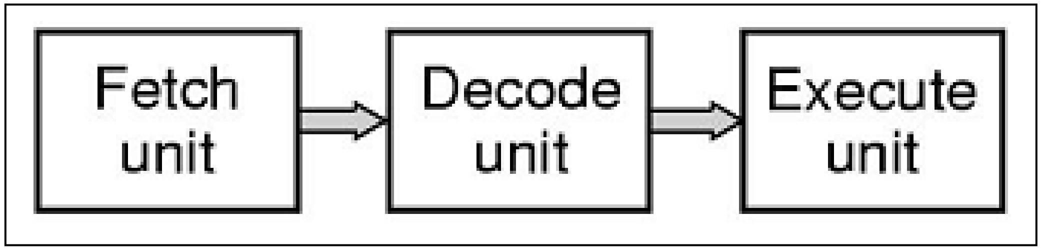
Computer Structure



Source: eastaughs.fsnet.co.uk

Processor

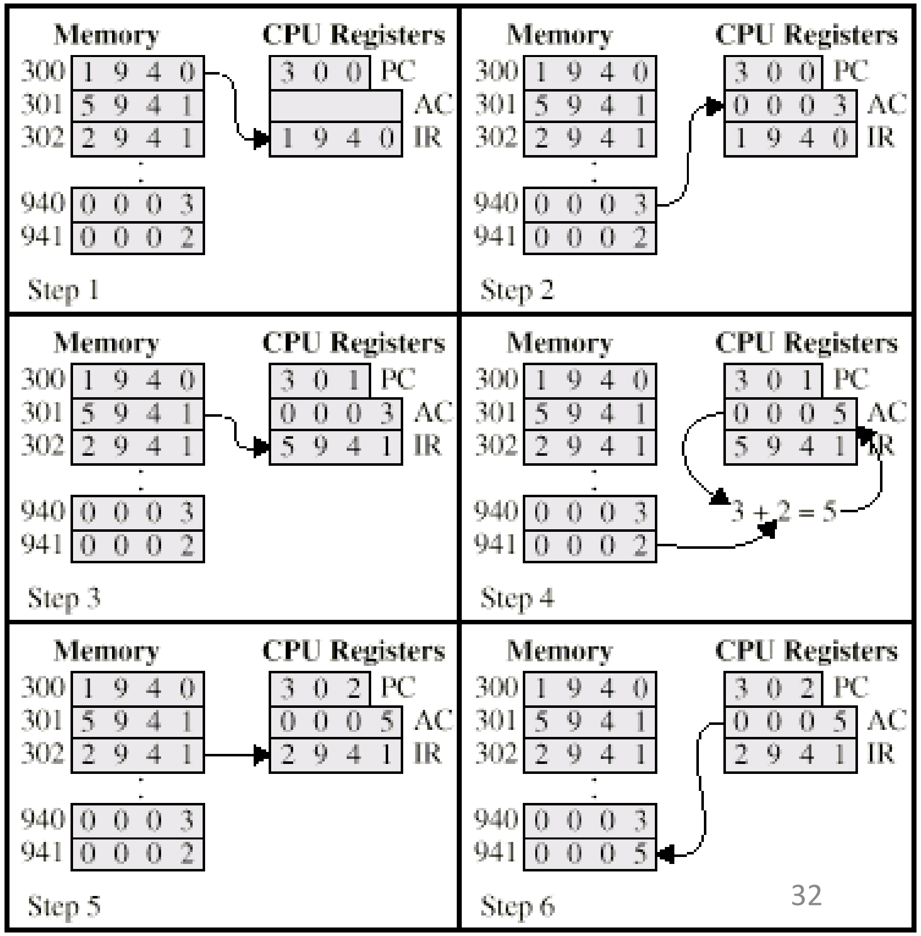
- Basic functions of a CPU:
  - Fetch an instruction
  - Decode the instruction
  - Execute the instruction



Source: Tannenbaum 2007

Processor

- Basic functions of a CPU:
  - Fetch an instruction
  - Decode the instruction
  - Execute the instruction



Processor

- Basic functions of a CPU:
  - Fetch an instruction
  - Decode the instruction
  - Execute the instruction

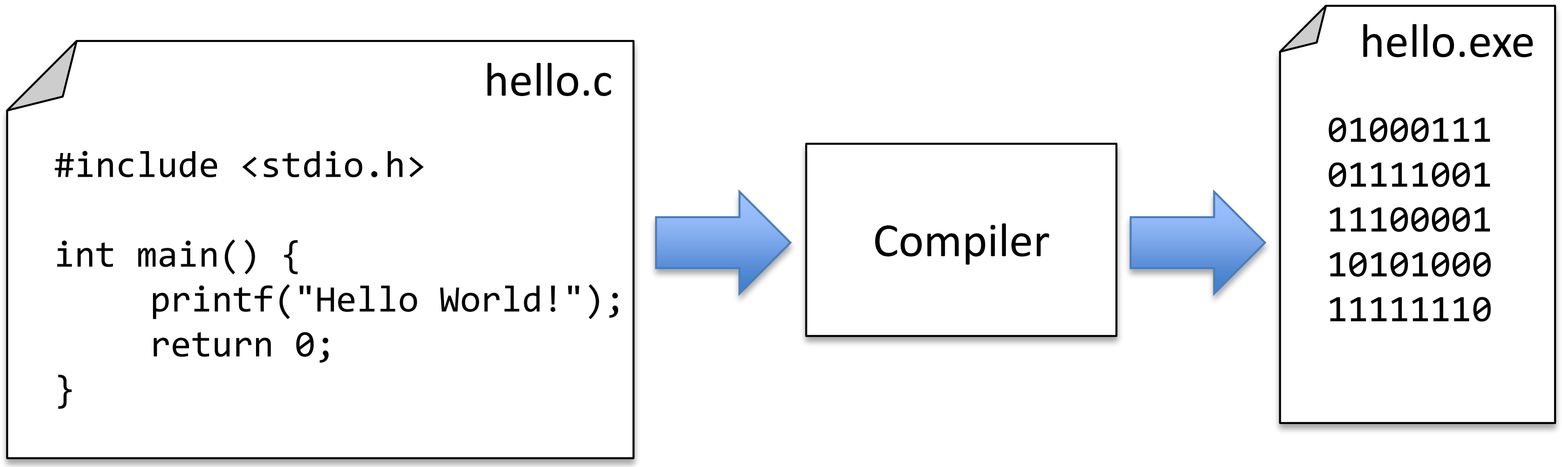
```
01010101010101010101000001110000      MOVEQ #5, R1
01010101010101010101000001100110      MOVEQ #0, R2
                                 LOOP ADDI #1,R2
                                      CMP R1, R2
                                      BNE LOOP
                                      .
                                      .
                                      .
```

Processor

Subset from 80386

Instruction	Meaning
BSF	Bit scan forward
BSR	Bit scan reverse
BT	Bit test
BTC	Bit test and complement
BTR	Bit test and reset
BTS	Bit test and set
CDQ	Convert double-word to quad-word
CMPSD	Compare string double-word
CWDE	Convert word to double-word
INSD	Input from port to string double-word
IRETx	Interrupt return; D suffix means 32-bit return, F suffix means do not generate epilogue code (i.e. LEAVE instruction)
JECXZ	Jump if ECX is zero
LFS, LGS	Load far pointer
LSS	Load stack segment
LODSD	Load string double-word
LOOPW, LOOPccW	Loop, conditional loop
LOOPD, LOOPccD	Loop while equal
MOVSD	Move string double-word
MOVSX	Move with sign-extension
MOVZX	Move with zero-extension
OUTSD	Output to port from string double-word
POPAD	Pop all double-word (32-bit) registers from stack
POPFD	Pop data into EFLAGS register
PUSHAD	Push all double-word (32-bit) registers onto stack
PUSHFD	Push EFLAGS register onto stack
SCASD	Scan string data double-word
SETcc	Set byte to one on condition, zero otherwise
SHLD	Shift left double-word
SHRD	Shift right double-word
STOSD	Store string double-word

Compiler



```
#include <stdio.h>

int main() {
     printf("Hello World!");
     return 0;
}
```

```
01000111
01111001
11100001
10101000
11111110
```

Compiler

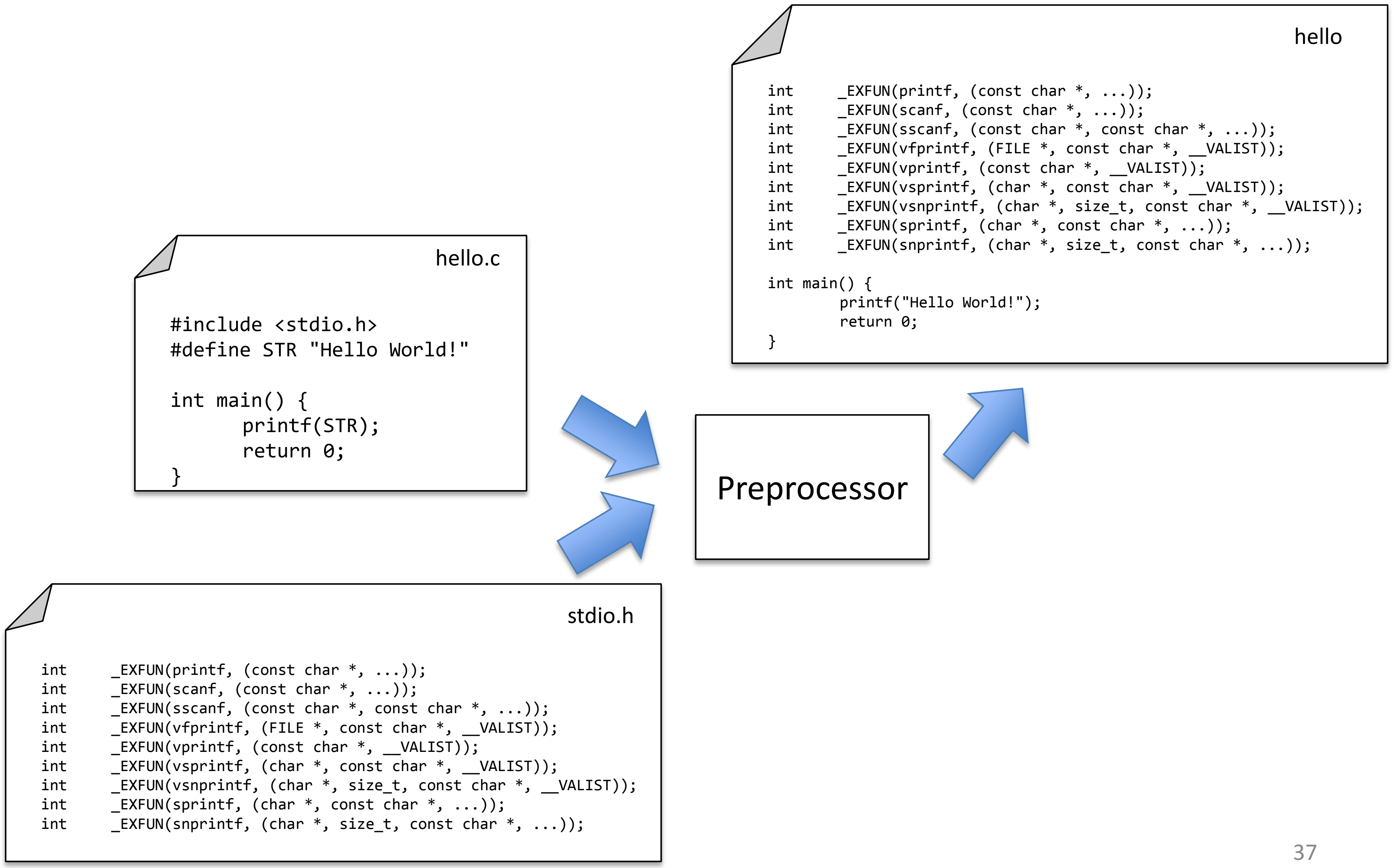
There are hundreds of C compilers

GCC

MinGW
Minimalist GNU for Windows

Atmel

Preprocessor



hello.c

```
#include <stdio.h>
#define STR "Hello World!"

int main() {
        printf(STR);
        return 0;
}
```

stdio.h

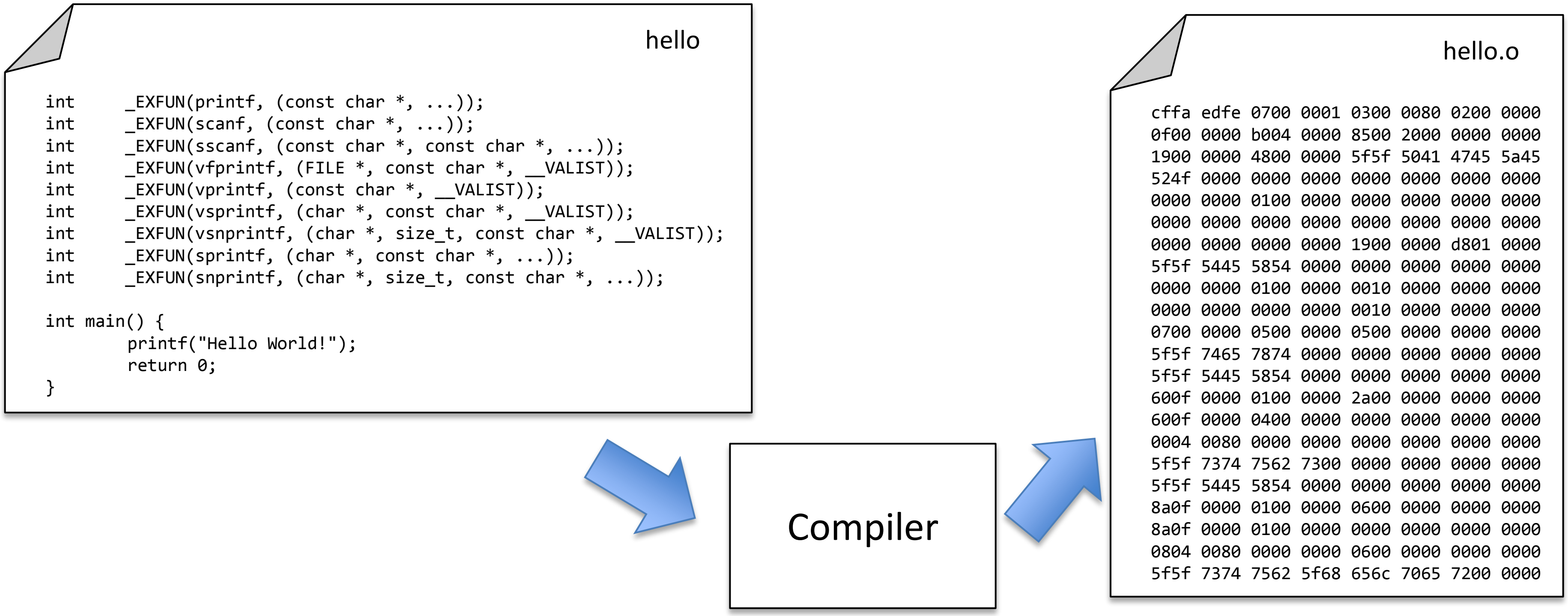
```
int     _EXFUN(printf, (const char *, ...));
int     _EXFUN(scanf, (const char *, ...));
int     _EXFUN(sscanf, (const char *, const char *, ...));
int     _EXFUN(vfprintf, (FILE *, const char *, __VALIST));
int     _EXFUN(vprintf, (const char *, __VALIST));
int     _EXFUN(vsprintf, (char *, const char *, __VALIST));
int     _EXFUN(vsnprintf, (char *, size_t, const char *, __VALIST));
int     _EXFUN(sprintf, (char *, const char *, ...));
int     _EXFUN(snprintf, (char *, size_t, const char *, ...));
```

hello

```
int     _EXFUN(printf, (const char *, ...));
int     _EXFUN(scanf, (const char *, ...));
int     _EXFUN(sscanf, (const char *, const char *, ...));
int     _EXFUN(vfprintf, (FILE *, const char *, __VALIST));
int     _EXFUN(vprintf, (const char *, __VALIST));
int     _EXFUN(vsprintf, (char *, const char *, __VALIST));
int     _EXFUN(vsnprintf, (char *, size_t, const char *, __VALIST));
int     _EXFUN(sprintf, (char *, const char *, ...));
int     _EXFUN(snprintf, (char *, size_t, const char *, ...));

int main() {
        printf("Hello World!");
        return 0;
}
```

Compiler

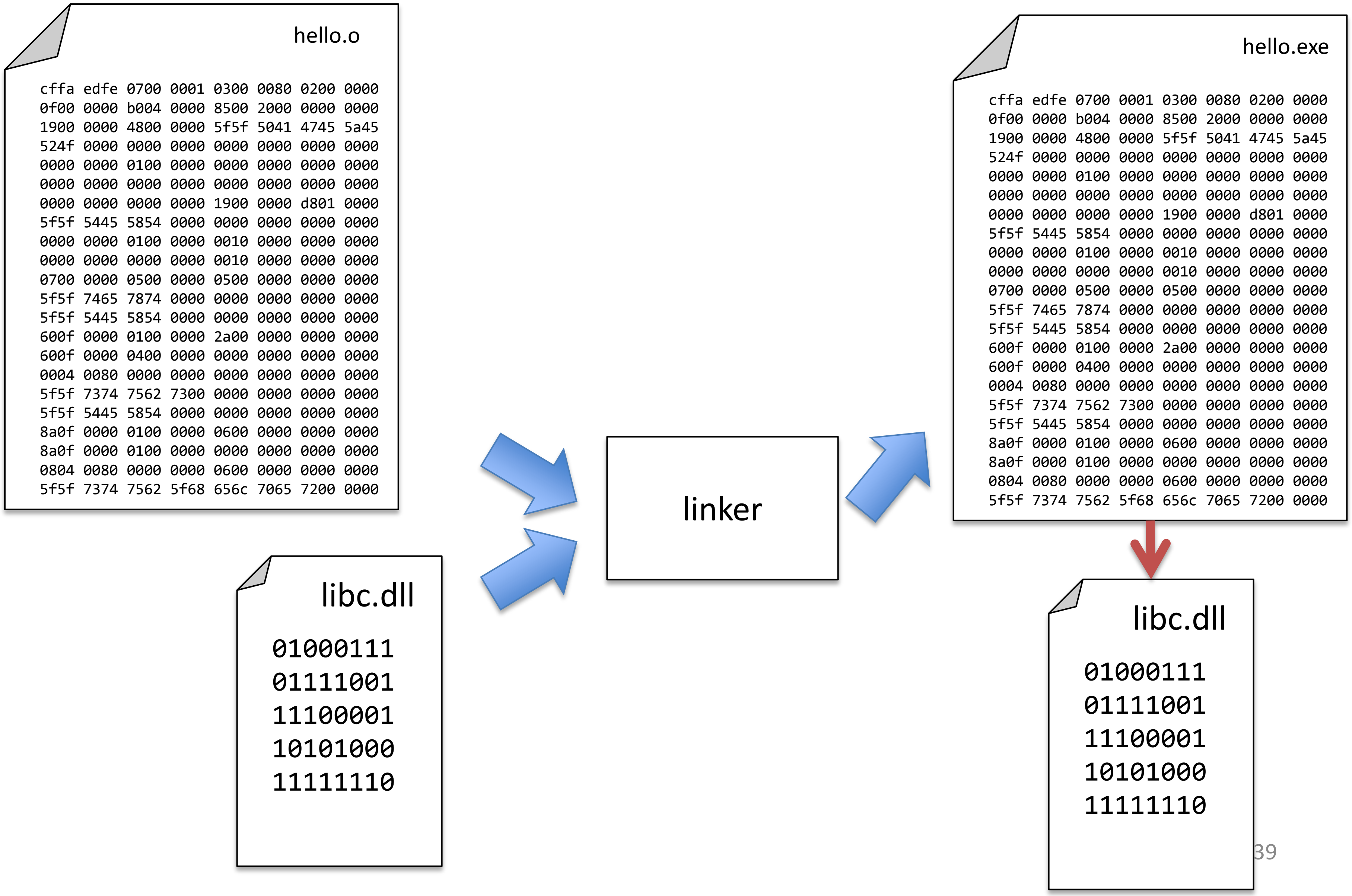


```
int     _EXFUN(printf, (const char *, ...));
int     _EXFUN(scanf, (const char *, ...));
int     _EXFUN(sscanf, (const char *, const char *, ...));
int     _EXFUN(vfprintf, (FILE *, const char *, __VALIST));
int     _EXFUN(vprintf, (const char *, __VALIST));
int     _EXFUN(vsprintf, (char *, const char *, __VALIST));
int     _EXFUN(vsnprintf, (char *, size_t, const char *, __VALIST));
int     _EXFUN(sprintf, (char *, const char *, ...));
int     _EXFUN(snprintf, (char *, size_t, const char *, ...));

int main() {
        printf("Hello World!");
        return 0;
}
```

```
cffa edfe 0700 0001 0300 0080 0200 0000
0f00 0000 b004 0000 8500 2000 0000 0000
1900 0000 4800 0000 5f5f 5041 4745 5a45
524f 0000 0000 0000 0000 0000 0000 0000
0000 0000 0100 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 1900 0000 d801 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
0000 0000 0100 0000 0010 0000 0000 0000
0000 0000 0000 0000 0010 0000 0000 0000
0700 0000 0500 0000 0500 0000 0000 0000
5f5f 7465 7874 0000 0000 0000 0000 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
600f 0000 0100 0000 2a00 0000 0000 0000
600f 0000 0400 0000 0000 0000 0000 0000
0004 0080 0000 0000 0000 0000 0000 0000
5f5f 7374 7562 7300 0000 0000 0000 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
8a0f 0000 0100 0000 0600 0000 0000 0000
8a0f 0000 0100 0000 0000 0000 0000 0000
0804 0080 0000 0000 0600 0000 0000 0000
5f5f 7374 7562 5f68 656c 7065 7200 0000
```

Linker



hello.o

```
cffa edfe 0700 0001 0300 0080 0200 0000
0f00 0000 b004 0000 8500 2000 0000 0000
1900 0000 4800 0000 5f5f 5041 4745 5a45
524f 0000 0000 0000 0000 0000 0000 0000
0000 0000 0100 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 1900 0000 d801 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
0000 0000 0100 0000 0010 0000 0000 0000
0000 0000 0000 0000 0010 0000 0000 0000
0700 0000 0500 0000 0500 0000 0000 0000
5f5f 7465 7874 0000 0000 0000 0000 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
600f 0000 0100 0000 2a00 0000 0000 0000
600f 0000 0400 0000 0000 0000 0000 0000
0004 0080 0000 0000 0000 0000 0000 0000
5f5f 7374 7562 7300 0000 0000 0000 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
8a0f 0000 0100 0000 0600 0000 0000 0000
8a0f 0000 0100 0000 0000 0000 0000 0000
0804 0080 0000 0000 0600 0000 0000 0000
5f5f 7374 7562 5f68 656c 7065 7200 0000
```

libc.dll

```
01000111
01111001
11100001
10101000
11111110
```

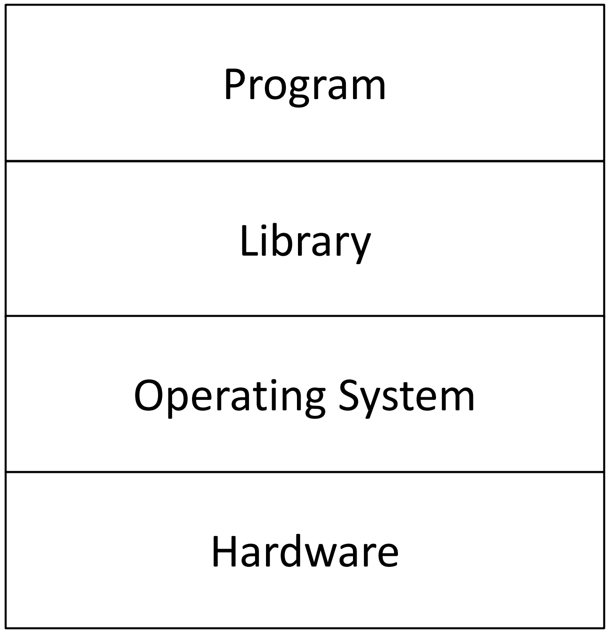
hello.exe

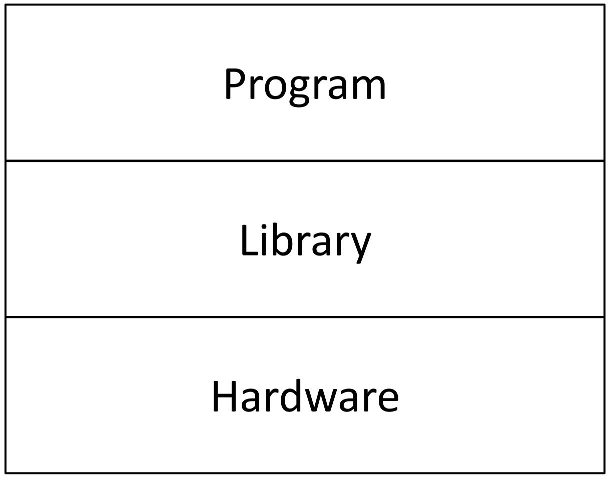
```
cffa edfe 0700 0001 0300 0080 0200 0000
0f00 0000 b004 0000 8500 2000 0000 0000
1900 0000 4800 0000 5f5f 5041 4745 5a45
524f 0000 0000 0000 0000 0000 0000 0000
0000 0000 0100 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 1900 0000 d801 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
0000 0000 0100 0000 0010 0000 0000 0000
0000 0000 0000 0000 0010 0000 0000 0000
0700 0000 0500 0000 0500 0000 0000 0000
5f5f 7465 7874 0000 0000 0000 0000 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
600f 0000 0100 0000 2a00 0000 0000 0000
600f 0000 0400 0000 0000 0000 0000 0000
0004 0080 0000 0000 0000 0000 0000 0000
5f5f 7374 7562 7300 0000 0000 0000 0000
5f5f 5445 5854 0000 0000 0000 0000 0000
8a0f 0000 0100 0000 0600 0000 0000 0000
8a0f 0000 0100 0000 0000 0000 0000 0000
0804 0080 0000 0000 0600 0000 0000 0000
5f5f 7374 7562 5f68 656c 7065 7200 0000
```

libc.dll

```
01000111
01111001
11100001
10101000
11111110
```

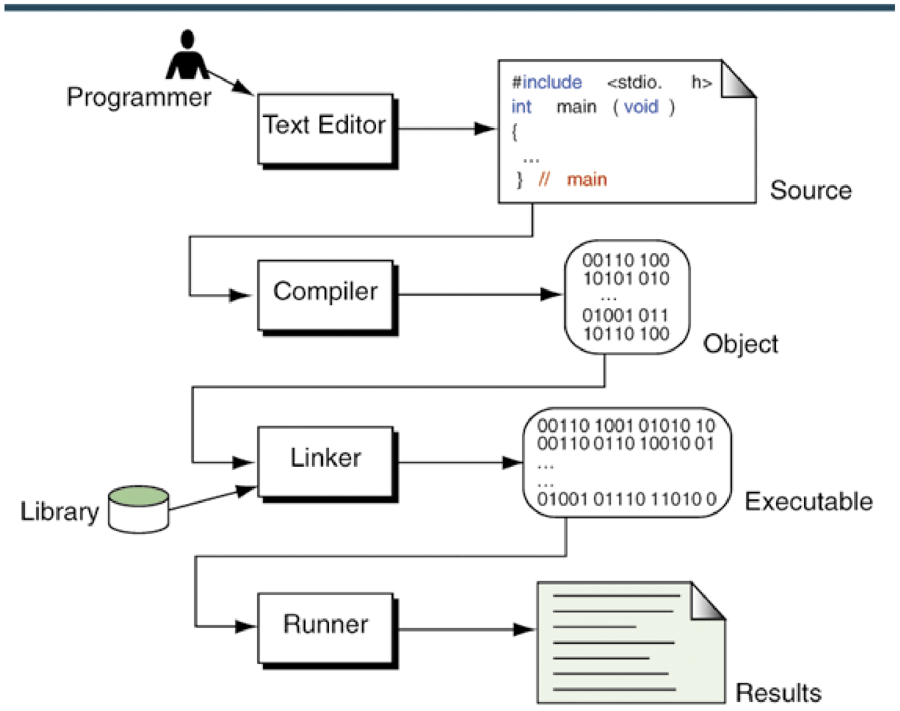
Operating Systems





Tools for programming in C

- You need a text editor to create and modify your code
  - Here you will create/modify *.c and *.h files
- You need a compiler, that is platform dependant
  - Intel/Win, Sun SPARC/Unix…
  - Here you produce object files
- You need a linker
  - Here you produce .exe files
- An environment to execute programs
  - Console
  - Command line



Integrated Development Environment (IDE)

- Source code editor
- Build automation
- Debugger
- Code completion
- Syntax highlighting
- Project browser
- Version control

eclipse

NetBeans

Visual Studio

JET BRAINS

Questions?