東北大學
Northeastern University

汇编语言程序设计

主讲： 刘松冉

单位： 东北大学计算机学院
智慧系统国际联合实验室

联系方式： liusongran@cse.neu.edu.cn

个人主页： http://faculty.neu.edu.cn/liusongran
https://liusongran.github.io/

课程概况

- 计算机科学与技术等专业学生的专业基础课
  - 计算机组成原理、接口技术、操作系统等其它课程的必要先行课
  - 对于训练学生掌握程序设计技术、熟练上机操作和程序调试技术有重要作用
- 结合一种机型讲述计算机的结构
  - 汇编语言的词法、语法及伪指令；
  - 数据的表示方法、指令系统；
  - 汇编语言程序的基本控制结构及其程序设计技巧；
  - 子程序的设计方法；输入输出及中断程序的设计方法和技巧
- 掌握汇编语言程序的基本概念、方法和技巧
- 培养阅读、分析、设计和调试汇编语言的能力

课程团队

信俊昌 教授

柳秀梅 副教授

陈景柱 高级实验师

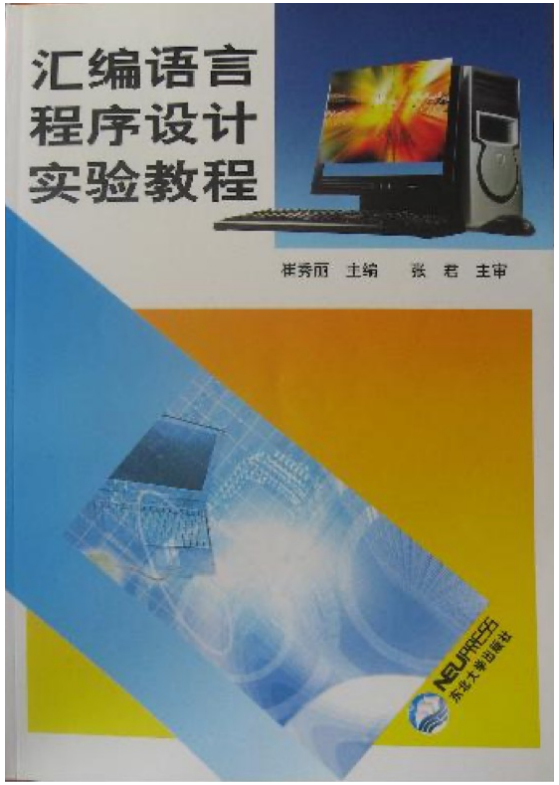
刘 莹 讲师

刘松冉 讲师

刘思宇 助理实验师

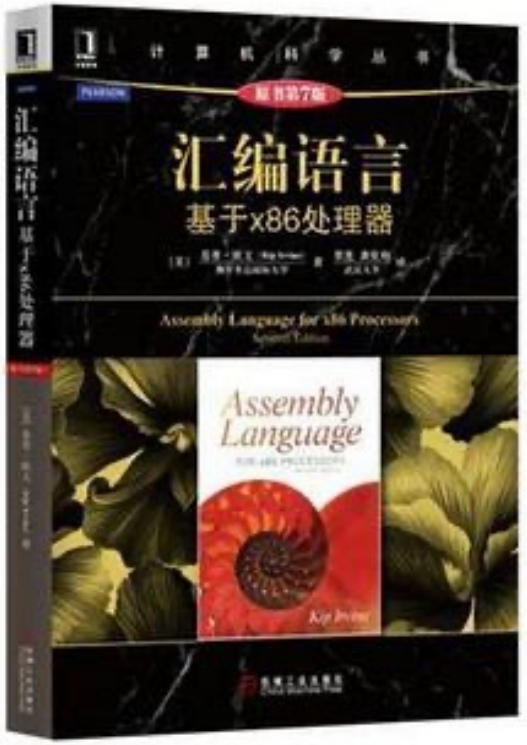
教材及参考书目

《汇编语言程序设计》第五版
高福祥、张君主编，东北大学出版社出版



《汇编语言程序设计实验教程》
崔秀丽主编 东北大学出版社出版

《汇编语言程序设计习题集》
张君主编 东北大学出版社出版



《汇编语言：基于x86处理器》
[美] 基普 · 欧文 机械工业出版社

教材主体内容

第1章 绪论
第2章 计算机运算基础
第3章 微型处理机的结构
第4章 汇编语言
第5章 顺序结构程序
第6章 分支结构程序
第7章 循环结构程序
第8章 子程序设计
第9章 条件汇编与宏指令
第10章 算术运算与代码转换
第11章 列表与字符串操作
第12章 输入输出与中断

课程考核及占比

- **授课 48学时**
  - 理论 32学时
  - 实验 16学时

- **考核 100分**
  - 平时 30分 （随堂测试+课后作业）
  - 上机 20分
  - 考试 50分

第一章 概述

一. 微型计算机的发展历程

二. 微机的主要特点

三. 计算机系统

四. 计算机语言

一. 微型计算机的发展历程

第一台通用计算机

- 1945年， ENIAC
- 用于计算弹道
- 美国宾夕法尼亚大学莫尔电工学院制造
- 体积庞大
  - 占地170多平方米
  - 重量约30吨
  - 近100千瓦的电力

一. 微型计算机的发展历程

计算机的演进

时代	年份	电子器件	软件	应用
一	46—58	电子管	机器语言 汇编语言	科学计算
二	58—64	晶体管	高级语言	数据处理 工业控制
三	64—71	集成电路	操作系统	文字处理 图形处理
四	71~至今	（超）大规模集成电路	数据库、网络等	社会各个领域

微机时代

一. 微型计算机的发展历程

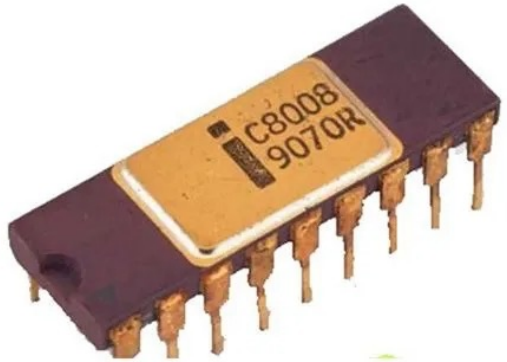
第1代微机(1971-1973)

- 4位微处理器Intel 4004
  - 1971年推出
  - 16只引脚
  - 2250个晶体管
  - 最高频率740kHz
  - 每秒运算6万次(108kHz)
  - 能执行4位运算
  - 支持8位指令集及12位地址集
  - 日本公司Busicom用来生产计算器
- 微型计算机MCS-4
  - 计算性能远ENIAC
  - 最初售价为200美元

一. 微型计算机的发展历程

第1代微机(1971-1973)

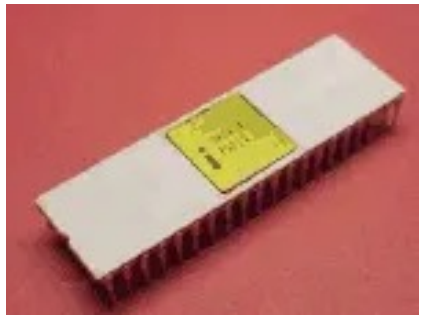
- 8位微处理器Intel 8008
  - 1972年推出
  - 可以支持到16KB的内存
  - 8位运算+14位地址总线
  - 两种速度
    - 0.5 MHz
    - 0.8 MHz
  - 采用PMOS，仍属第一代微处理器



一. 微型计算机的发展历程

第2代微机(1974-1977)

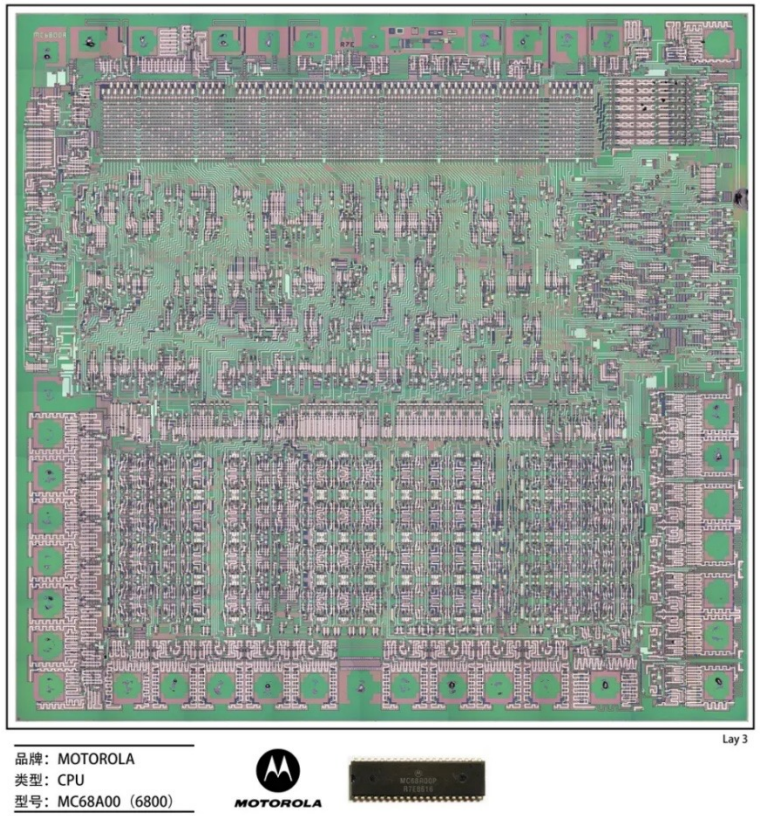
- 8位微处理器Intel 8080
  - 1974年推出
  - 2MHz 和3MHz的频率
  - 8 位数据总线
  - 16 位地址总线
  - 6000个晶体管
  - 64KB 可寻址内存
  - 40引脚
  - 0.64 MIPS 运算能力
  - 采用NMOS，第2代微处理器
  - 同8008相同的汇编语言



一. 微型计算机的发展历程

第2代微机(1974-1977)

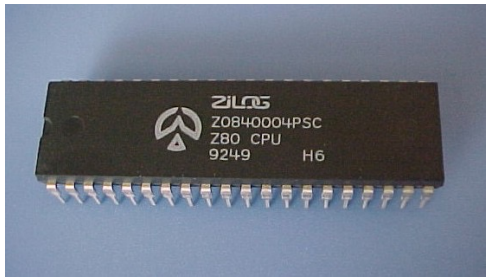
- 8位微处理器Motorola MC6800
  - 1974年推出
  - 摩托罗拉第一款
  - 苹果Lisa首次采用Motorola6800
  - 84年，苹果的Macintosh机型，继续沿用Motorola MC68000，只不过速度提升到了8Mhz
  - 同8080相比，6800处理指令需要的指令周期更少
  - 2MHz的8080处理能力尚且不如1MHz的6800
  - 最低工作频率可以降到100KHz



一. 微型计算机的发展历程

第2代微机(1974-1977)

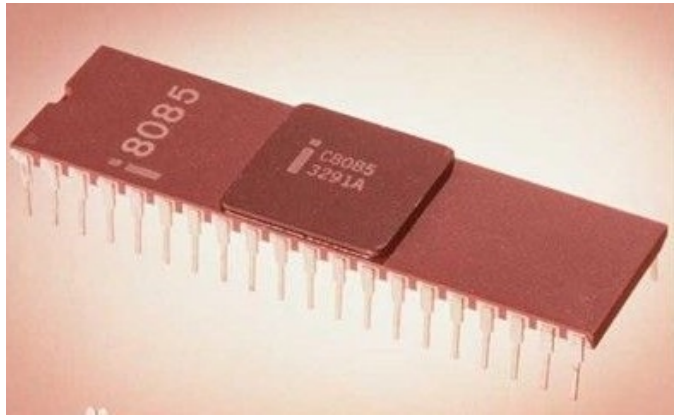
- 8位微处理器Zilog Z80
  - 1976年推出
  - 2MHz ~8MHz的频率
  - 8 位数据总线
  - 16 位地址总线
  - 64KB 可寻址内存
  - 40引脚
  - 与8080兼容的汇编语言
  - 扩充了指令系统
  - Zilog公司的创始人费根原本是英特尔公司从4004到8080芯片开发的功臣



一. 微型计算机的发展历程

第2代微机(1974-1977)

- 8位微处理器Intel 8085
  - 1976年推出
  - 基本与8080相同
  - 单一5V供电(以前要12V)
  - 增加了几条指令，使Z-80不能兼容它了

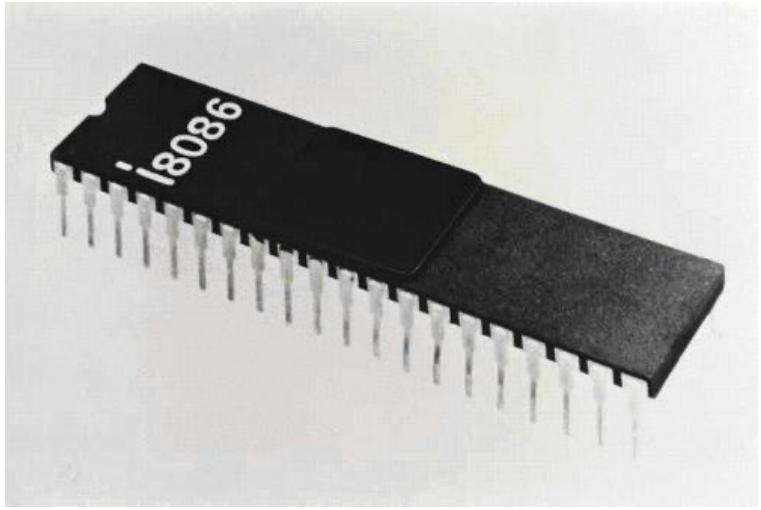


当时，Zilog、Motorola和Intel在微处理器领域三足鼎立

一. 微型计算机的发展历程

第3代微机(1978-1984)

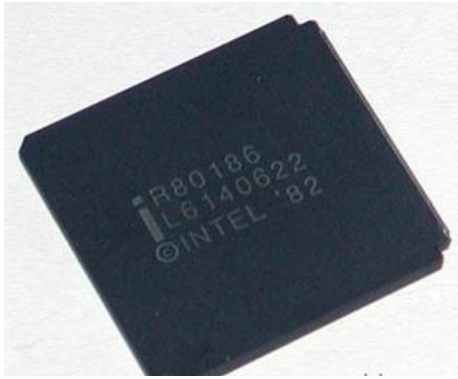
- 16位微处理器Intel 8086
  - 1978年推出
  - 5、8、10MHz的频率
  - 16 位数据总线
  - 20 位地址总线
  - 2.8万个晶体管
  - 1MB 可寻址内存
  - 40引脚
  - 标准的X86指令集



一. 微型计算机的发展历程

第3代微机(1978-1984)

- 16位微处理器Intel 8088
  - 1979年推出
  - 基本与8086相同
  - 外部数据总线变为8位
  - 内部仍为16位
- 16位微处理器Intel 80186/188
  - 1980年推出，针对工业控制/通信等嵌入式市场
  - 8086/8088内核
  - 包括中断控制器、定时器、DMA、I/O、UART、片选电路等外设



一. 微型计算机的发展历程

第3代微机(1978-1984)

- 16位微处理器Intel 80286
  - 1982年推出
  - 20MHz的频率
  - 16 位数据总线
  - 24 位地址总线
  - 13万个晶体管
  - 16MB 可寻址内存
  - 两种工作模式
    - 实模式
    - 保护模式

一. 微型计算机的发展历程

第4代微机(1985-1992)

- 32位微处理器Intel 80386-DX
  - 1985年推出
  - 最高33MHz的频率
  - 32 位数据总线
  - 32 位地址总线
  - 27.5万个晶体管
  - 4GB 可寻址内存
  - 三种工作模式
    - 实模式
    - 保护模式
    - 虚拟86模式
- 32位微处理器80386-SX
  - 1988年推出
  - 内部与386-DX相同
  - 外部接口采用16位数据总线

一. 微型计算机的发展历程

第4代微机(1985-1992)

- 32位微处理器Intel 80486
  - 1989年推出
  - 最高25~50MHz的频率
  - 32 位数据总线
  - 32 位地址总线
  - 120万个晶体管
  - 4GB 可寻址内存
  - 386CPU+387CPU+Cache
  - 采用了倍频技术
- 32位微处理器Intel 80486SX
  - 1991年推出
  - 不带387的80486

一. 微型计算机的发展历程

第5代微机(1993-1995)

- 32位微处理器Intel Pentium(奔腾)
  - 1993年推出
  - 60、66MHz的频率
  - 310万个晶体管
  - 超标量流水线设计
  - 每秒可执行10000 万条以上(112MIPS)的指令
  - 迎合了用户在图形图像、实时图像处理、语音识别和CAD/CAM等方面对高性能工作平台的需求

一. 微型计算机的发展历程

第6、7代微机(1995末-)

- 32位微处理器Pentium Pro(高能奔腾)
  - 1995年末推出
  - 二级Cache
    - 有256KB或512KB
    - 最大有1MB的二级Cache
  - 工作频率有
    - 133/66MHz(工程样品)
    - 150/60MHz
    - 166/66MHz
    - 180/60MHz
    - 200/66MHz

一. 微型计算机的发展历程

第6、7代微机(1995末-)

- Pentium MMX
  - 1996年推出
  - 增加了MMX指令集
- Intel Pentium II
  - 1997年发布
  - 更改了封装形式
- Intel Celeron
  - 1998年发布
  - 取消或减少了二级Cache

一. 微型计算机的发展历程

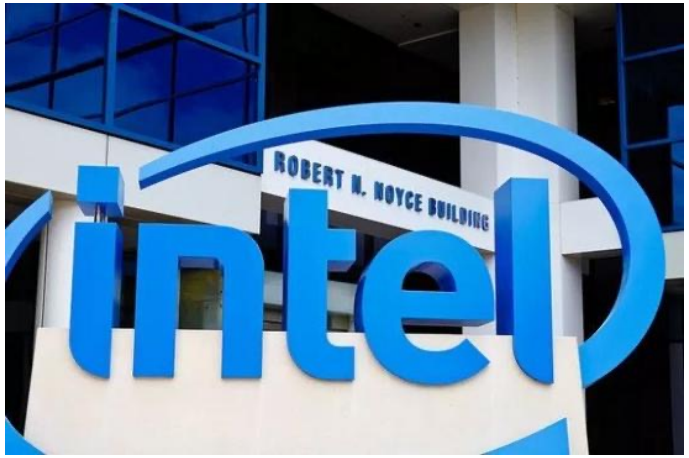
第6、7代微机(1995末-)

- Intel Pentium III
  - 1999年发布
  - 733MHz~1GHz
- Intel奔腾IV
  - 2000年发布
  - 全新架构
  - 提高了频率
  - 32位与64位两种
  - 多核

- 2022年初，新一代的奔腾处理器采用了与 12 代酷睿一样的 Intel 7 工艺，但没有大小核架构。
- 奔腾 G7400 为 2 核 4 线程，3.7GHz，6MB 三级缓存，46W TDP，支持 DDR4-3200 内存和 DDR5-4800 内存。核显为 UHD 710，16 EU 1.35GHz

一. 微型计算机的发展历程

为什么学习8086/8088的汇编语言？



Intel历代微处理器产品型号（Intel 芯片50年）：

Intel 4004
Intel 8008
Intel 8080
Intel 8086/8088
Intel 80186/80188
Intel 80286
Intel 80386(DX/SX)
Intel 80486 (DX/SX)
Intel Pentium (MMX)
Intel PII (Celeron)
Intel PIII(Celeron II)
Intel P4
…

自8086起，指令集向后兼容

第一章 概述

一. 微型计算机的发展历程

二. 微机的主要特点

三. 计算机系统

四. 计算机语言

二.微机的主要特点

- 体积小， 重量轻
- 简单灵活， 可靠性高， 功耗低
- 对工作条件要求低
- 性能价格比高

第一章 概述

一. 微型计算机的发展历程

二. 微机的主要特点

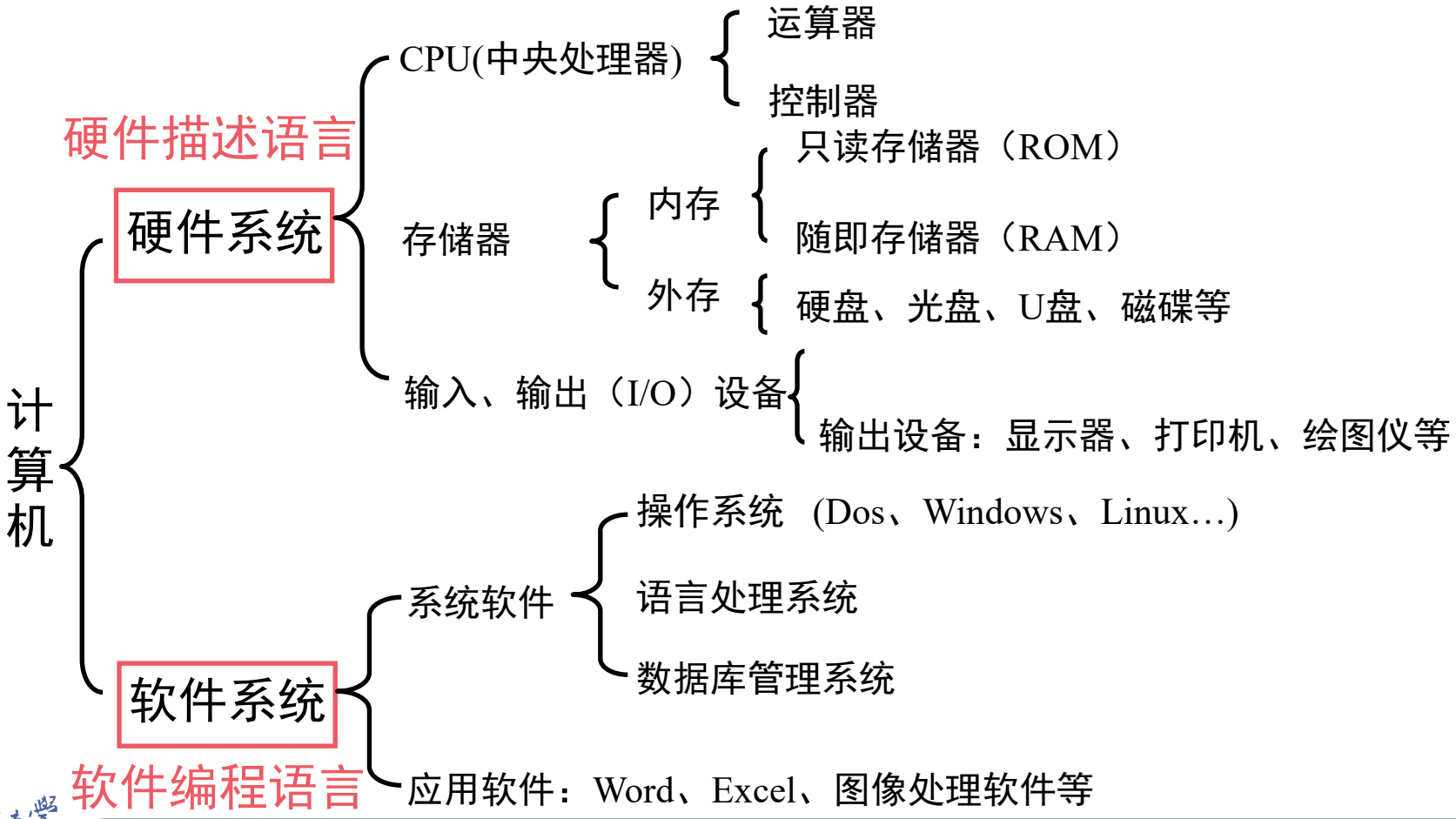
三. 计算机系统

四. 计算机语言

三. 计算机系统

计算机系统由硬件子系统和软件子系统组成。

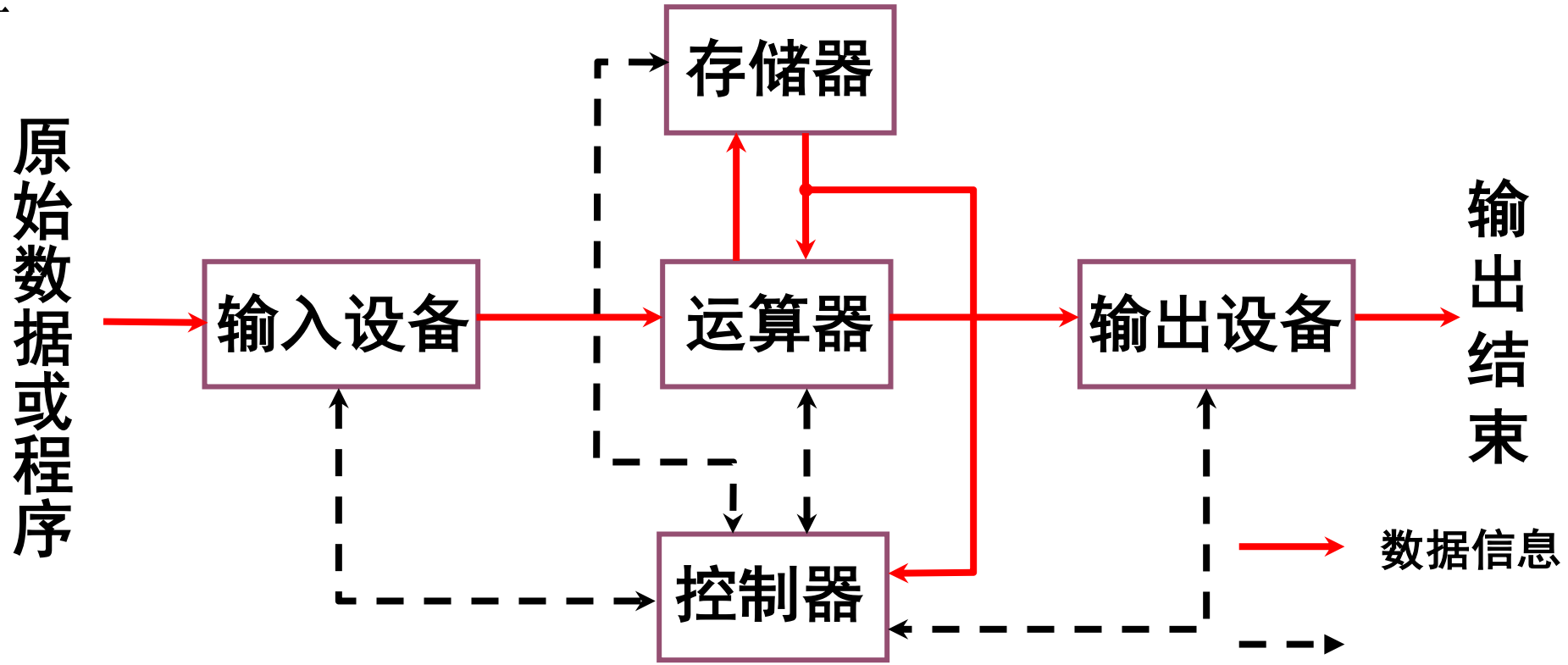
- 系统：相关事物组成的集合称为系统，组成系统的事物称为该系统的元件或部件



三. 计算机系统

计算机系统由硬件子系统和软件子系统组成。

- 硬件子系统：组成计算机系统的所有电子的、机械的、光学的和磁性的元部件



冯·诺依曼计算机硬件框图

三. 计算机系统

硬件子系统

打印机

显示器

键盘

鼠标

外设 是人机交换信息的通道

主机

是计算机的核心设备，决定着计算机的整机性能。

硬件子系统

主机箱内部参考图

主机中的主要部件：

（1）**中央处理器（简称CPU）**

是组成计算机的核心部件，相当于计算机的“大脑”。

（2）内存条（简称**内存**）

（3）驱动器

软盘驱动器（软驱）

硬盘和硬盘驱动器（硬盘）

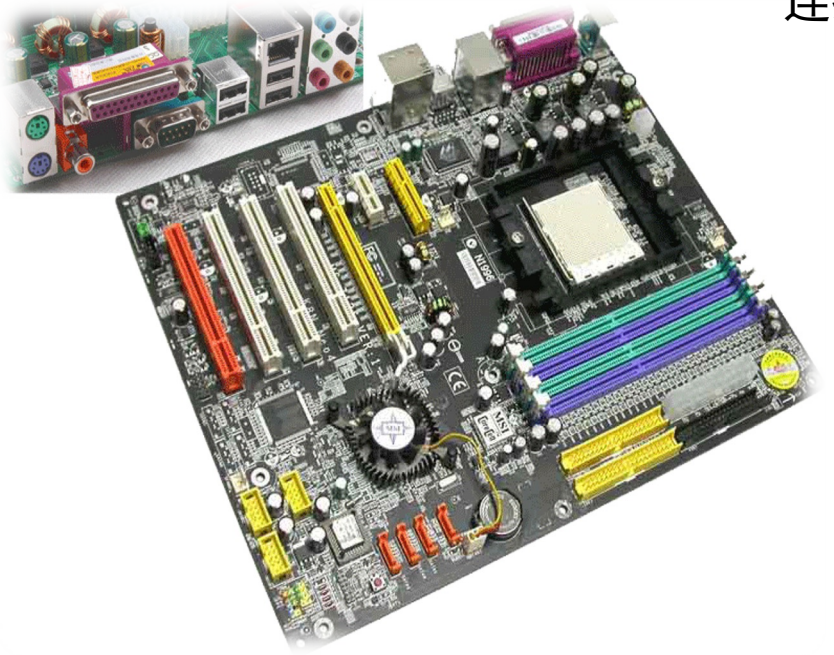
光盘驱动器（光驱）

它是计算机的重要储存设备。

硬件子系统

（4）主板

相当于计算机的“身体”，用于连接计算机的各个部件



（5）显卡

也叫显示适配器，用于 主机和显示器的连接。

（6）声卡

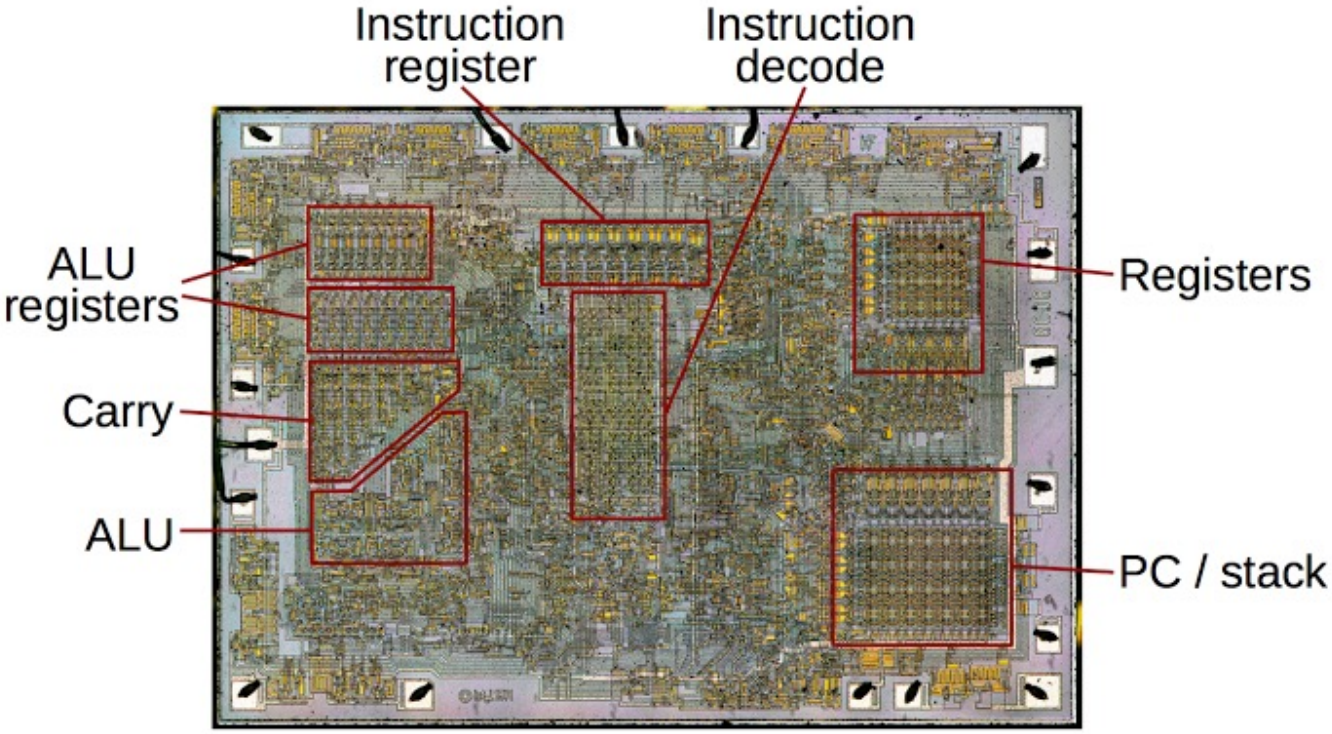
专门处理音频信号的输入输出。

（7）网卡

也叫网络适配器，用于计算机之间的互联。

硬件子系统

- 中央处理器CPU
  - 算术逻辑部件：完成算术和逻辑运算
  - 通用寄存器组：由若干寄存器组成
  - 标志寄存器：用于保存运算结果的状态和CPU用的逻辑控制标志
  - 控制部件：用于保存指令，对指令进行 译码，产生控制各个部件的信号



Die photo of 8086[1]

[1] https://www.quora.com/Which-is-the-best-way-to-study-microprocessor-8086

硬件子系统

- 中央处理器CPU
- 存储装置
  - 内存：又称主存储器，用于存储计算机当前正在运行的程序，正在处理的原始数据、中间结果及最终结果等。直接寻址。
  - 外存：又称为辅助存储器，如磁盘、磁带、 光盘等。用于存放计算机当前没有执行的一 些系统程序和用户程序。

HUAWEI P50 Pro

存储

运行内存 + 机身内存

8 GB RAM + 128/256/512 GB ROM

12 GB RAM + 512 GB ROM

*可使用的内存容量小于此值，因为手机软件占用部分空间。

扩展存储

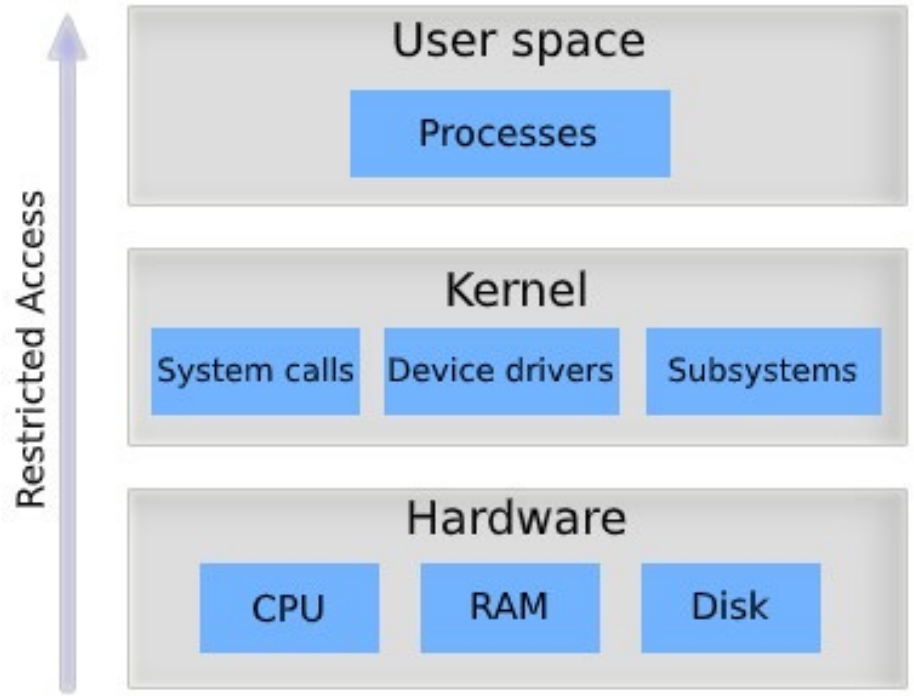
NM 存储卡，最大支持 256 GB

[1] https://consumer.huawei.com/cn/phones/p50-pro/specs/

三. 计算机系统

计算机系统由硬件子系统和软件子系统组成。

- 软件子系统：是指为了充分发挥计算机硬件子系统的功能，方便用户使用计算机，提高计算机系统效率而编制的各种程序。由系统软件和应用软件组成。
  - 系统软件：
    - 用户应用程序接口（或称中间件Middleware等）
    - 硬件驱动程序等
    - 操作系统代码（Operating System，OS）
      - ✓ DOS, Window xp/10/11, MacOS, Linux, RTOS等
  - 用户层：应用程序
    - 文字处理：Word
    - 科学计算：MATLAB
    - 图像处理：Photoshop，PR，AE
    - …



Linux components[1]

[1] https://patchthenet.com/tutorials/linux/getting-started-with-linux/chapter-1-introduction-to-linux/

第一章 概述

一. 微型计算机的发展历程

二. 微机的主要特点

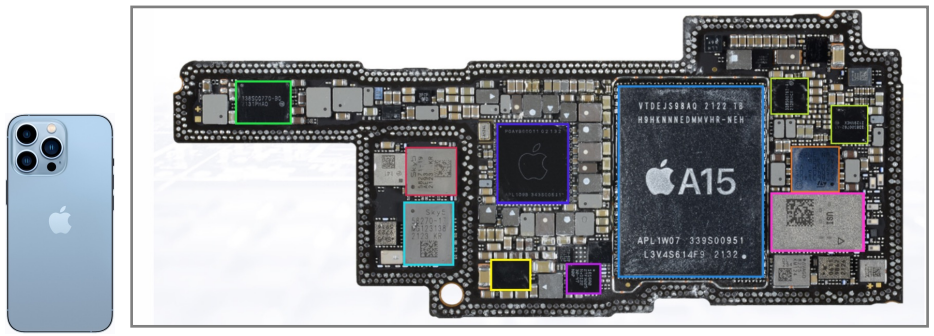
三. 计算机系统

四. 计算机语言

四. 计算机语言

计算机语言是人与计算机之间交流信息的工具。

- 程序：用计算机语言描述的处理步骤
- 程序设计：编制处理步骤的过程



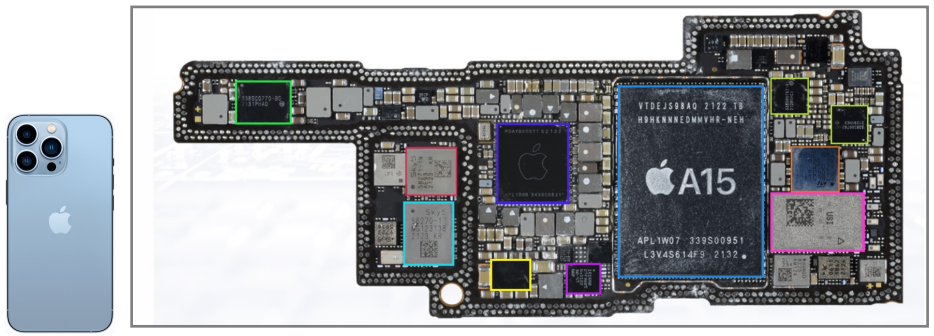
计算机硬件系统结构图[1]

[1] https://www.techinsights.com/blog/teardown/apple-iphone-13-pro-teardown

四. 计算机语言

计算机语言是人与计算机之间交流信息的工具。

- 程序：用计算机语言描述的处理步骤
- 程序设计：编制处理步骤的过程



计算机硬件系统结构图[1]

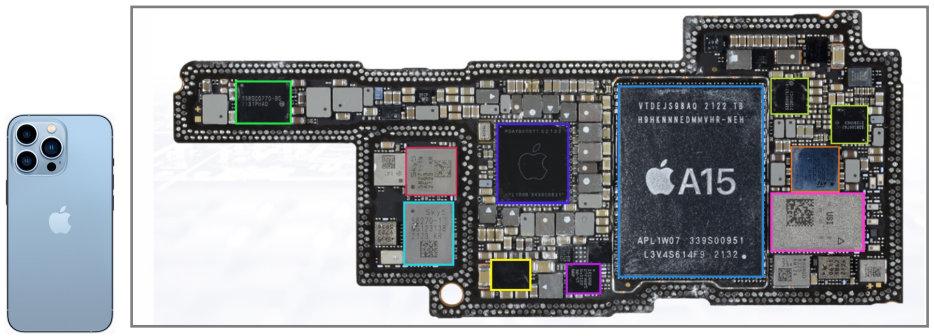
```
TERMINAL    PROBLEMS    OUTPUT    DEB

❯ gcc hello_world.c -o hello_world
❯ ./hello_world
Hello World!
```

[1] https://www.techinsights.com/blog/teardown/apple-iphone-13-pro-teardown

四. 计算机语言

计算机语言是人与计算机之间交流信息的工具。

- 程序：用计算机语言描述的处理步骤
- 程序设计：编制处理步骤的过程
- 选择编程语言（计算机语言）？



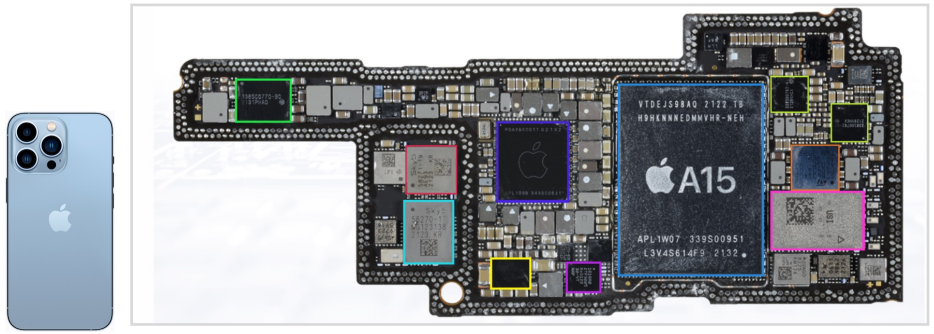
计算机硬件系统结构图[1]

```
TERMINAL    PROBLEMS    OUTPUT    DEB

❯ gcc hello_world.c -o hello_world
❯ ./hello_world
Hello World!
```


[1] https://www.techinsights.com/blog/teardown/apple-iphone-13-pro-teardown

四. 计算机语言

计算机语言是人与计算机之间交流信息的工具。



计算机硬件系统结构图[1]

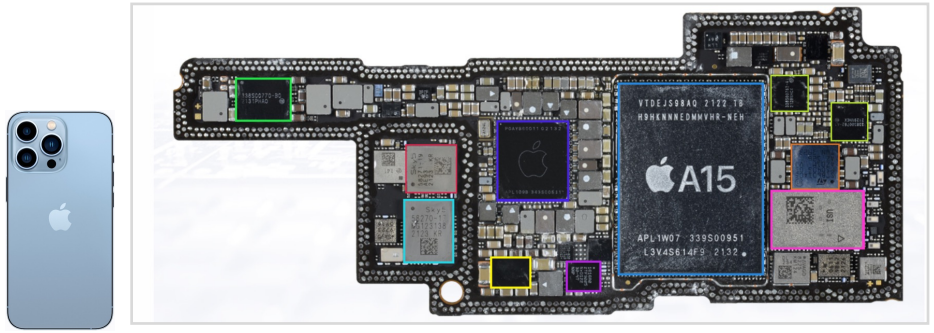
```
2    0000 0001
3    0000 0000
4    1000 1010    ; 2号单元内容取入AH寄存器
5    0010 0110
6    0000 0010
7    0000 0000
8    0000 0000    ; AL的内容与AH的内容相加,结
9    1110 0000    ; 果存入AL中
10   1010 0010    ; AL的内容送入3号单元
11   0000 0011
12   0000 0000
```

机器语言（machine code）

[1] https://www.techinsights.com/blog/teardown/apple-iphone-13-pro-teardown

四. 计算机语言

计算机语言是人与计算机之间交流信息的工具。



计算机硬件系统结构图[1]

Assembly for ARM

```
9       mov x29, sp
10      .cfi_def_cfa w29, 16
11      .cfi_offset w30, -8
12      .cfi_offset w29, -16
13      adrp    x0, l_.str@PAGE
14      add x0, x0, l_.str@PAGEOFF
15      bl  _printf
16      mov w0, #0
17      ldp x29, x30, [sp], #16             ; 16-byte Folded Reload
18      ret
19      .cfi_endproc
20                                          ; -- End function
21      .section    __TEXT,__cstring,cstring_literals
22  l_.str:                                 ; @.str
23      .asciz  "Hello World!\r\n"
```

汇编语言（Assembly）

[1] https://www.techinsights.com/blog/teardown/apple-iphone-13-pro-teardown

四. 计算机语言

计算机语言是人与计算机之间交流信息的工具。

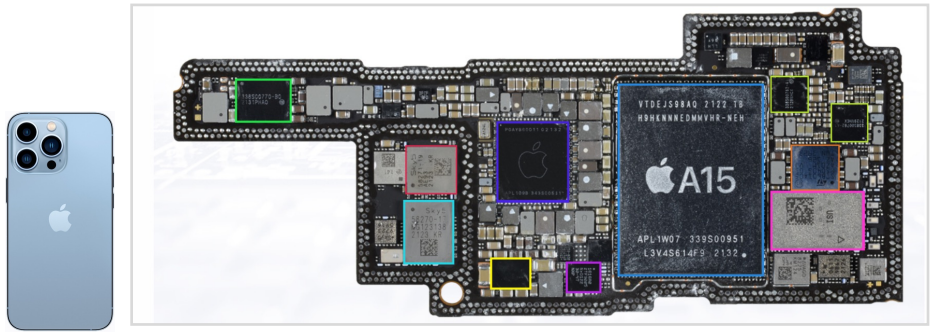
C

```c
#include <stdio.h>

int main(){
    printf("Hello World!\r\n");
}
```

Python

```python
x = 1
if x == 1:
    print("hello world!")

```

高级语言（通用语言）



计算机硬件系统结构图[1]

[1] https://www.techinsights.com/blog/teardown/apple-iphone-13-pro-teardown

四. 计算机语言

计算机语言是人与计算机之间交流信息的工具。

```
0000 0001
0000 0000
1000 1010   ; 2号单元内容取入AH寄存器
0010 0110
0000 0010
0000 0000
0000 0000   ; AL的内容与AH的内容相加，结
1110 0000   ; 果存入AL中
1010 0010   ; AL的内容送入3号单元
0000 0011
0000 0000
```

机器语言（machine code）

Assembly for ARM

```
    mov x29, sp
    .cfi_def_cfa w29, 16
    .cfi_offset w30, -8
    .cfi_offset w29, -16
    adrp    x0, l_.str@PAGE
    add x0, x0, l_.str@PAGEOFF
    bl  _printf
    mov w0, #0
    ldp x29, x30, [sp], #16             ; 16-byte Folded Reload
    ret
    .cfi_endproc
                                        ; -- End function
    .section    __TEXT,__cstring,cstring_literals
l_.str:                                 ; @.str
    .asciz  "Hello World!\r\n"
```

汇编语言（Assembly）

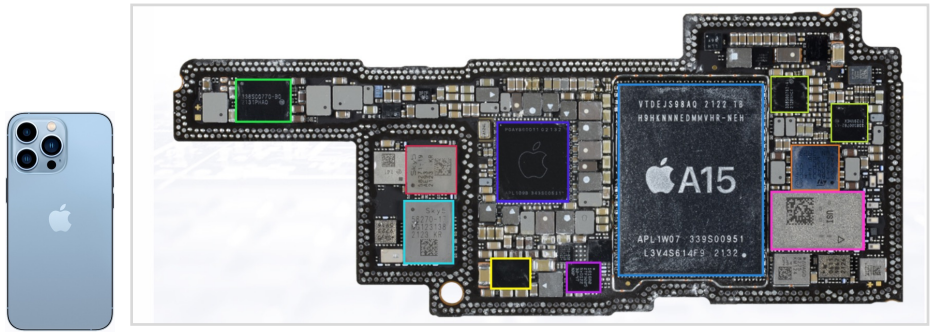
C

```c
#include <stdio.h>

int main(){
    printf("Hello World!\r\n");
}
```

Python

```python
x = 1
if x == 1:
    print("hello world!")

```

高级语言（通用语言）



计算机硬件系统结构图[1]

```
TERMINAL   PROBLEMS   OUTPUT   DEB

❯ gcc hello_world.c -o hello_world
❯ ./hello_world
Hello World!
```


[1] https://www.techinsights.com/blog/teardown/apple-iphone-13-pro-teardown

四. 计算机语言

1. 机器语言（Machine Language）

机器指令：可以使计算机完成各种操作的“1”和“0”的不同组合的数码串

机器语言：就是与机器硬件有紧密联系的机器指令的集合，是最初级且依赖于硬件的语言。是由0和1组成的二进制代码表示的语言，是计算机唯一能直接识别并执行的语言。

示例：

```
1010 0000    ；1号单元内容取入AL寄存器
0000 0001
0000 0000
1000 1010    ；2号单元内容取入AH寄存器
0010 0110
0000 0010
0000 0000
1111 0100    ；停机
```

四. 计算机语言

1. 机器语言 （Machine Language ）

优点：

1) 计算机能直接认识、执行;
2) 算法刻画细致;
3) 程序紧凑,占用内存空间少,执行速度高;
4) 能充分发挥计算机的硬件功能。

缺点：

1) 机器语言难记,程序难编,易错,调试困难;
2) 机器语言通用性差。

2. 汇编语言（Assembly Language）

汇编语言：是任何一种用于电子计算机、微处理器、微控制器等硬件的低级语言。是用字母和符号等助记符表示指令和操作数地址的计算机语言,又称为符号语言。

种类：8086汇编，8051汇编，MSP430汇编，ARM汇编等等

2. 汇编语言（Assembly Language）

汇编语言：是任何一种用于电子计算机、微处理器、微控制器等硬件的低级语言。是用字母和符号等助记符表示指令和操作数地址的计算机语言,又称为符号语言。

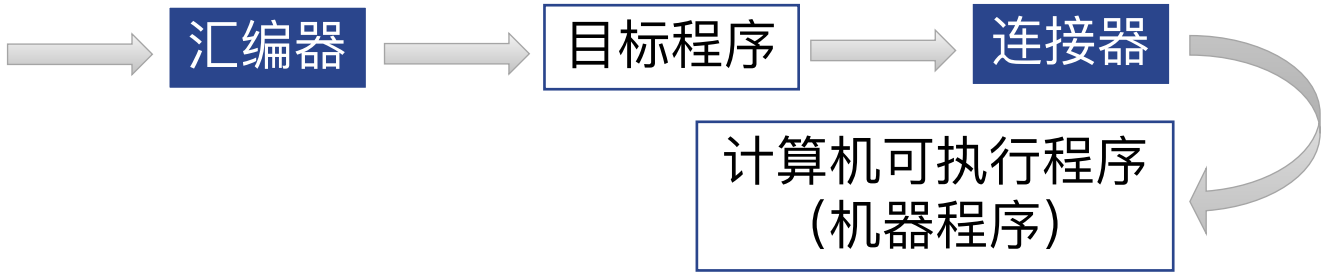
种类：8086汇编，8051汇编，MSP430汇编，ARM汇编等

其他概念：

- 汇编过程：把汇编语言程序翻译成机器语言程序的过程称为汇编过程
- 汇编程序（汇编器）：完成汇编过程的程序称为汇编程序
- 汇编语言程序：程序设计人员用汇编语言编写的程序， 称为源程序
- 目标程序：汇编器产生的结果

```
MOV     AL, DATA1    ；取第一个数据
MOV     AH, DATA2    ；取第二个数据
ADD     AL, AH       ；求和
MOV     RLT, AL      ；保存结果
HLT                  ；停机
```

源程序 main.S



2. 汇编语言（Assembly Language）

优点：

1) 程序容易编制、出错机会少、容易调试;
2) 算法刻画细致;
3) 程序紧凑、占用内存空间少、执行速度高;
4) 能充分发挥计算机的硬件功能；

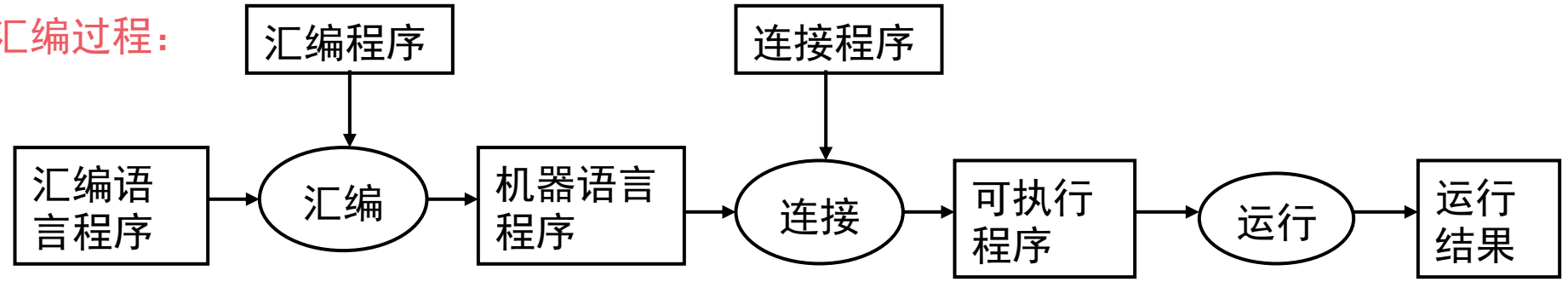
缺点：

1) 计算机不能直接认识、执行，必须经过汇编变为机器语言程序;
2) 通用性差;

3. 高级语言

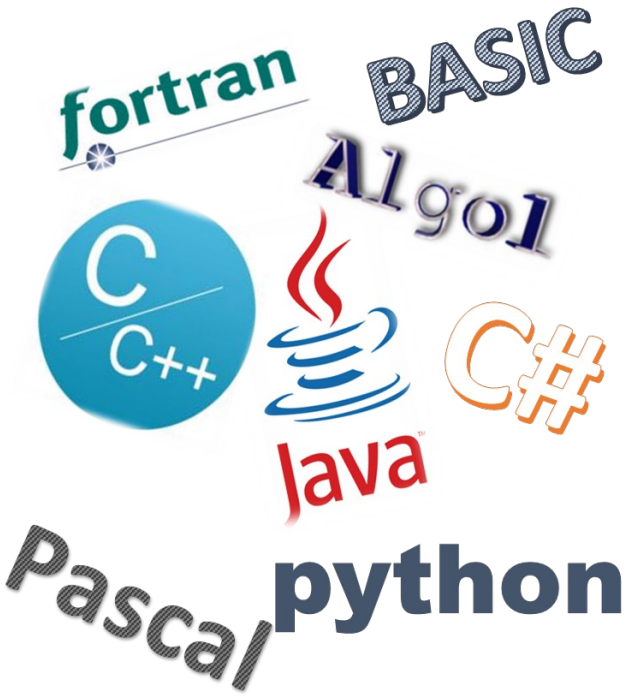
- 低级语言-面向机器 （发展时期 1946—1953 ）
  - 机器语言：计算机能直接识别的语言，这种语言编写的源程序都是由0和1的二进制编码组成，能唯一被计算机识别的语言。其缺点显而易见：
    - 可读性差，编程难度大
    - 可移植性差（通用性差）
    - 内存需要由人工分配
  - 汇编语言：用英文单词作为助记符，来代表机器语言中的各种指令。
    - 如：ADD AX, 5　含义：AX=AX+5

汇编过程：



3. 高级语言（发展时期1954—至今）

- 定义：人工创造的各种计算机都通用的、接近于人类“自然语言”的程序设计语言，又称“算法语言”。
- 特征：与计算机内部队指令系统无关，完全独立于计算机机型，而表达方式接近人类语言，和数学公式，容易被人所掌握和书写。



3. 高级语言（发展时期1954—至今）

- 以语言设计方法学分类
  - 面向过程：说明做什么，怎么做。
    - 例如： C 、PASCAL等
  - 面向对象：能够描述同一类对象的共同属性和行为。
    - 例如：C++、JAVA 、VB等

3. 高级语言（发展时期1954—至今）

- 以应用场景区分

编写操作系统语言

C语言(首选)：比汇编语言更精巧、更简单的版本，适于编写系统级的程序，比如操作系统。第一个使得系统级代码移植成为可能的编程语言。

C++语言：C++语言灵活，运算符的数据结构丰富、具有结构化控制语句、程序执行效率高，而且同时具有高级语言与汇编语言的优点。

编写应用软件语言

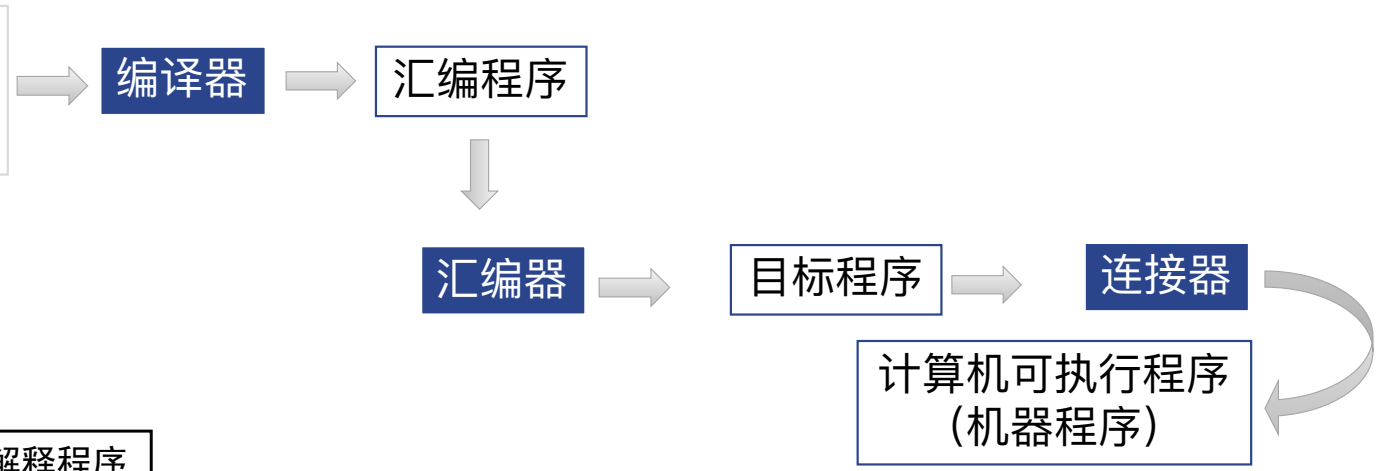
C、C++、PHP、Delphi、Java、Pascal、JSP、Objective-C、Swift

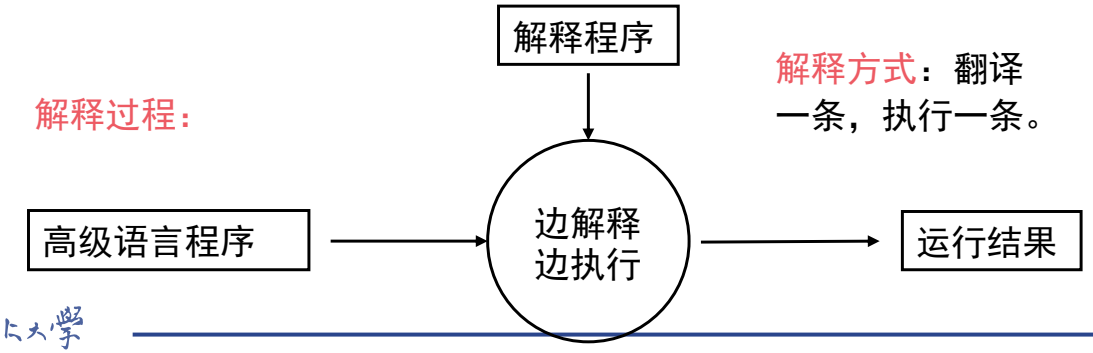
3. 高级语言（发展时期1954—至今）

- 翻译过程示例
  - 将用高级语言编写的程序（源程序）翻译成机器语言程序（目标程序）。这个翻译方式分为“编译方式”和“解释方式”。

源程序 main.c

```c
#include <stdio.h>

int main(){
    printf("Hello World!\r\n");
}
```





解释方式：翻译一条，执行一条。

四. 计算机语言

3. 高级语言（发展时期1954—至今）

.Net Framework语言	C++/CLI · C# · F# · IronPython · J# · Visual C# · Visual Basic .NET	
C/C++语言	C · C++ · Turbo C++ · Borland C++ · C++ Builder- C++/CLI · Objective-C · Visual C++	
BASIC语言	有行号	BASIC · BASICA · GW-BASIC · ETBASIC · GVBASIC
	无行号	QBASIC · QuickBASIC · True BASIC · Turbo BASIC· PowerBASIC · DarkBASIC Visual Basic · Visual Basic .NET · VBScript · Visual Basic for Applications（VBA） · REALbasic
Pascal/Delphi语言	Pascal语法	Pascal · Turbo Pascal · Object Pascal · Free Pascal
	Pascal+Delphi语法	Delphi · Lazarus
ECMAScript语言	ActionScript · DMDScript · JavaScript · JScript	
Python语言	Python · IronPython · Jython	
数据库相关语言	Clipper · Visual FoxPro · SQL（PL/SQL · T-SQL） · SQL存储过程	
GPU着色器语言	Cg · GLSL · HLSL	
学术语言	APL/J · Clean · Haskell · Jess · Logo · ML · Prolog · Scheme · SAC	
其他语言	A+ · Ada B · Brainfuck · COBOL · Curl · D · Eiffel · Erlang · FORTRAN · Java · LISP · Lua · SCILAB · MATLAB · MATHEMATICA · Nuva · Oberon · OCaml · Perl · PHP · PostScript · Powerbuilder · Python · R · REXX · Ruby · Self · Smalltalk · Tcl/Tk · ALGOL · Forth · Modula-2/Modula-3 · MUMPS · PL/I · Simula· 汇编语言	

3. 高级语言

优点：

1) 程序容易编制、出错机会少、容易调试;
2) 容易表达算法;
3) 通用性强;

缺点：

1) 计算机不能直接认识、执行，必须经过编译变为机器语言程序;
2) 编译后生成的机器语言程序冗长、占用内存空间多、执行速度低;
3) 不能充分发挥计算机的硬件功能；

为什么学习汇编语言？

1. 嵌入式程序：空调控制程序、磁盘驱动器程序等

   需求：较小的代码体积，汇编程序代码体量小

2. 实时控制程序：汽车燃油点火控制程序等

   需求：执行时间的确定性，汇编程序可精确到CPU cycle

3. 电脑游戏软件

   需求：执行效率优化，汇编语言可以直接访问硬件

4. 设备驱动程序：

   需求：设备寄存器等基本硬件逻辑的控制

5. 有助于形成对计算机硬件、操作系统和应用程序之间交互的全面理解

本章小结

一个例子：

- 计算机语言
  - 高级语言
  - 汇编语言
  - 机器语言
- 计算机硬件
  - CPU
  - 存储

C Code

```
int sum(int x, int y) {
  int t = x+y;
  return t;
}
```

code.c

本章小结

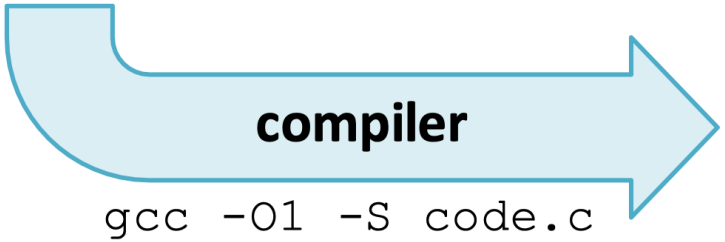
一个例子：

- 计算机语言
  - 高级语言
  - 汇编语言
  - 机器语言
- 计算机硬件
  - CPU
  - 存储

C Code

```
int sum(int x, int y) {
  int t = x+y;
  return t;
}
```

code.c



Generated IA32 Assembly Code

Human-readable language close to machine code.

```
sum:
    pushl %ebp
    movl %esp,%ebp
    movl 12(%ebp),%eax
    addl 8(%ebp),%eax
    movl %ebp,%esp
    popl %ebp
    ret
```

code.s

本章小结

一个例子：

- 计算机语言
  - 高级语言
  - 汇编语言
  - 机器语言
- 计算机硬件
  - CPU
  - 存储

C Code

```
int sum(int x, int y) {
  int t = x+y;
  return t;
}
```

code.c

compiler

gcc -O1 -S code.c

Generated IA32 Assembly Code

Human-readable language close to machine code.

```
sum:
    pushl %ebp
    movl %esp,%ebp
    movl 12(%ebp),%eax
    addl 8(%ebp),%eax
    movl %ebp,%esp
    popl %ebp
    ret
```

code.s

assembler

Object Code

```
01010101100010011110010110
00101101000101000011000000
00110100010100001000100010
01111011000101110111000011
```

code.o

Linker: create full executable

Resolve references between object files, libraries, (re)locate data

本章小结

一个例子：

- 计算机语言
  - 高级语言
  - 汇编语言
  - 机器语言
- 计算机硬件
  - CPU
  - 存储
- 提问 →

C Code

```
int sum(int x, int y) {
  int t = x+y;
  return t;
}
```

code.c

Generated IA32 Assembly Code

Human-readable language close to machine code.

```
sum:
    pushl %ebp
    movl %esp,%ebp
    movl 12(%ebp),%eax
    addl 8(%ebp),%eax
    movl %ebp,%esp
    popl %ebp
    ret
```

code.s

Object Code

```
01010101100010011110010110
00101101000101000011000000
00110100010100001000100010
01111011000101110111000011
```

code.o

映射关系？

(1) Processor

(2) Memory

(3) Disk Drive

四. 本章小结

思考题：

1） 试分析机器语言、汇编语言、通用语言之间的优缺点。