



東北大學
Northeastern University

汇编语言程序设计

主讲： 刘松冉（讲师）

单位： 东北大学计算机学院

智慧系统国际联合实验室

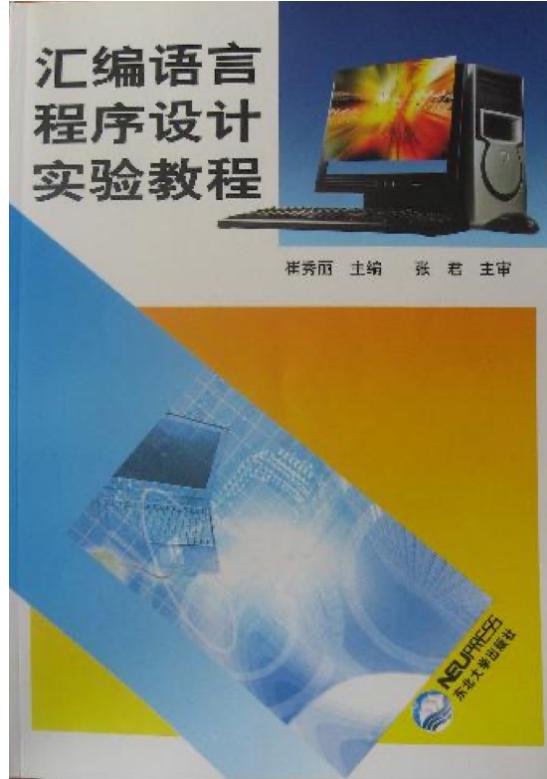
联系方式： liusongran@cse.neu.edu.cn

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

参考书目



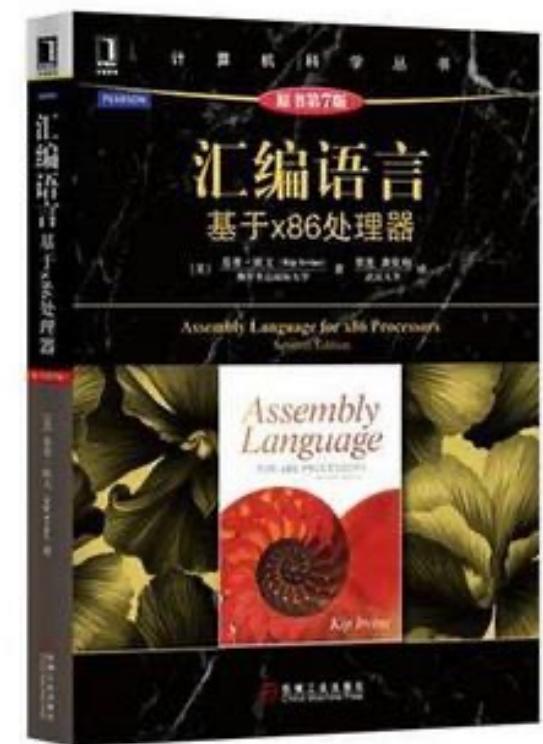
《汇编语言程序设计》第五版
高福祥、张君主编，东北大学出版社出版



《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·欧文著 机械工业出版社

第一章 概述

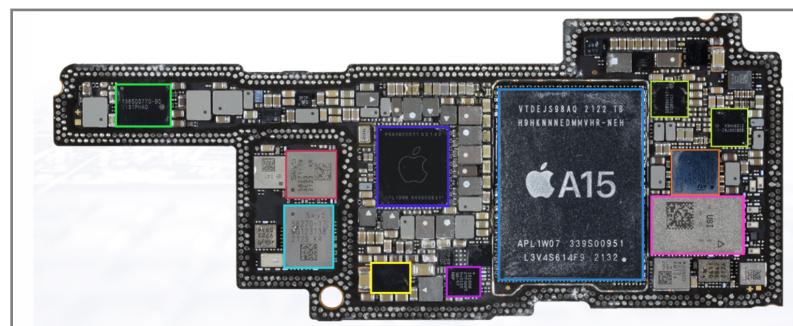
- 一. 计算机语言
- 二. 为什么要学习汇编语言
- 三. 计算机系统



一. 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）？



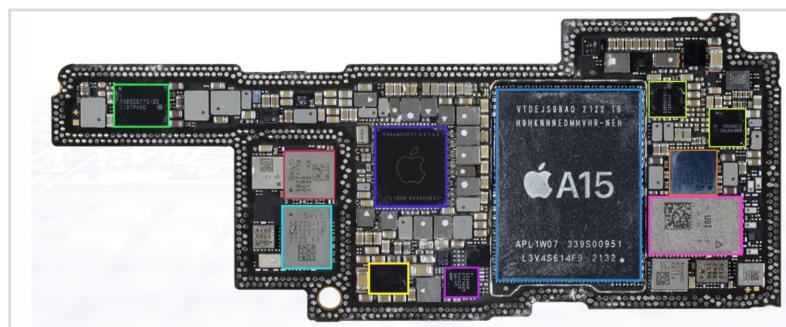
TERMINAL PROBLEMS OUTPUT DEB

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

计算机硬件系统结构图^[1]

一. 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。



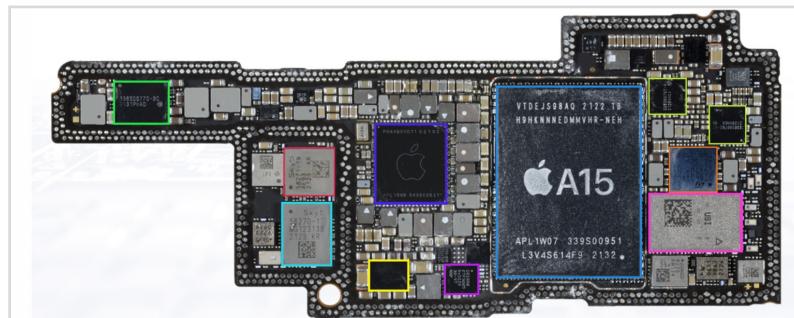
计算机硬件系统结构图^[1]

```
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, 结果存入AL中  
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)

一. 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]

```
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
21     .section __TEXT,__cstring,cstring_literals ; -- End function
22 l_.str:
23     .asciz "Hello World!\r\n"
```

Assembly for ARM

汇编语言 (Assembly)

一. 计算机语言

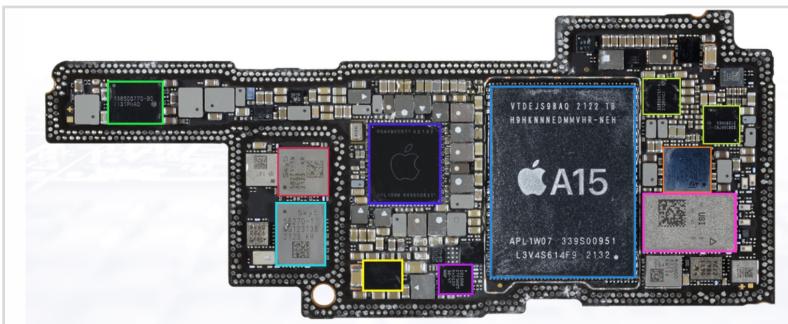
► 计算机语言是人与计算机之间交流信息的工具。

```
1 #include <stdio.h>
2
3 int main(){
4     printf("Hello World!\r\n");
5 }
```

c

```
1 x = 1
2 if x == 1:
3     print("hello world!")
4
```

Python



计算机硬件系统结构图^[1]

高级语言（通用语言）

一. 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。

```
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)

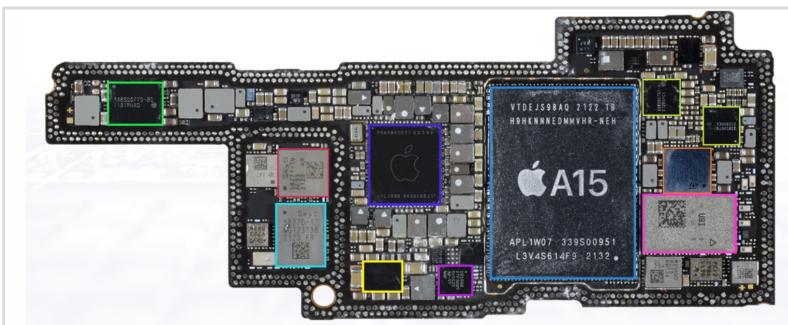
```
9      mov x29, sp  
10     .cfi_def_cfa w29, 16  
11     .cfi_offset w30, -8  
12     .cfi_offset w29, -16  
13     adrp x0, l_.str@PAGE  
14     add x0, x0, l_.str@PAGEOFF  
15     bl _printf  
16     mov w0, #0  
17     ldp x29, x30, [sp], #16           ; 16-byte Folded Reload  
18     ret  
19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

汇编语言 (Assembly)

```
1 #include <stdio.h>          C  
2  
3 int main(){  
4     printf("Hello World!\r\n");  
5 }
```

```
1 x = 1                      Python  
2 if x == 1:  
3     print("hello world!")  
4
```

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!
```

一. 计算机语言

► 1. 机器语言

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

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  
0000 0000 ; AL的内容与AH的内容相加, 结果存入AL中  
1110 0000 ; 果存入AL中  
1010 0010 ; AL的内容送入3号单元  
0000 0011  
0000 0000  
1111 0100 ; 停机
```

一. 计算机语言

► 1. 机器语言

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。

一. 计算机语言

▶ 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。一种汇编语言专用于某种计算机系统结构。

种类：8086汇编，8051汇编，MSP430汇编，ARM汇编等

一. 计算机语言

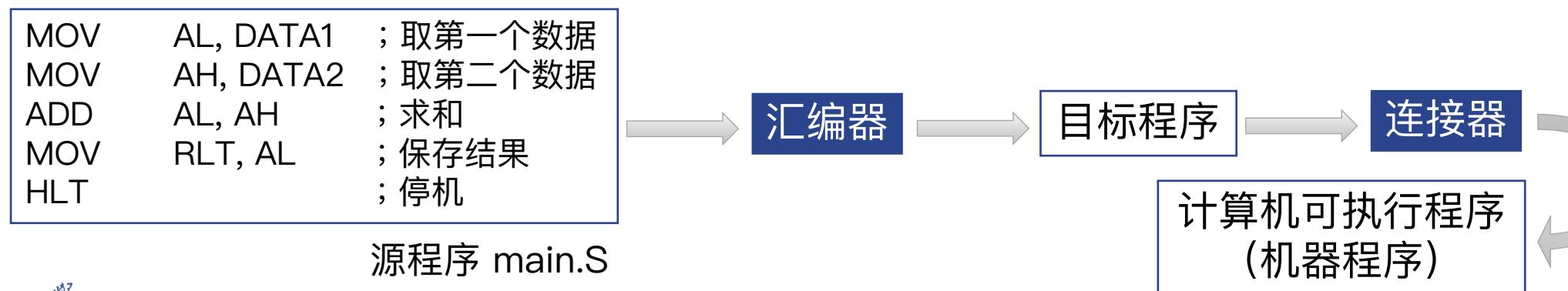
2. 汇编语言 (Assembly Language)

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。一种汇编语言专用于某种计算机系统结构。

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



一. 计算机语言

▶ 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;

一. 计算机语言

3. 高级语言（通用语言）

高级语言：是用来对计算机操作步骤进行描述的一整套标记符号、表达格式、结构及其使用的语法规则。

种类：C，C++，Rust，Java，Python，BASIC，Matlab等

过程示例：

源程序 main.c

```
1 #include <stdio.h>
2
3 int main(){
4     printf("Hello World!\r\n");
5 }
```

C

编译器

汇编程序



汇编器

目标程序

连接器

计算机可执行程序
(机器程序)



一. 计算机语言

▶ 3. 高级语言（通用语言）

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；

第一章 概述

- 一. 计算机语言
- 二. 为什么要学习汇编语言
- 三. 计算机系统



二. 为什么要学习汇编语言

▶ 汇编语言的用途

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

二. 为什么要学习汇编语言

► 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 80486 (DX/SX) |
| • Intel 8008 | Intel Pentium (MMX) |
| • Intel 8080 | Intel PII (Celeron) |
| • Intel 8086/8088 | Intel III(Celeron II) |
| • Intel 80186/80188 | Intel P4 |
| • Intel 80286 | ... |
| • Intel 80386(DX/SX) | |



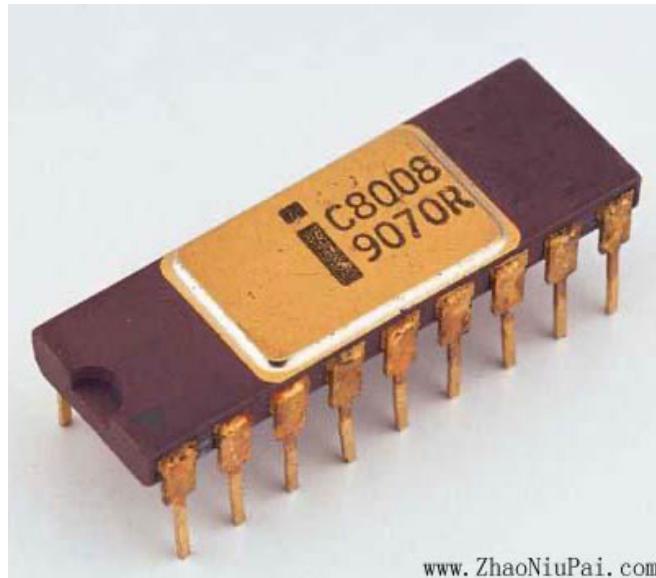
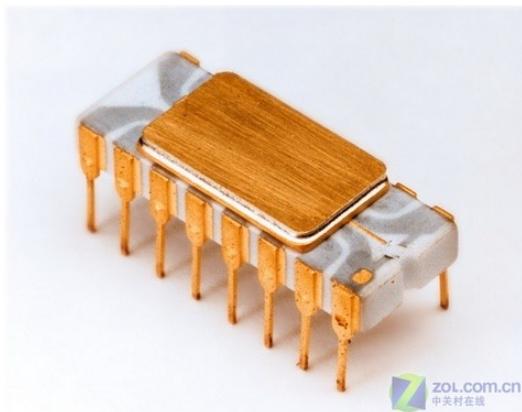
自8086起，指令集向后兼容

二. 为什么要学习汇编语言

▶ 为什么学习8086/8088的汇编语言？

Intel 4004

1971年推出，16只引脚，内有2250个晶体管，频率740k Hz，能执行4位运算，支持8位指令集及12位地址集。为日本公司Busicom 设计用来生产计算器的。

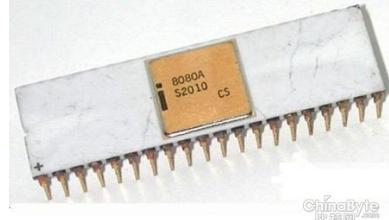


Intel 8008

1972年推出。可以支持到16KB的内存。8位运算+16位地址总线。
两种速度：0.5 MHz和0.8 MHz。

Intel 8080

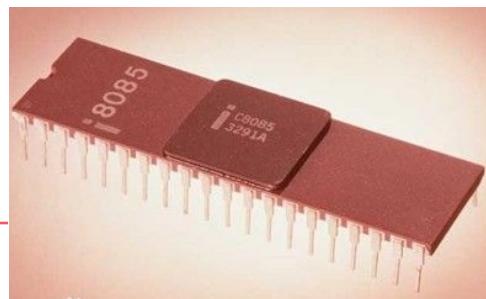
1974年推出2MHz 和3MHz的频率8位数据总线16位地址总线6000个晶体管64KB 可寻址内存40引脚0.64 MIPS 运算能力同8008相同的汇编语言



ChinaByte
比特网

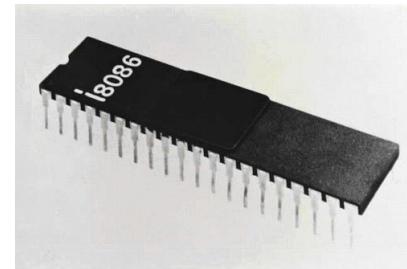
Intel 8085

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。



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



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

Intel 8088

1979年推出
基本与8086相同
但是外部数据总线变为了8位，内部仍为16位。



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：实模式和保护模式



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，另外包括了中断
控制器、定时器、DMA、I/O、
UART、片选电路等外设。



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、保护模式和虚拟86模式

Intel 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



Intel 80486SX

1991年推出

不带387的80486



intel®

i486™ DX2™

A80486DX2-50
C2271308
SX626
INTEL ©1989

Intel 80486

1989年推出

最高25~50MHz的频率

32 位数据总线

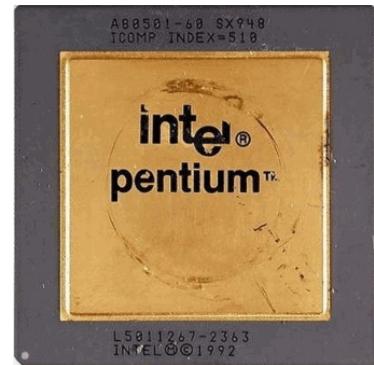
32 位地址总线

120万个晶体管

4GB 可寻址内存

386CPU+387CPU+Cache

采用了倍频技术



Intel Pentium(奔腾)

1993年推出

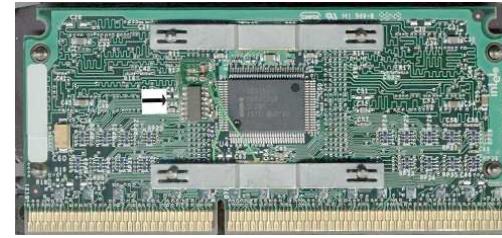
60、66、100MHz的频率

310万个晶体管

二级Cache 256KB和512KB

Intel 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



Intel Pentium III

1999年发布
733MHz~1GHz

Intel Pentium 4

Pentium 4诞生于2000年，生产工艺升级为0.18微米，晶体管数量也高达4200万，历经Willamette、Northwood、Prescott三代。



Intel Pentium Dual Core

2006年

至今

Intel® Core™ Processors	Intel® Pentium® Processor	Intel® Celeron® Processor
Intel® Xeon® Processors	Intel® Xeon Phi™ Processors	Intel® Itanium® Processor
Intel Atom® Processor	Intel® Quark™ SoC	

异构多核（Adler Lake），AI核，量子核
未完待续

第一章 概述

- 一. 计算机语言
- 二. 为什么要学习汇编语言
- 三. 计算机系统



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。

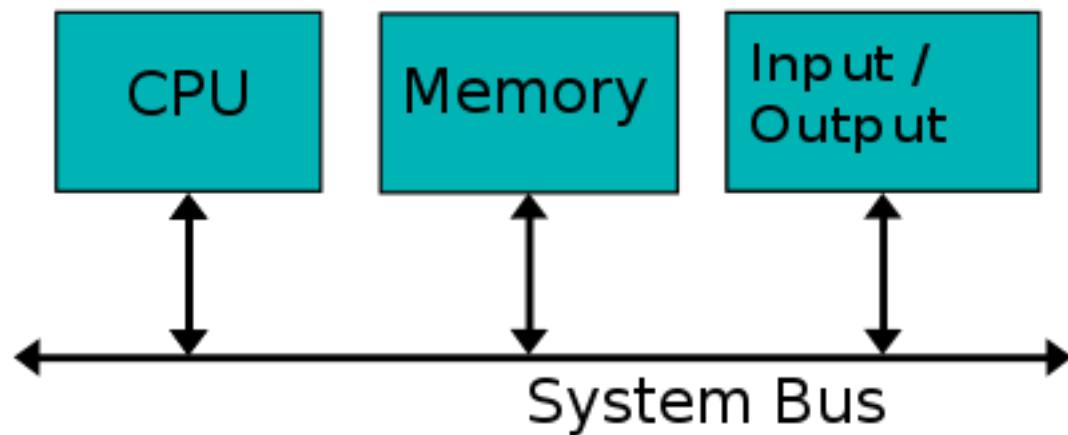
- 系统：相关事物组成的集合称为系统，组成系统的事物称为该系统的元件或部件

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

- 中央处理器CPU
- 存储装置
- 外围设备
- 各级总线

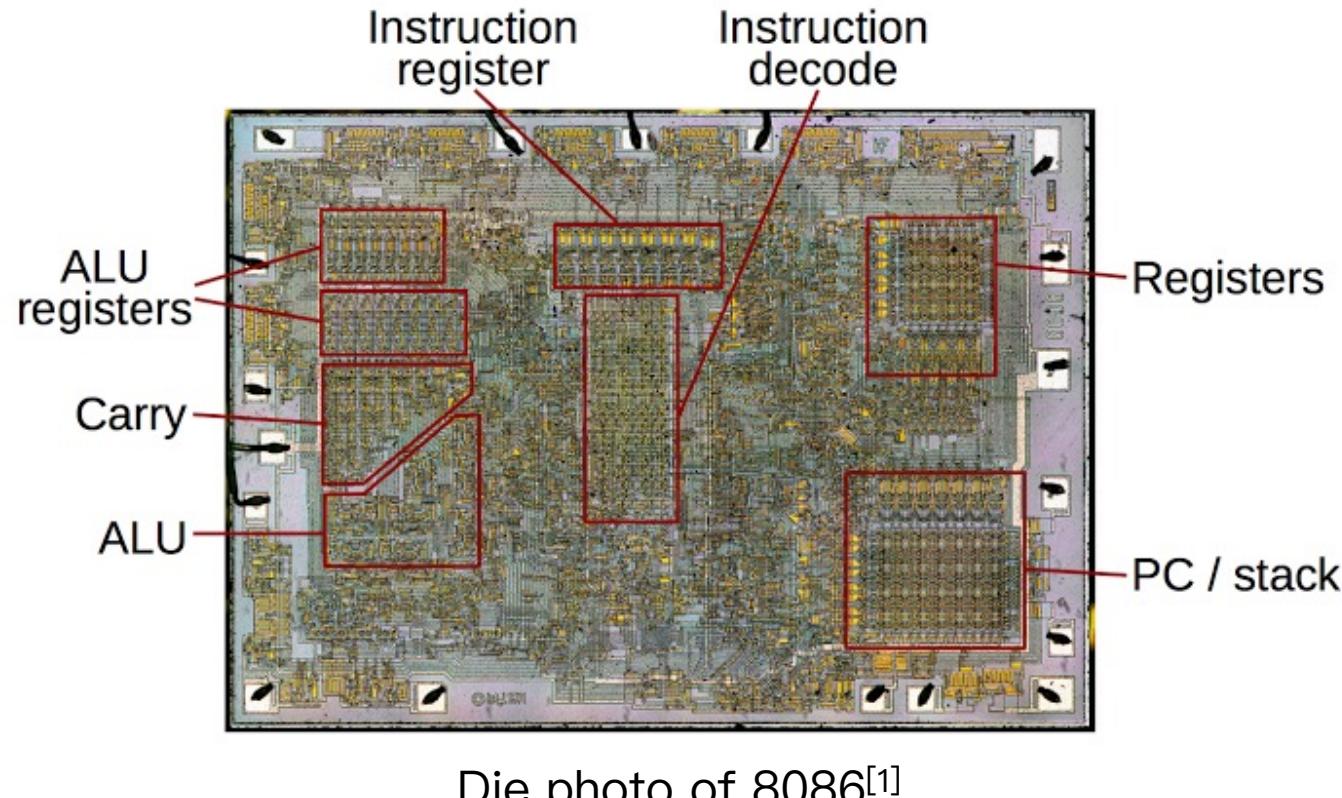


三. 计算机系统

▶ 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



三. 计算机系统

▶ 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

三. 计算机系统

► 硬件子系统

- 中央处理器CPU
- 存储装置
- 外围设备
 - 输入设备：用来向计算机的主存储器或CPU送入程序或数据的装置。如：键盘
 - 输出设备：用来将计算机的处理结果以 某种形式输出给用户的装置。如：显示器
- 各级总线

三. 计算机系统

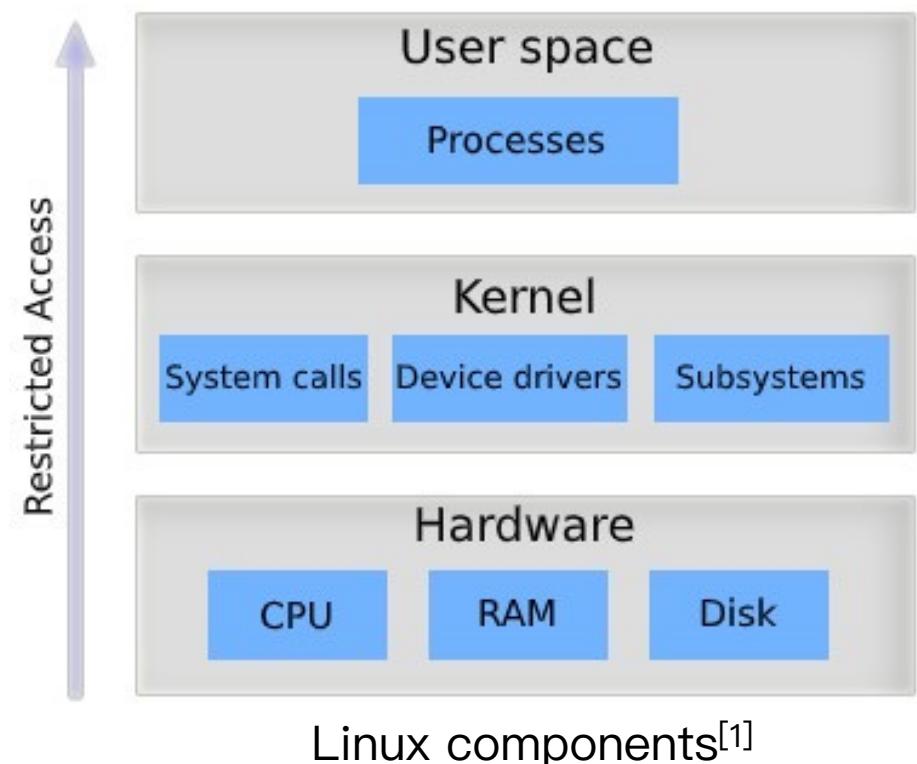
► 计算机系统由硬件子系统和软件子系统组成。

- 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。

➤ 用户层：应用程序

➤ 系统层：

- 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



四. 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

四. 本章小结

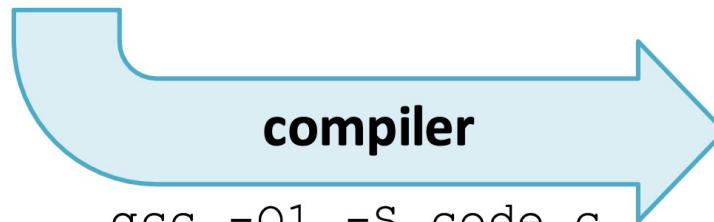
一个例子：

- 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



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

四. 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

四. 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