

程序的机器级表示（1）

谢旻晖

2025.10

Outline

- CPU发展
- 自下而上：构建计算机体系
- CPU Architecture
- Memory and Registers
- 处理器架构实例：华为鲲鹏处理器

ENIAC

- 第一台电子计算机，宾夕法尼亚大学，1946年
 - 建造的目的在于取代人工，计算导弹弹道
 - **ENIAC**用了**18000**电子管、**1500**继电器、重**30**吨、占地**170m²**、耗电**140kw**、每秒计算**5000**次加法
 - 相当于手工计算的**20**万倍、机电式计算机的**1000**倍
 - 纸带输入程序和数据，人工时间占比很高



图 2.8 世界上第一台数字式电子计算机 ENIAC

企业级计算机：IBM和DEC

IBM S/360, 1960s

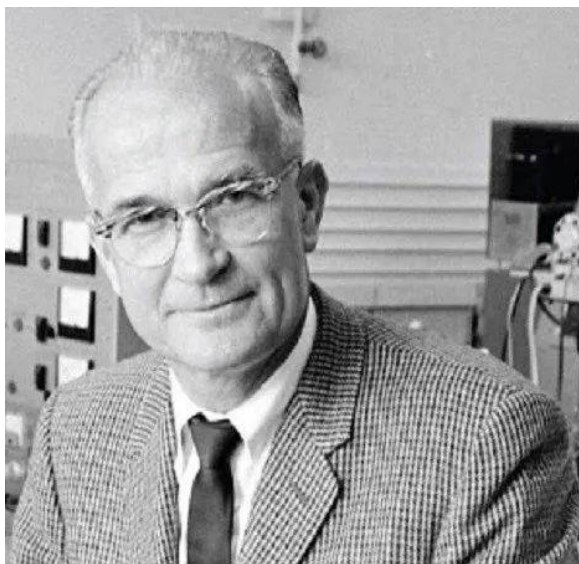


DEC PDP-11



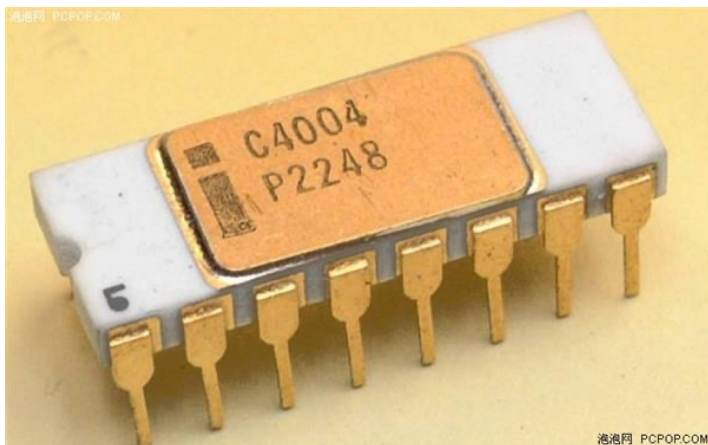
仙童

- 1957年，“晶体管之父”肖克利建立公司开始晶体管研发
- 旗下“八叛徒”建立仙童公司，后建立**Intel**，**AMD**等公司



Intel x86 CPU

- Intel成立于1968年，创始人包括来自仙童的葛罗夫（A. Grove）和戈登·摩尔（Gordon Moore）；AMD成立于1969年，创始人是仙童的销售杰里·桑德斯（Jerry Sanders）
- 1971年第一款4位微处理器Intel 4004问世，1974年第一款8位微处理器Intel 8080问世



Intel 4004是第一个商用微处理器

采用**4004**的**Busicom Unicom⁶**计算器

X86 family

8088CPU

- 8086(1978, 29K)
 - The heart of the **IBM PC** & DOS (**8088**)
 - 16-bit, 1M bytes addressable, 640K for users
 - x87 for floating pointing



- 80286(1982, 134K)
 - 80186编号给了不太重要的一个芯片
 - 为了与摩托罗拉68000系列CPU竞争
 - Basis of the IBM PC-AT & Windows



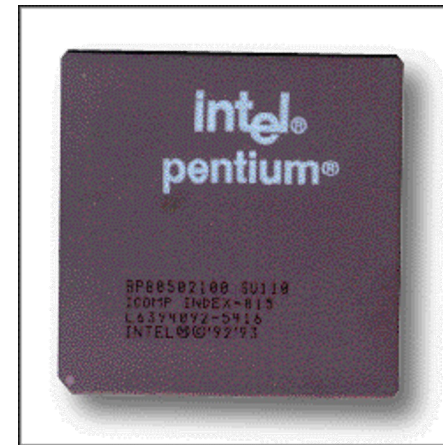
IBM PC机

- i386(1985, 275K)
 - 32 bits architecture, flat addressing model
 - Support a Unix operating system



X86 family

- I486(1989, 1.9M)
 - Integrated the **floating-point** unit onto the processor chip
- Pentium(1993, 3.1M)
 - Improved performance, added minor extensions
- PentiumPro(1995, 5.5M)
 - P6 microarchitecture
 - Conditional mov
- Pentium II(1997, 7M)
 - Continuation of the P6



X86 family



- Pentium III(1999, 8.2M)
 - New class of instructions for manipulating vectors of floating-point numbers(SSE, Stream SIMD Extension)
 - Later to 24M due to the incorporation of the level-2 cache
- Pentium 4(2001, 42M)
 - Netburst microarchitecture with high clock rate but high power consumption
 - SSE2 instructions, new data types (eg. Double precision)

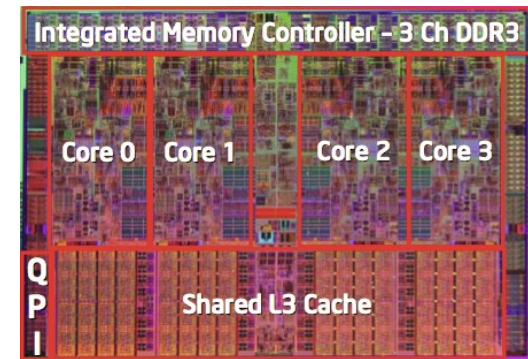


X86 family

- Pentium 4E: (2004, 125M transistors).
 - Added *hyperthreading*
 - run two programs simultaneously on a single processor
 - EM64T, 64-bit extension to IA32
 - First developed by Advanced Micro Devices (AMD)
 - x86-64
- Core 2: (2006, 291M transistors)
 - back to a microarchitecture similar to P6
 - multi-core (multiple processors a single chip)
 - Did not support hyperthreading

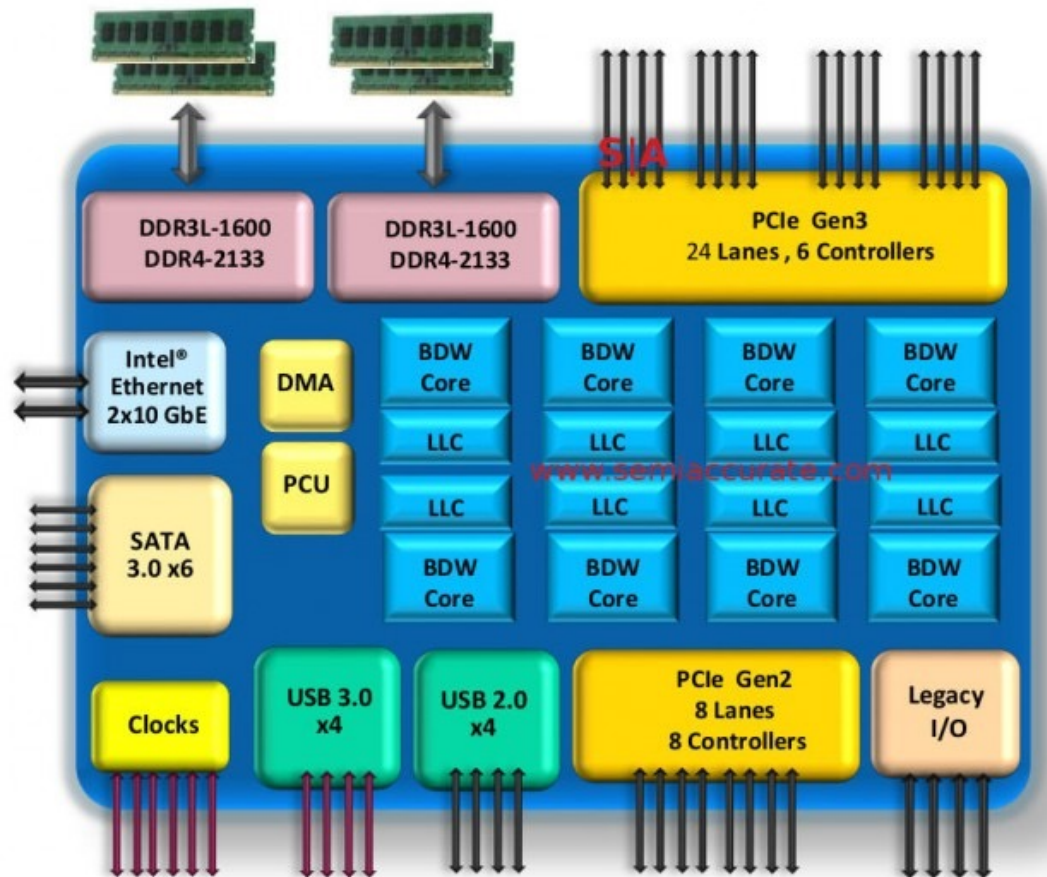
X86 family

- Core i7: (2008, 781 M transistors).
 - Incorporated **both hyperthreading and multi-core**
 - the initial version supporting two executing programs on each core
- Core i7: (2012, 2.27B transistors, Sandy Bridge-EP)
 - 8 cores on each chip
 - 2.7G (3.3G)
- Core i7: (2015, 5.5B transistors, Haswell-EP)
 - 18-core Xeon



X86 family

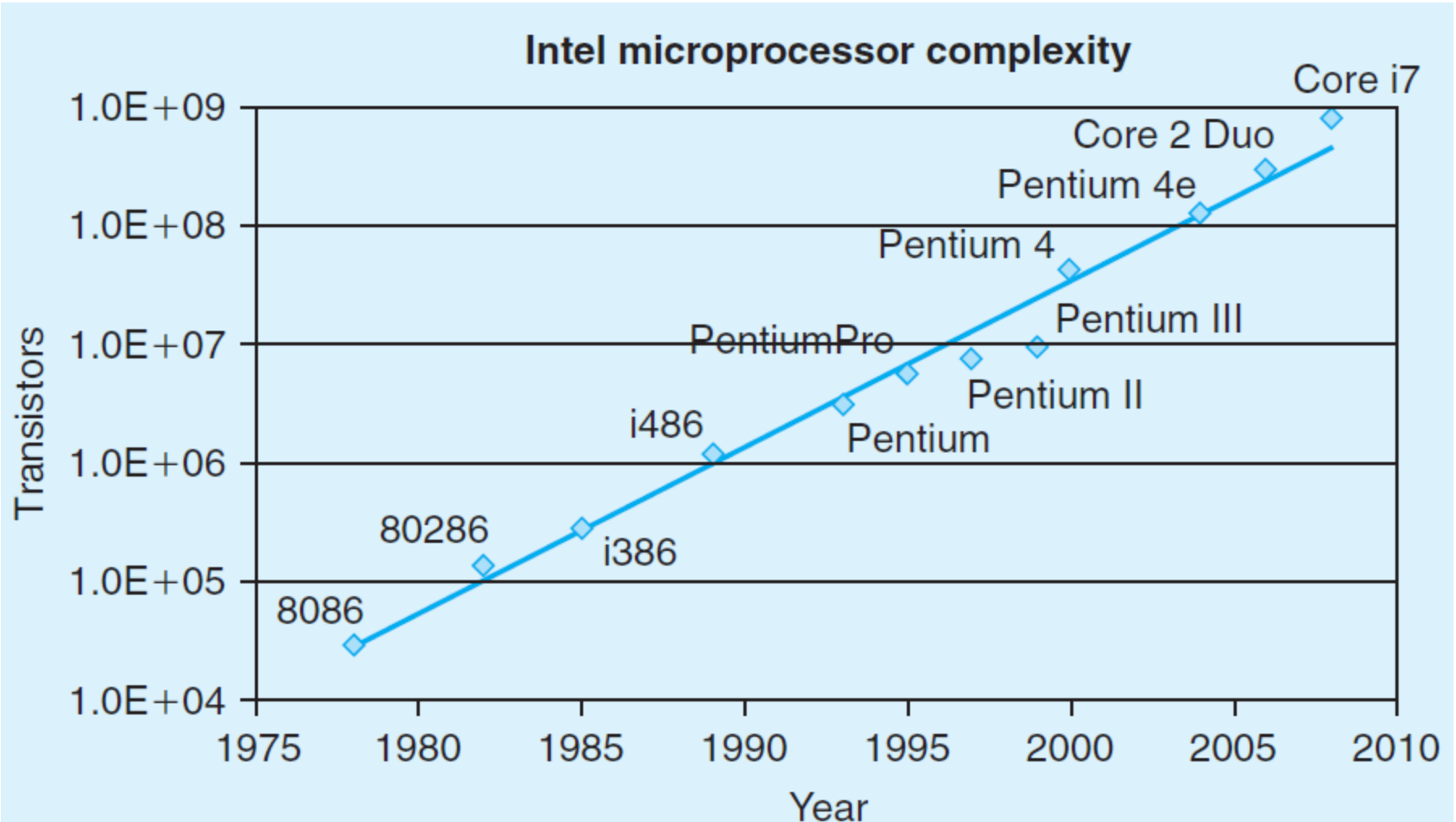
- Core i7 Broadwell 2015
- Desktop Model
 - 4 cores
 - Integrated graphics
 - 3.3-3.8 GHz
 - 65W
- Server Model
 - 8 cores
 - Integrated I/O
 - 2-2.6 GHz
 - 45W



X86 family

- Advanced Micro Devices (AMD)
 - At beginning,
 - lagged just behind Intel in technology,
 - produced less expensive and lower performance processors
- In 1999
 - First broke the 1-gigahertz clock-speed barrier
- In 2002
 - Introduced x86-64
 - The widely adopted 64-bit extension to IA32

Moor's Law



苹果M1 Pro和M1 Max

- M1 Pro
 - 内存带宽200GB/s
 - 337 亿晶体管
- M1 Max
 - 内存带宽400GB/s
 - 570 亿晶体管

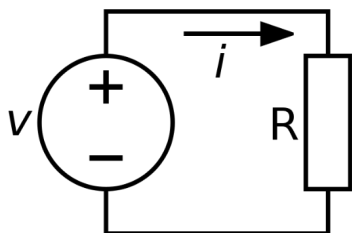


Our Coverage

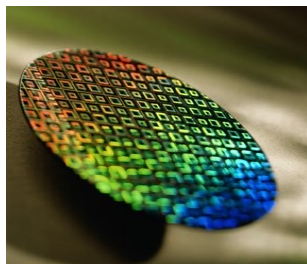
- IA32
 - The traditional x86
 - shark> gcc -m32 hello.c
- x86-64
 - The emerging standard
 - shark> gcc hello.c
 - shark> gcc -m64 hello.c

什么是芯片

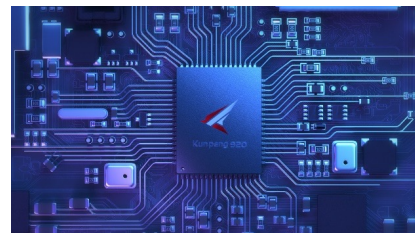
- 芯片 (Chip) 是一种把电路（主要包括半导体设备，也包括被动组件等）小型化的方式，并制造在半导体晶圆表面上。



电路



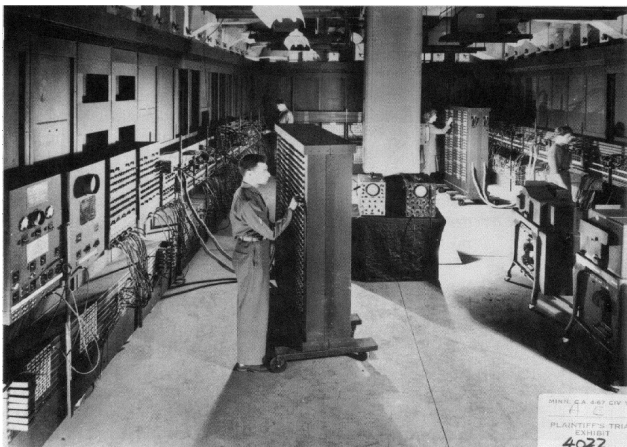
晶圆



芯片

芯片是信息社会的基石

世界第一台通用电子计算机—ENIAC



1946年诞生，由18,800多个电子管组成，重量30多吨，占地面积170多平方米。

得益于芯片的发展



小型化，高性能



内存



手机

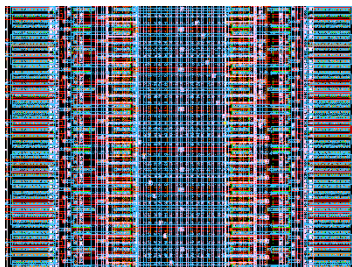


计算机



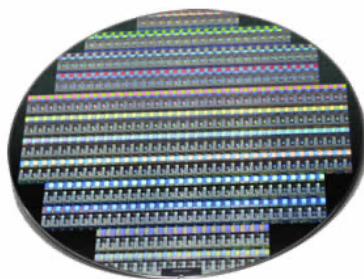
数码相机

芯片行业产业链



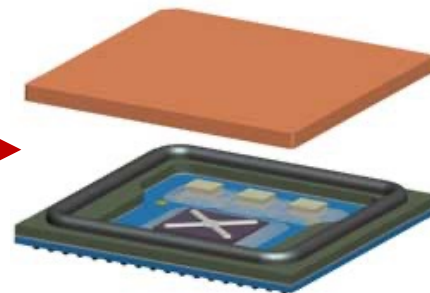
芯片设计

集成电路设计涉及对电子器件和器件间互连线模型的创建。



晶圆加工

晶圆加工是一系列化学处理步骤，使得电子电路逐渐形成在使用纯半导体材料制作的芯片上。



芯片封装测试

封装是将器件的核心晶粒封装在一个支撑物之内的过程，这个封装可以防止物理损坏以及化学腐蚀，并提供对外连接的引脚，之后将进行集成电路性能测试。

Outline

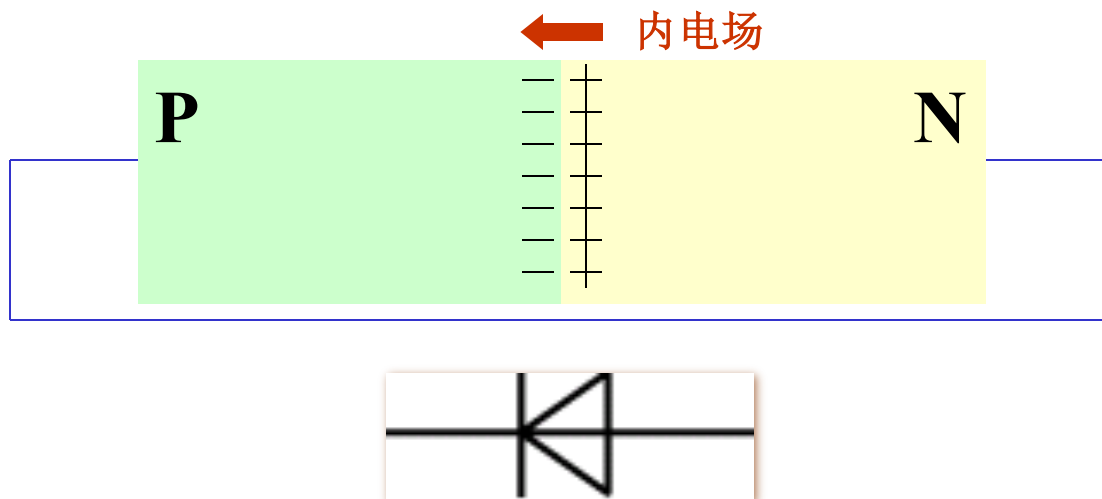
- CPU发展
- 自下而上：构建计算机体系
- CPU Architecture
- Memory and Registers
- 处理器架构实例：华为鲲鹏处理器

自下而上：构建计算机体系

- 1) 物理：PN结->门电路
- 2) 数字电路：门电路->组合逻辑电路/时序逻辑电路（CPU、寄存器、控制器...）
- 3) 抽象：冯诺伊曼体系结构
- 4) 硬件接口：机器语言/指令集
- 5) 软硬件分界面：汇编语言（汇编器）
- 6) 编译器：高级语言->汇编语言
- 7) 链接器：目标文件（机器代码序列）->可执行文件
- 8) 代码执行：程序->进程（加载器）
- 9) 进程空间：虚拟内存管理

自下而上：构建计算机体系

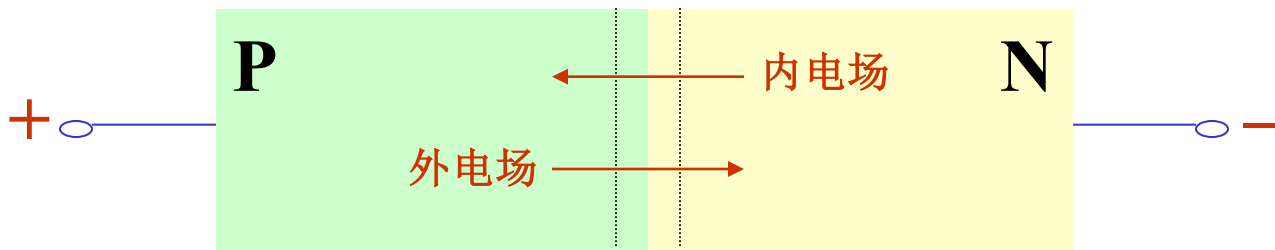
- 1) 物理：PN结->二极管->门电路
 - PN结：P型半导体和N型半导体构成
 - 单向导电性
 - 加正向电压时PN结导通
 - 加反向电压时PN结截止



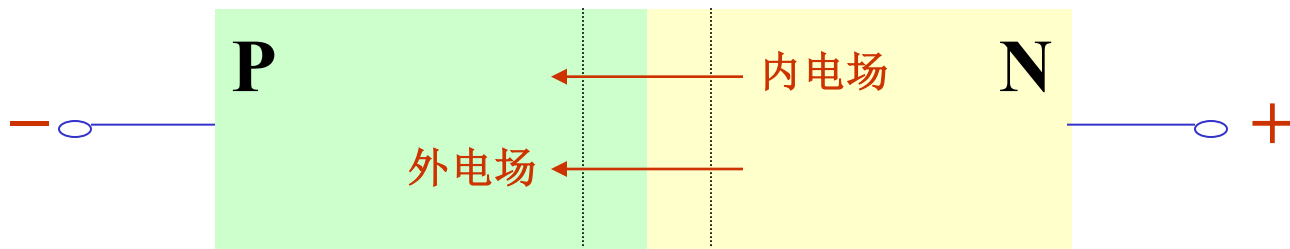
自下而上：构建计算机体系

在PN结两端适当加以外部电压，就可以打破上述平衡，实现单向导电的目地。

加正向电压时PN结导通：加正向电压时，内外电场方向相反，阻挡层变窄，相当于PN结的电阻很小。因此加正向电压时，PN结导通。

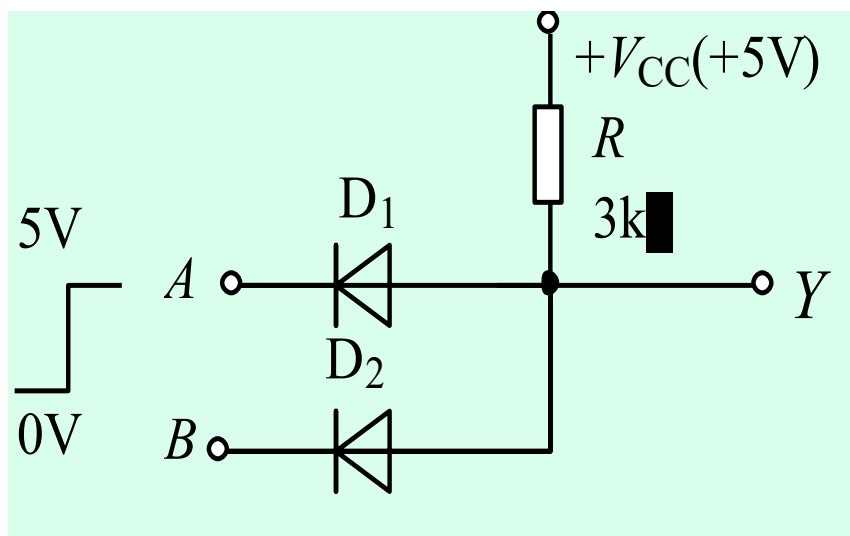


加反向电压时PN结截止：加反向电压时，内外电场方向相同，阻挡层变宽，相当于PN结的电阻很大。因此加反向电压时，PN结截止。



自下而上：构建计算机体系

- 1) 物理：PN结->二极管（一种晶体管）->门电路
 - 与门（逻辑运算）

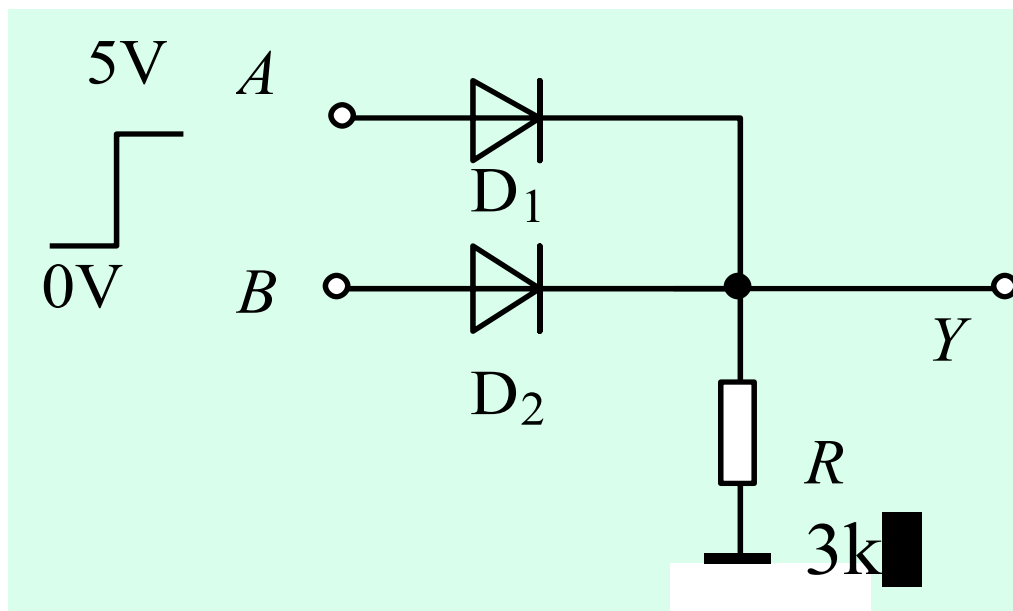


$$Y=AB$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

自下而上：构建计算机体系

- 1) 物理：PN结->二极管->门电路
 - 或门（逻辑运算）



$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

自下而上：构建计算机体系

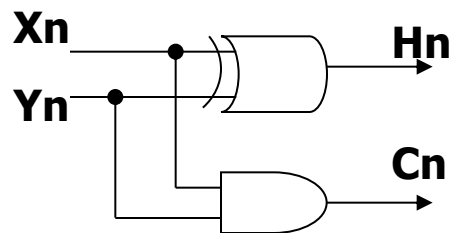
- 2) 数字电路：门电路->组合逻辑电路/时序逻辑电路（CPU、寄存器、控制器...）
 - 组合逻辑电路：
 - 加法器、ALU、译码器
 - 时序逻辑电路
 - 寄存器、计数器

X_n	Y_n	H_n	C_n
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

半加器：
不考虑进位输入时，两数码 X_n ， Y_n 相加称为半加

H_n ：本位二进制值
 C_n ：想更高位的进位

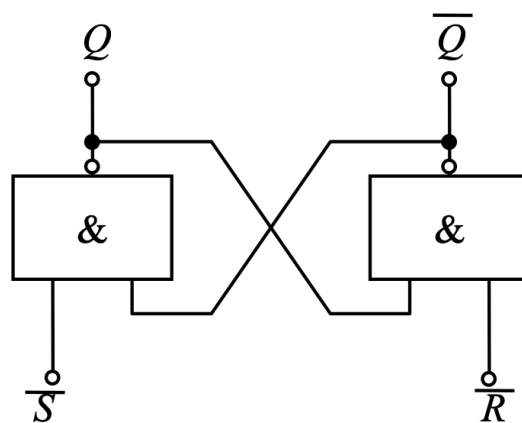
$$H_n = X_n \cdot Y_n + X_n \cdot Y_n = X_n \oplus Y_n$$
$$C_n = X_n \cdot Y_n$$



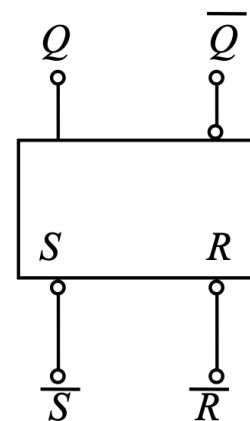
自下而上：构建计算机体系

- 2) 数字电路：门电路->组合逻辑电路/时序逻辑电路（CPU、寄存器、控制器...）
 - 时序逻辑电路举例：用与非门构造RS触发器
 - 具有记忆功能
 - 不断电的前提下

信号输出端， $Q=0$ 、 $\overline{Q}=1$ 的状态称**0**状态， $Q=1$ 、 $\overline{Q}=0$ 的状态称**1**状态

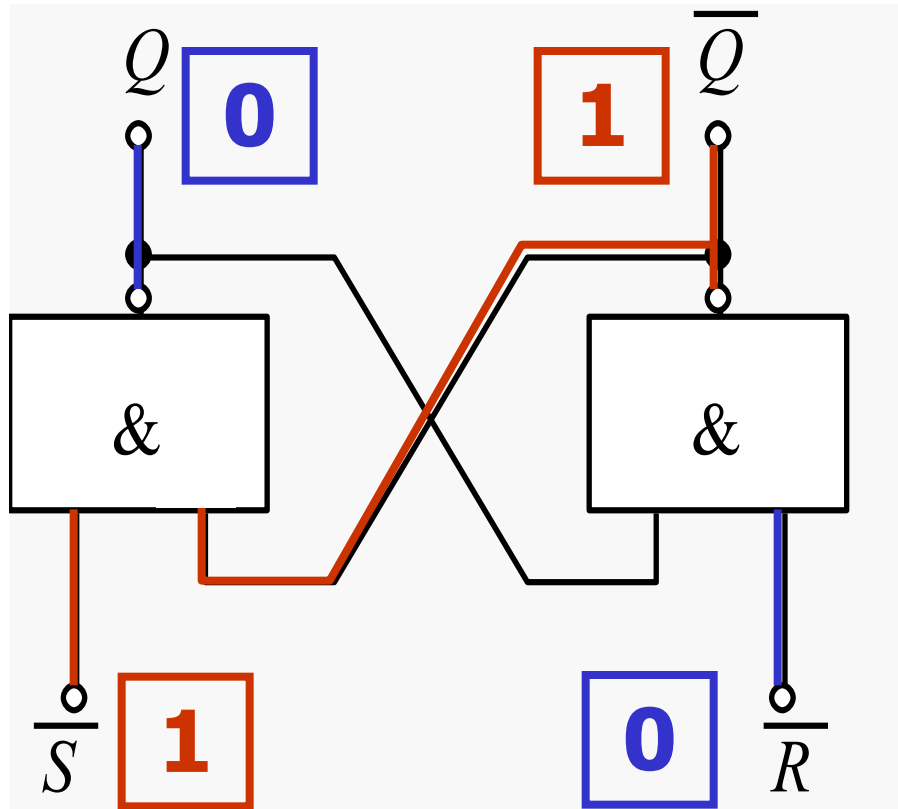


(a) 逻辑图



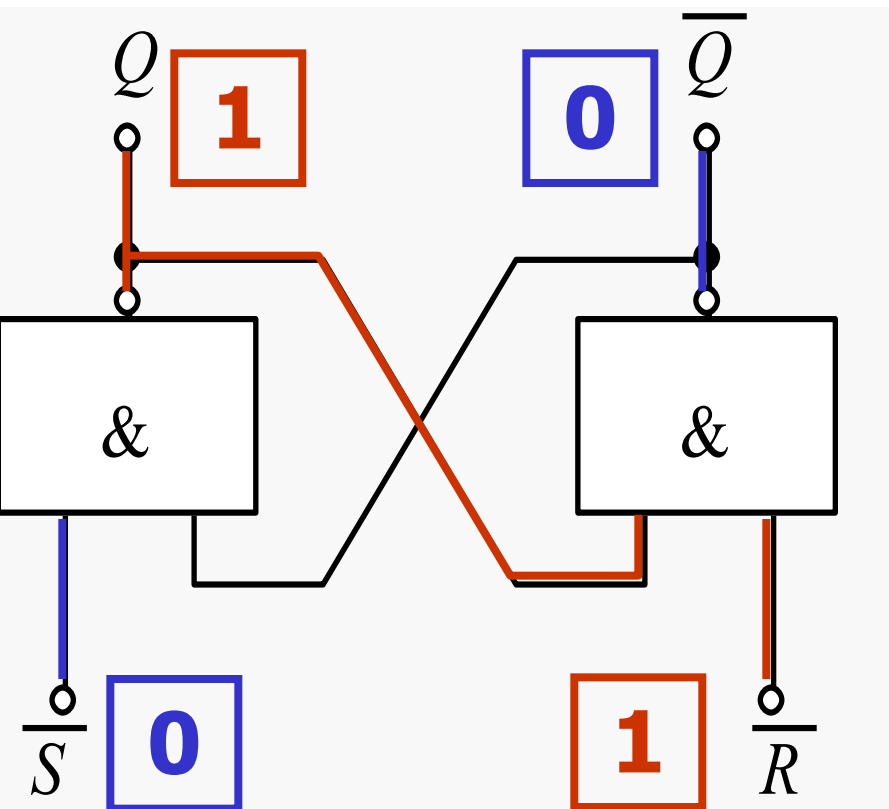
(b) 逻辑符号

自下而上：构建计算机体系



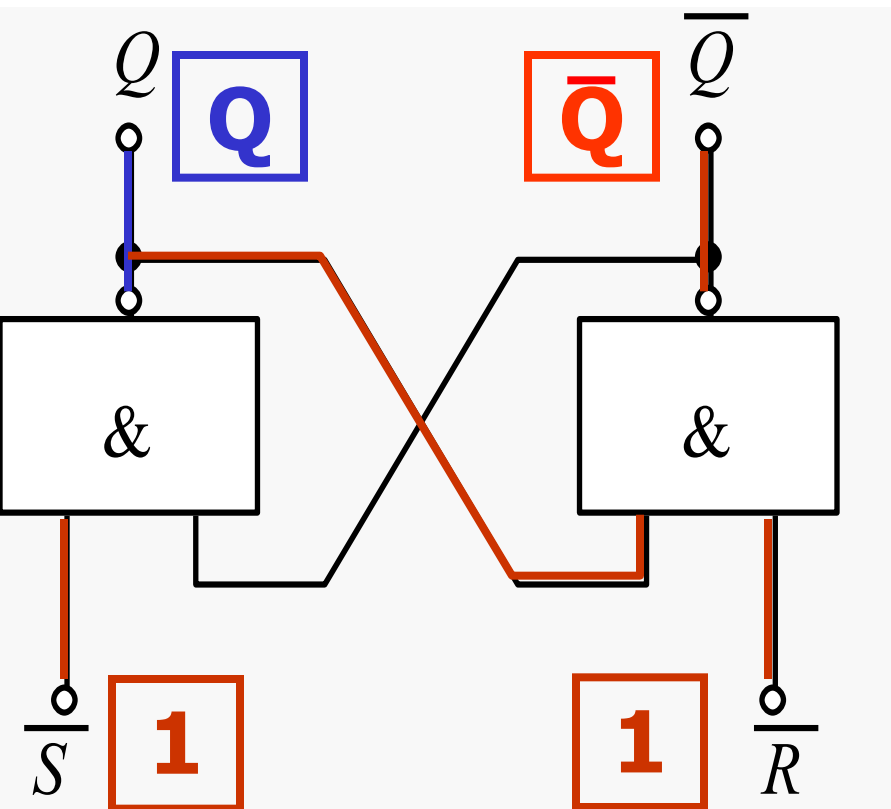
$\overline{R}$	$\overline{S}$	Q
0	1	0

自下而上：构建计算机体系



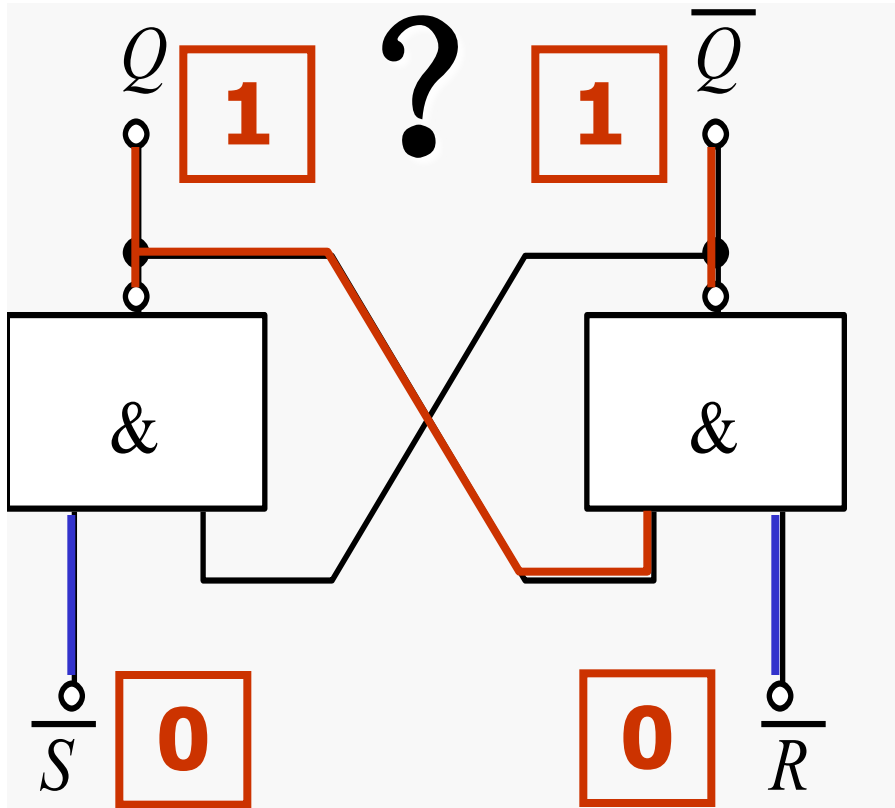
$\bar{R}$	$\bar{S}$	Q
0	1	0
1	0	1

自下而上：构建计算机体系



$\bar{R}$	$\bar{S}$	Q
0	1	0
1	0	1
1	1	不变

自下而上：构建计算机体系



$\bar{R}$	$\bar{S}$	Q
0	1	0
1	0	1
1	1	不变
0	0	不定

自下而上：构建计算机体系

现态：
状态，
也就是
触发器
接收输
入信号
之前的
稳定状
态。

$\bar{R}$	$\bar{S}$	Q^n	Q^{n+1}	功能
0	0	0	不用	不允许
0	0	1	不用	
0	1	0	0	$Q^{n+1} = 0$
0	1	1	0	置 0
1	0	0	1	$Q^{n+1} = 1$
1	0	1	1	置 1
1	1	0	0	$Q^{n+1} = Q^n$
1	1	1	1	保持

次态：
触发器
接收输
入信号
之后所
处的新
的稳定
状态。

自下而上：构建计算机体系

- 1949年在英国剑桥大学，计算机先驱Maurice Wilkes与EDSAC计算机内存的合影
- 图中内存叫做汞延迟线内存
 - 需要依赖比光速慢很多的音速实现存储功能
 - 在玻璃管的一端是扬声器，另一端则是麦克风。输入的数据脉冲会被转换成声音，然后借助汞传递到玻璃管的另一头，并被重新转换为电信号脉冲，进行整理，然后重新转换并通过环路发送出去。



自下而上：构建计算机体系

- 外存的存储原理

- 磁盘

- 磁化方向→感应电流

- 闪存（Flash）

- 是否捕获/捕获多少电子→电阻/电流

- 相变存储器（PCM）

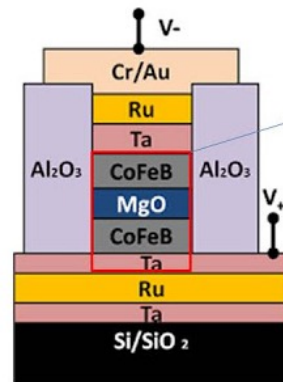
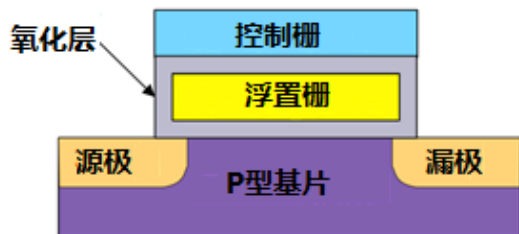
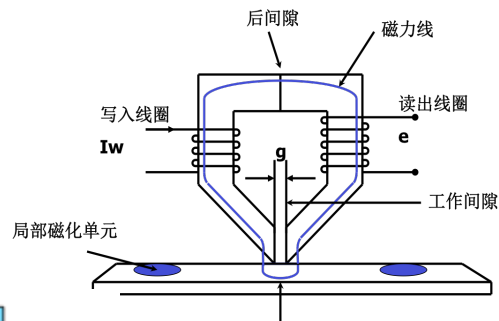
- 加热温度→结晶状态→电阻

- MTT-MRAM

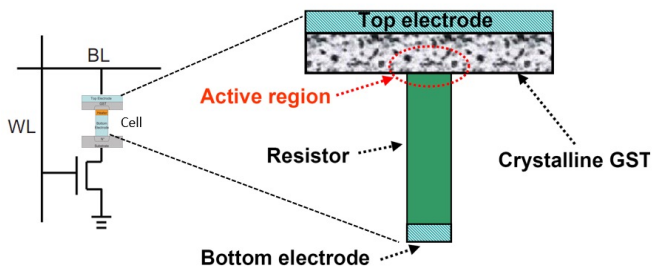
- 电子自旋



世界上第1块硬盘诞生于1956年：**IBM**公司的**305 RAMAC**



典型的四層 MTJ 結構。
由上而下依次為自由層、氧化層、固定層和鎖定位。

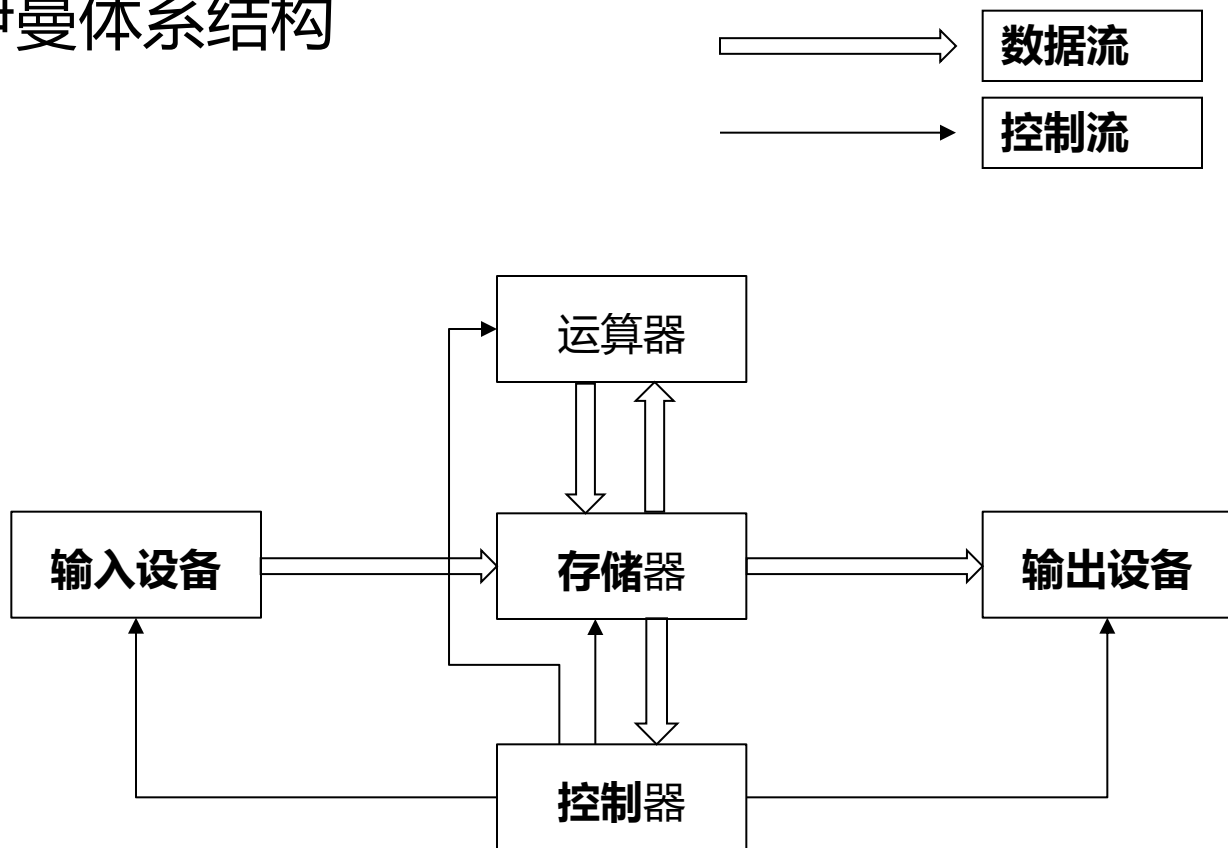


自下而上：构建计算机体系

- 3) 抽象：冯诺伊曼体系结构



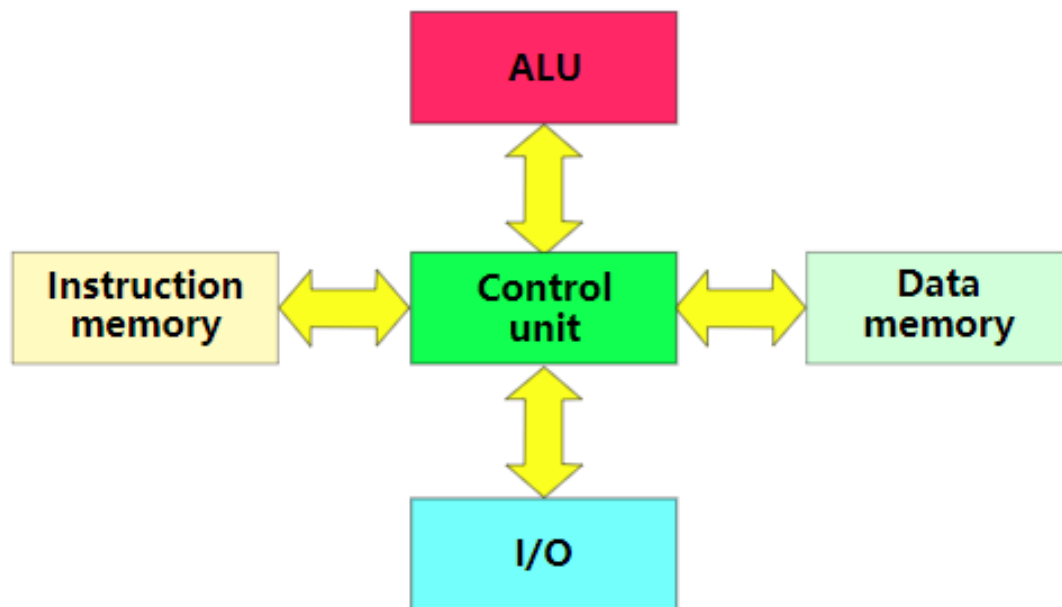
计算机之父——约翰·冯·诺依曼



自下而上：构建计算机体系

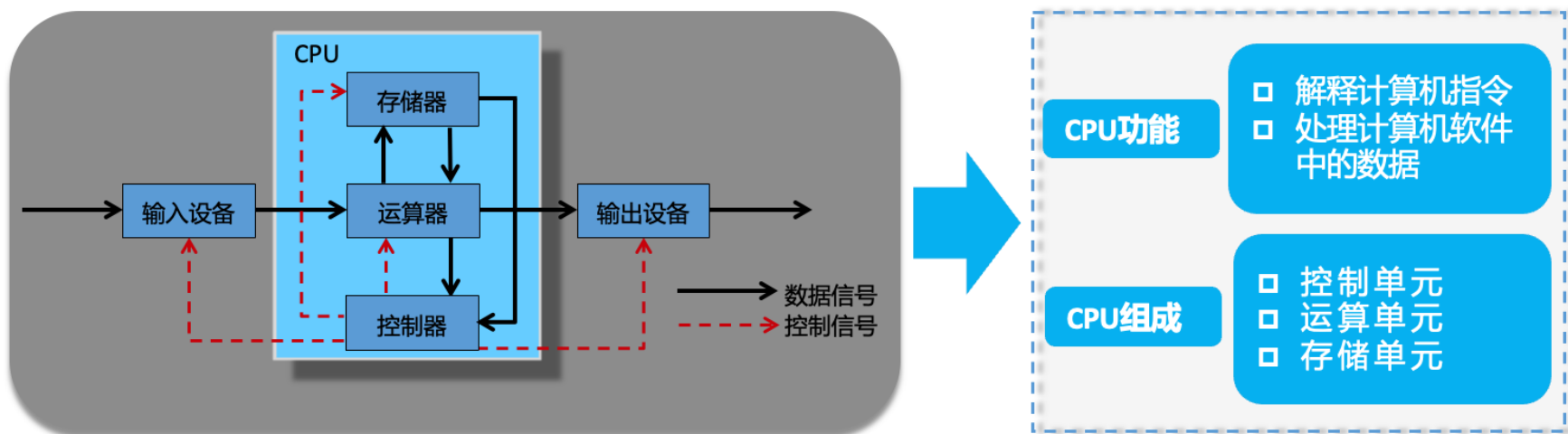
- 哈佛结构

- 哈佛结构是一种将程序指令存储和数据存储分开的存储器结构，它的主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个独立的存储器，每个存储器独立编址、独立访问，目的是为了减轻程序运行时的访存瓶颈。哈佛架构的中央处理器典型代表**ARM9/10**及后续**ARMv8**的处理器，例如：华为**鲲鹏920**处理器。



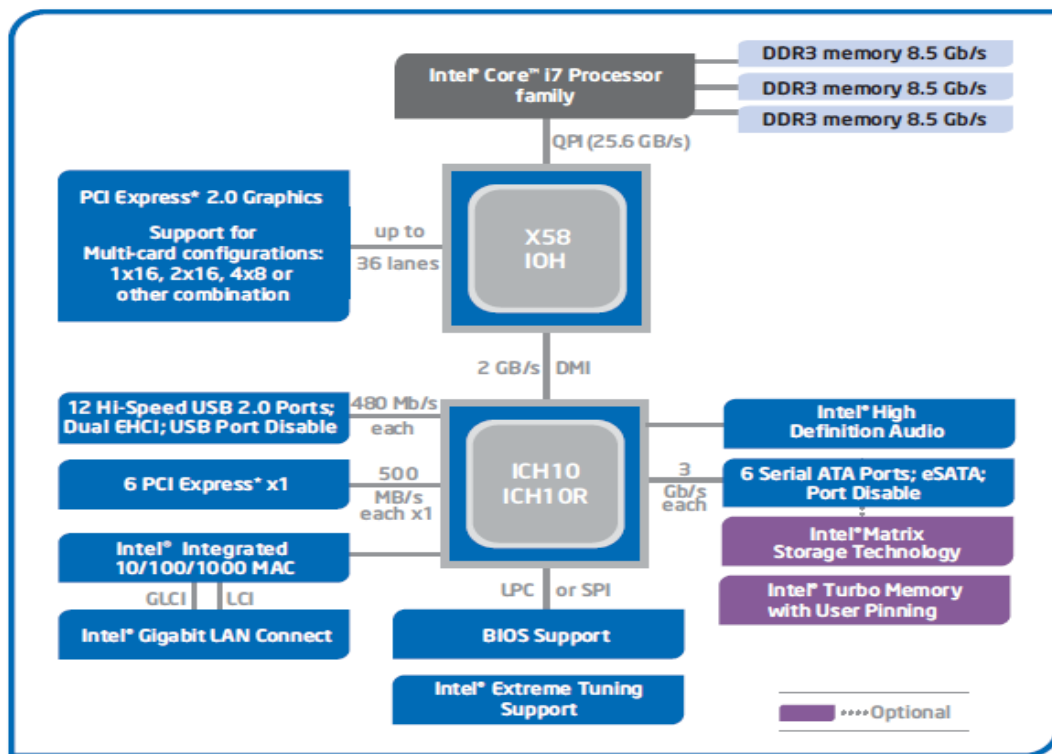
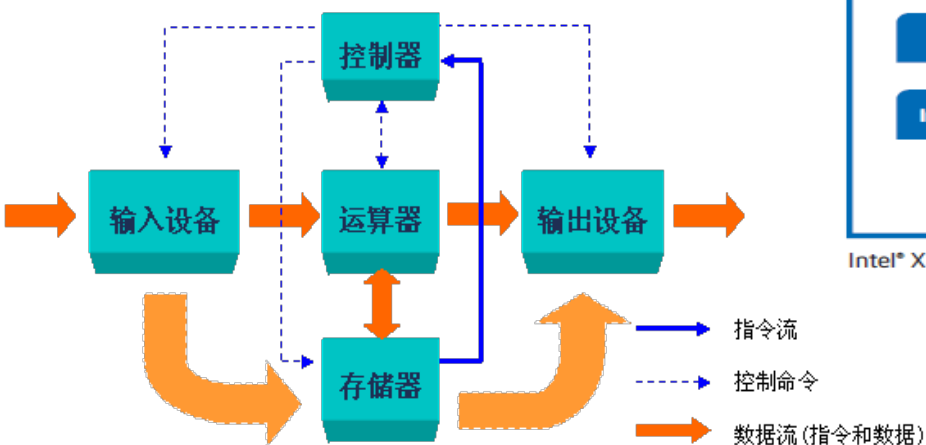
自下而上：构建计算机体系

- 中央处理器（Central Processing Unit, CPU）是计算机的主要设备之一，功能主要是解释计算机指令以及处理计算机软件中的数据；
- 中央处理器主要包括**控制器**，**运算器**，高速缓冲存储器（**Cache**）及实现它们之间联系的数据、控制及状态的总线；
- 它与主存（Memory）和输入/输出设备合称为电子计算机三大核心部件。



自下而上：构建计算机体系

- 运算器、控制器（寄存器、Cache）->CPU
- 存储器->内存
- 输入输出设备->磁盘、显示器、打印机、键盘鼠标、...

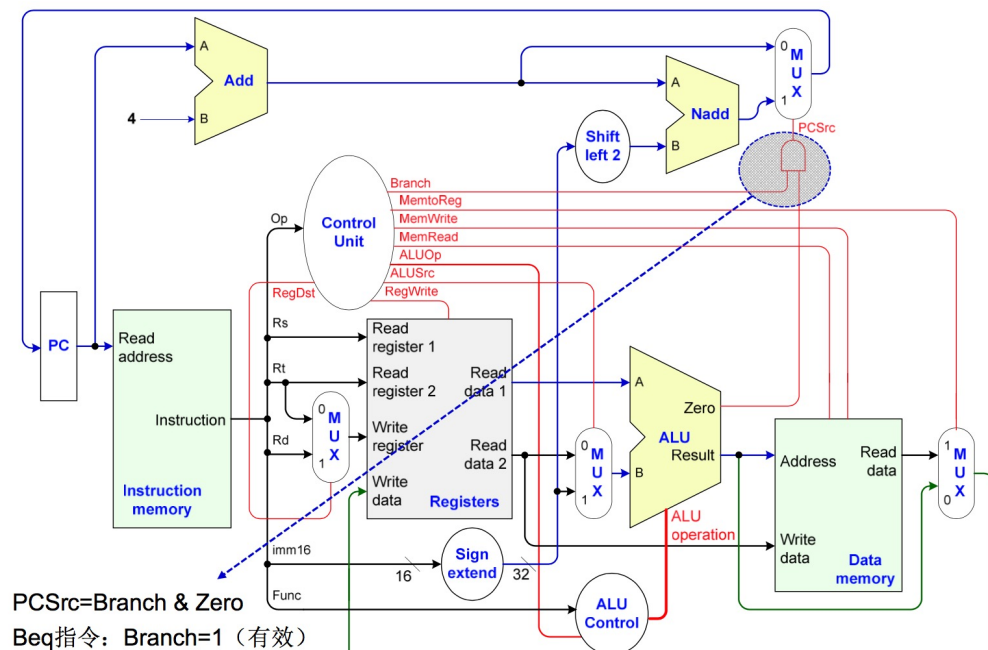


Intel X58 Express Chipset Block Diagram

自下而上：构建计算机体系

• CPU设计

- 出口：指令集（机器语言），如加法、乘法、跳转、过程调用等
- 运算器（ALU）
 - 组合逻辑电路，进行各种计算
- 控制器
 - 输入：当前指令、指令阶段
 - 输出：所有器件的控制信号
- 寄存器
 - 少量临时存储器件，寄存器 快于 Cache 快于 内存
 - 时序逻辑电路，具有记忆功能



自下而上：构建计算机体系

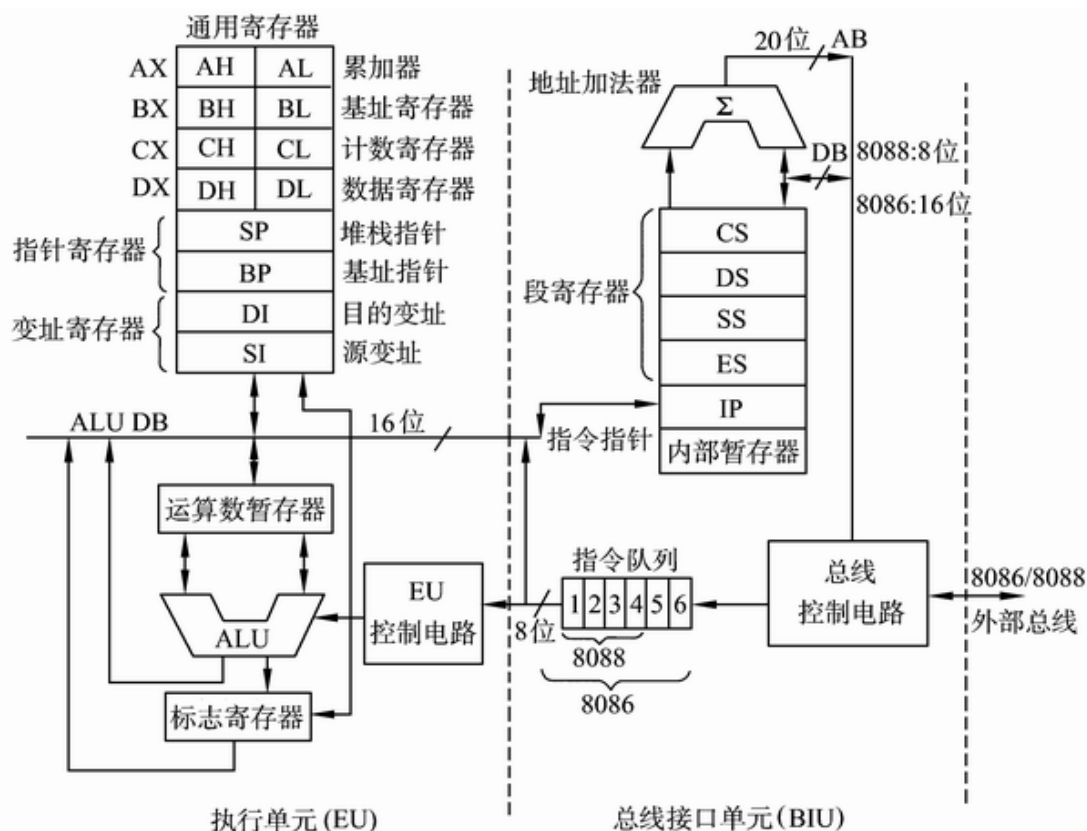
• CPU工作流程

```
While (1) {  
    Fetch_an_instruction();  
    Understand_the_instruction()  
    Execute_the_instruction();  
}
```

–Instructions:

- 移动数据(among register, memory, and I/O)
- 算数、逻辑、位运算
- 跳转类（指令跳转，函数调用等）

Intel 8086处理器的结构



自下而上：构建计算机体系

- 4) 硬件接口：机器语言/指令集
- 5) 软硬件分界面：汇编语言（汇编器）

```
long plus(long x, long y);

void sumstore(long x, long y,
              long *dest)
{
    long t = plus(x, y);
    *dest = t;
}
```

objdump -d sum

```
0000000000400595 <sumstore>:
 400595: 53                push    %rbx
 400596: 48 89 d3          mov     %rdx,%rbx
 400599: e8 f2 ff ff ff   callq   400590
<plus>
 40059e: 48 89 03          mov     %rax, (%rbx)
 4005a1: 5b                pop     %rbx
 4005a2: c3                retq
```

gcc -Og -S sum.c 生成汇编源文件sum.s

Outline

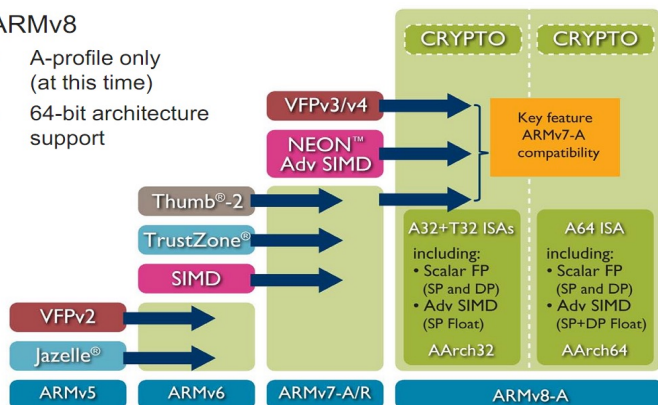
- CPU发展
- 自下而上：构建计算机体系
- **CPU Architecture**
- Memory and Registers
- 处理器架构实例：华为鲲鹏处理器

指令集：RISC vs CISC

- ARM：使用精简指令集（RISC），大幅简化架构，仅保留所需要的指令，可以让整个处理器更为简化，拥有小体积、高效能的特性。ARMv8架构支持64位操作，指令32位，寄存器64位，寻址能力64位。指令集使用NEON扩展结构。
- X86：使用复杂指令集（CISC），以增加处理器本身复杂度为代价，换取更高的性能。X86指令集从MMX，发展到了SSE，AVX。

ARMv8

- A-profile only (at this time)
- 64-bit architecture support



ARMv8架构的特点：

- 31个64位通用寄存器，原来架构只有15个通用寄存器；
- 新指令集支持64位运算，指令中的寄存器编码由4位扩充到5位；
- 新指令集仍然是32位，减少了条件执行指令，条件执行指令的4位编码释放出来用于寄存器编码；
- 堆栈指针SP和程序指针PC都不再是通用寄存器了，同时推出了零值寄存器（类似PowerPC的r0）；
- A64与A32的高级SIMD和FP相同；
- 高级SIMD与VFP共享浮点寄存器，支持128位宽的vector；
- 新增加解密指令。

ARM与X86架构关键差别：RISC与CISC指令集的差异

	RISC简单指令集（ARM）	CISC复杂指令集（X86）
吃饭的例子	指令：拿勺子->舀起一勺饭->送到嘴里->咽下去	指令：吃饭
优劣势	指令多，需要好几个指令实现一个复杂功能 针对常用指令优化，处理效率高 不常用指令则需要指令组合方式，可以通过流水技术和超标量技术弥补。指令执行周期<=1cycles	指令简单，不需要下达太多指令 指令动作复杂，速度慢；设计复杂，且流水中很多操作浪费 指令执行周期大都在5~8 cycles

处理器性能=

代码执行所需时间

$$\frac{\text{指令条数}}{\text{程序}}$$

代码数量
指令集&编译器

X

$$\frac{\text{Cycle数}}{\text{指令}}$$

指令执行周期
微架构

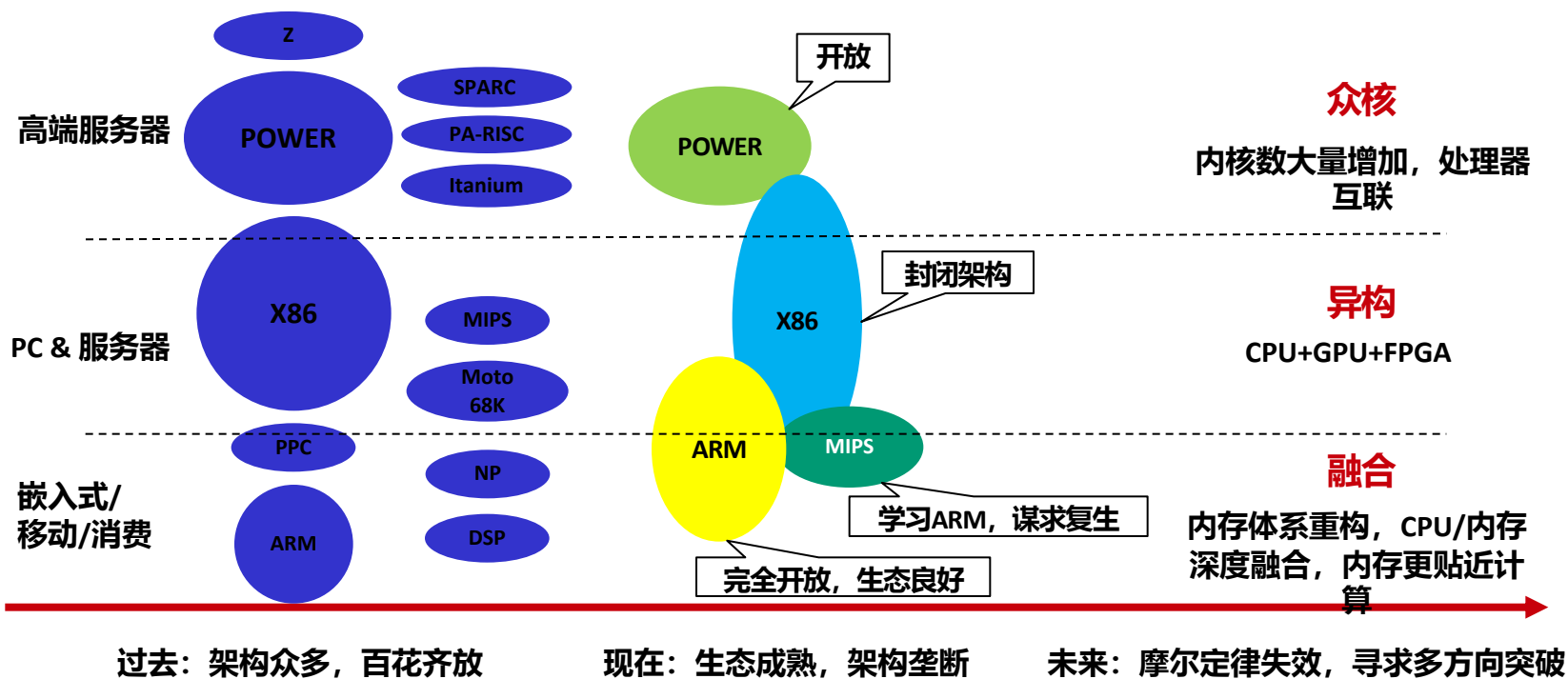
X

$$\frac{\text{时间}}{\text{Cycle数}}$$

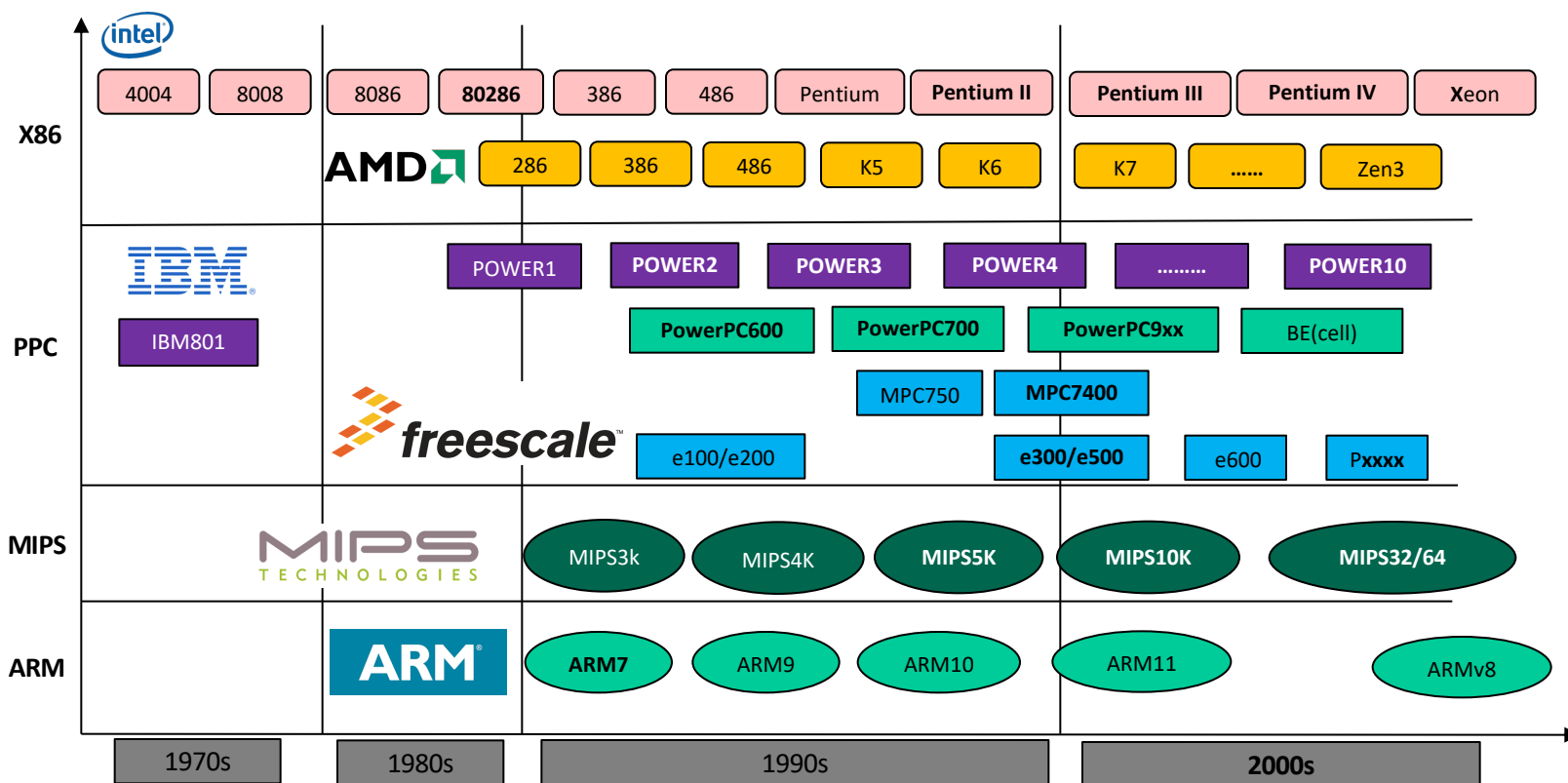
时钟周期
芯片实现

处理器性能和指令集&编译器、处理器微架构（流水）、架构（核数、工艺）均相关。通常所说的ARM处理器仅是指采用ARM指令集。即使均为ARM处理器，不同微架构，不同实现，芯片在性能、功耗方面的差异也非常大。

处理器发展趋势



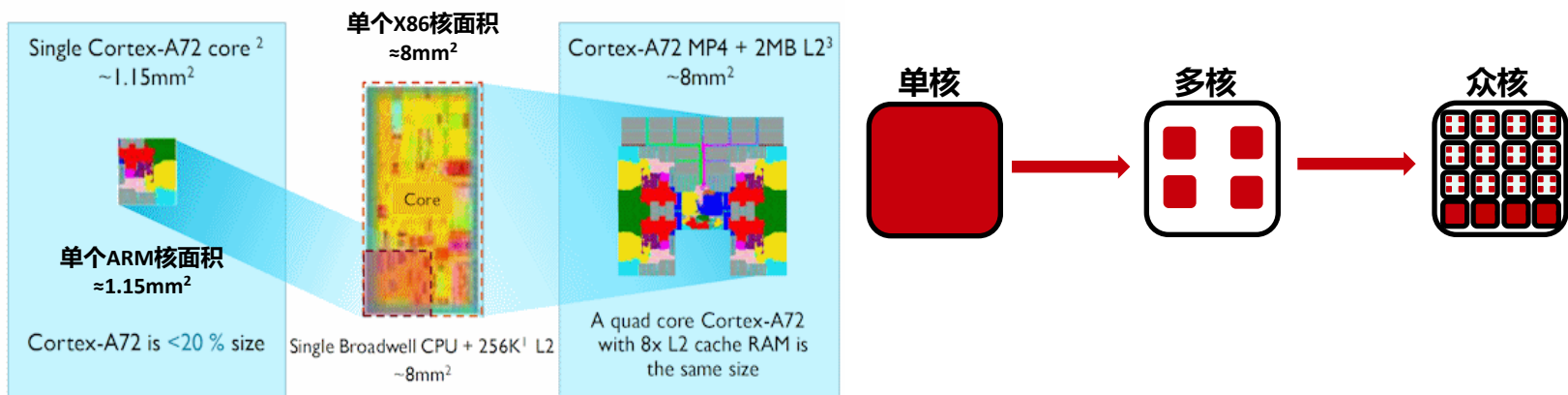
主流CPU发展路径



ARM提供更多计算核心

- 工艺、主频遇到瓶颈后，开始通过增加核数的方式来提升性能；
- 芯片的物理尺寸有限制，不能无限制的增加；
- ARM的众核横向扩展空间优势明显。

Cortex-A72: Ideal for dense compute environments



ARM服务器级别处理器一览



Hi1612

32C, 2.1GHz
16nm

Hi1616

32C, 2.4GHz
16nm

Hi1620

64C, 3.0GHz
7nm

CAVIUM

Thunder-X

48C, 2.5GHz
28nm

Thunder-X2

32-54C, 3.0GHz
14nm

高通

2400

48Cores, 2.2-2.6GHz
14nm

AMPERE

X-Gene3

32C, 3.3GHz
16nm

飞腾

FT1500

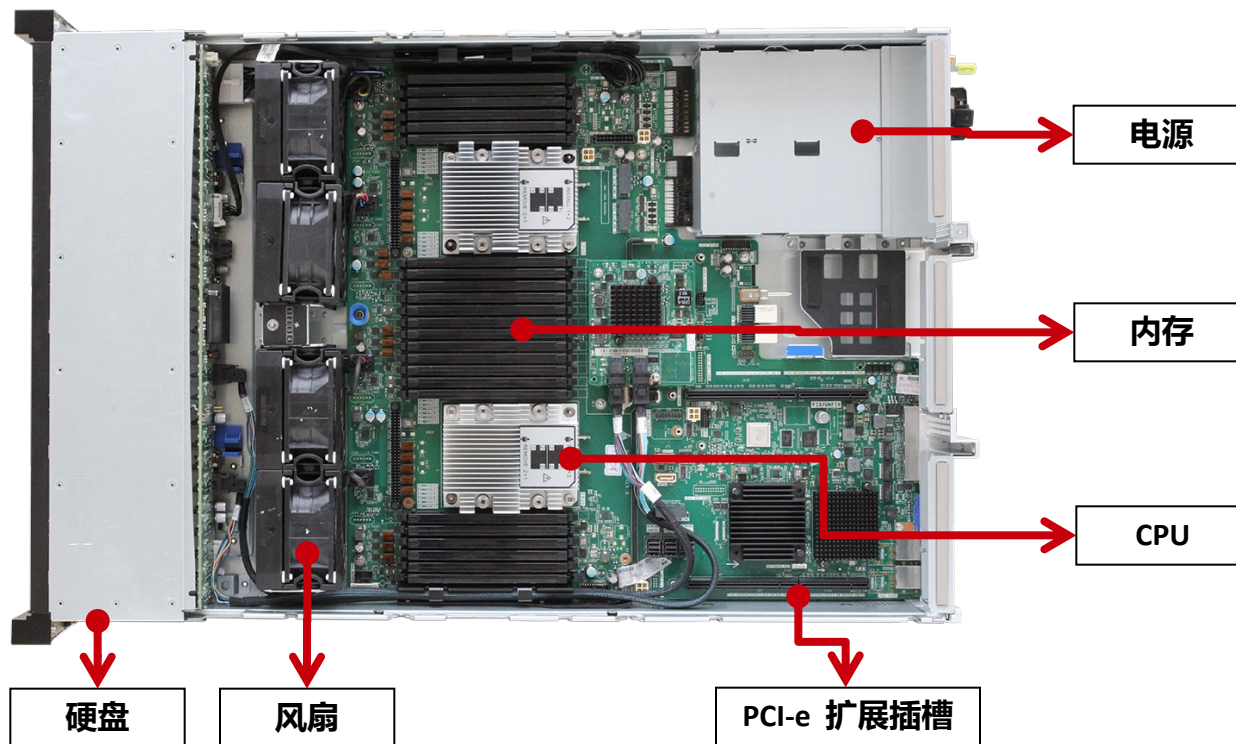
16 Cores, 1.6GHz
28nm

FT2000+

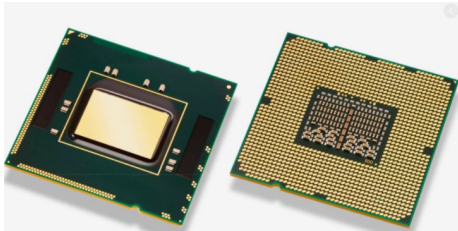
64Cores, 2.3GHz
16nm

横轴代表性能

计算机基础硬件 (1)



计算机基础硬件 (2)



CPU

中央处理器是一块集成电路，是一台计算机的运算核心和控制核心。



内存

内存是介于CPU 和外部存储之间，是CPU 高速运算时存放程序指令、数据和中间结果的临时场所，它的物理实质就是一组具备数据输入输出和数据存储功能的高速集成电路。



硬盘

常见的硬盘种类（按接口类型）：

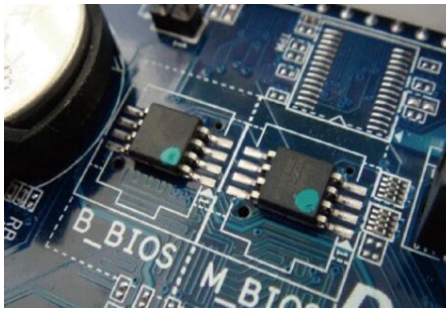
- SATA (Serial ATA)硬盘
- SAS (Serial Attached SCSI) 硬盘
- M.2 硬盘（介质为SSD）
- NVMe 硬盘（介质为SSD）

计算机基础硬件 (3)



PCIe卡

PCI Express接口, 简称PCIe, 是计算机总线的一种, 它沿用现有的PCI编程概念及通信标准, 基于更快的串行通信系统。主板上的PCIe插槽通过插接不同的扩展卡如GPU卡、RAID卡、网卡等获得额外的扩展。



BIOS

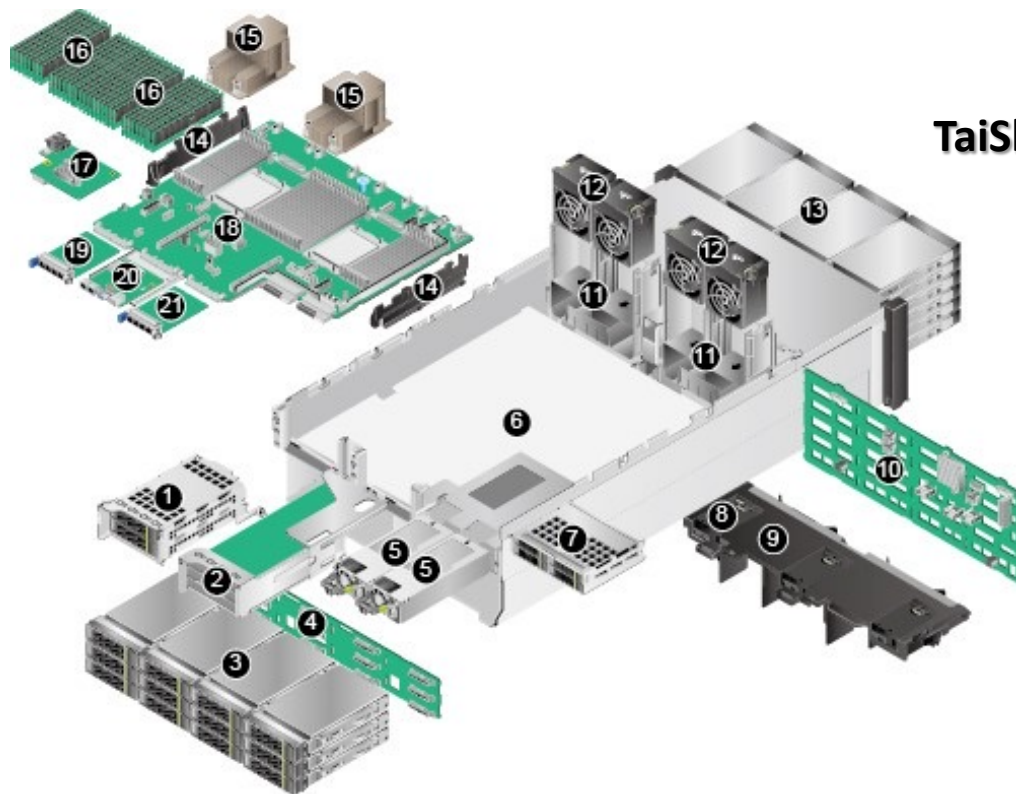
BIOS是Basic Input/Output System的缩写, 基本输入输出系统, 亦称为ROM BIOS、System BIOS, 是在通电引导阶段运行硬件初始化, 以及为操作系统和程序提供运行时服务的组件。



BMC

BMC是基板管理控制器 Baseboard Management Controller。它可以在机器未开机的状态下, 对服务器进行固件升级, 查看设备状态等操作。

TaiShan服务器物理结构

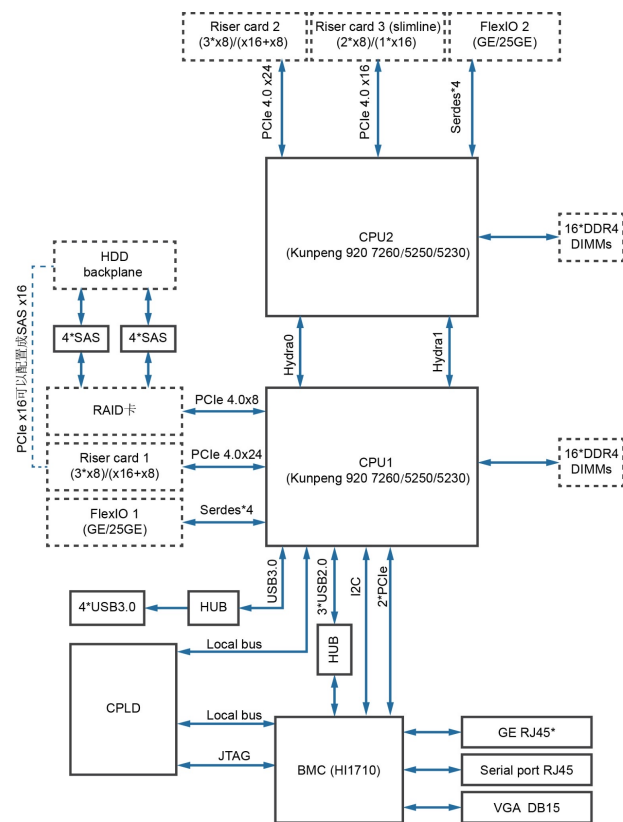


TaiShan 200服务器 (型号5280)

- | | | | |
|----|-----------------|----|--------|
| 1 | I/O模组1 | 2 | I/O模组2 |
| 3 | 后置硬盘 | 4 | 后置硬盘背板 |
| 5 | 电源模块 | 6 | 机箱 |
| 7 | I/O模组3 | 8 | 超级电容 |
| 9 | 导风罩 | 10 | 前置硬盘背板 |
| 11 | 风扇支架 | 12 | 风扇模块 |
| 13 | 前置硬盘 | 14 | 理线架 |
| 15 | 散热器 | 16 | DIMM |
| 17 | RAID控制扣卡 | 18 | 主板 |
| 19 | 灵活I/O卡1 (CPU 1) | 20 | iBMC插卡 |
| 21 | 灵活I/O卡2 (CPU 2) | - | - |

TaiShan服务器逻辑结构

- 支持两路华为自研鲲鹏920 7260、5250、5240或5230处理器，每个处理器支持16个DDR4 DIMM。
- 以太网灵活插卡可支持2种插卡，包括4*GE和4*25GE，通过CPU本身自带高速Serdes接口连接通信介质。
- RAID卡通过PCI-e总线跟CPU1连接，RAID卡引出的SAS信号线缆跟硬盘背板连接，通过不同的硬盘背板可支持多种本地存储规格。
- BMC使用华为自研管理芯片Hi1710，可外出VGA、管理网口、调试串口等管理接口。



Outline

- CPU发展
- 自下而上：构建计算机体系
- CPU Architecture
- **Memory and Registers**
- 处理器架构实例：华为鲲鹏处理器

Register

• 80386的寄存器

通用寄存器			
	31	16 15	0
EAX		AH	AL
EBX		BH	BL
ECX		CH	CL
EDX		DH	DL
ESP		SP	
EBP		BP	
ESI		SI	
EDI		DI	

指令指针和标志寄存器			
	31	16 15	0
EIP		IP	
EFLAGS		FLAGS	

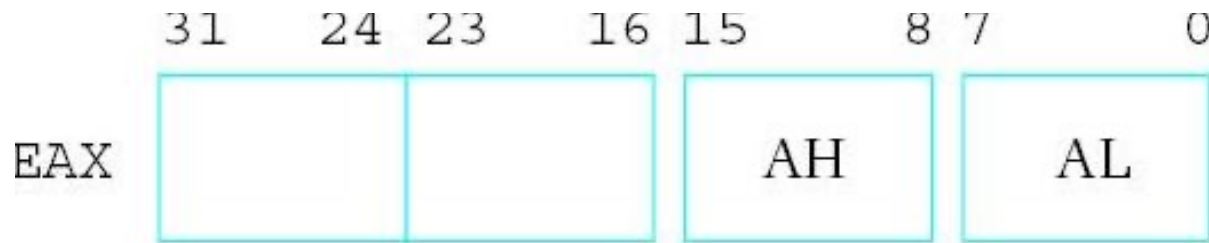
控制寄存器			
	31	16 15	0
CR0		AH	AL
CR1		BH	BL
CR2		CH	CL
CR3		DH	DL

段寄存器	
	15 0
CS	16位选择符
SS	16位选择符
DS	16位选择符
ES	16位选择符
FS	16位选择符
GS	16位选择符

段描述符寄存器		
	31 0	31 0
	32位段基址、20位段限、11位其它属性	
	32位段基址、20位段限、11位其它属性	
	32位段基址、20位段限、11位其它属性	
	32位段基址、20位段限、11位其它属性	
	32位段基址、20位段限、11位其它属性	
	32位段基址、20位段限、11位其它属性	

Register

- EAX, EBX, ECX, EDX



mov **eax**, 10

00	00	00	0a
----	----	----	----

mov **ax**, 10

XX	XX	00	0a
----	----	----	----

mov **ah**, 10

XX	XX	0a	XX
----	----	----	----

mov **al**, 10

XX	XX	XX	0a
----	----	----	----

Register

- EAX、EBX、ECX、EDX
 - 通用寄存器
- ESP, EBP, ESI, EDI
 - 32位通用寄存器
 - 各自的低16位可以独立使用：SP, BP, SI, DI
 - ESP：系统栈的栈指针寄存器
 - EBP：基址指针寄存器
 - ESI、EDI：索引寄存器，从ESI指定位置拷贝到EDI指定位置

Register

- 标志寄存器 **EFLAGS**

31...18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	VM	RF		NT	IOPL	OF	DF	IF	TF	SF	ZF		AF		PF		CF	

Bit	Mnemonic	Usage
0	CF	carry flag
2	PF	parity flag
6	ZF	zero flag
7	SF	sign flag
10	DF	direction flag
11	OF	overflow flag

Register

- 64位计算机
 - X86-64指令集扩展了IA32的32位寄存器
 - 用'r'代替'e', %eax->%rax, %esp->%rsp
 - 增加了8个寄存器
 - %r8~%r15
 - 浮点数XMM寄存器
 - %xmm0~%xmm15, 128位, 能放下4个float, 或2个double

Memory

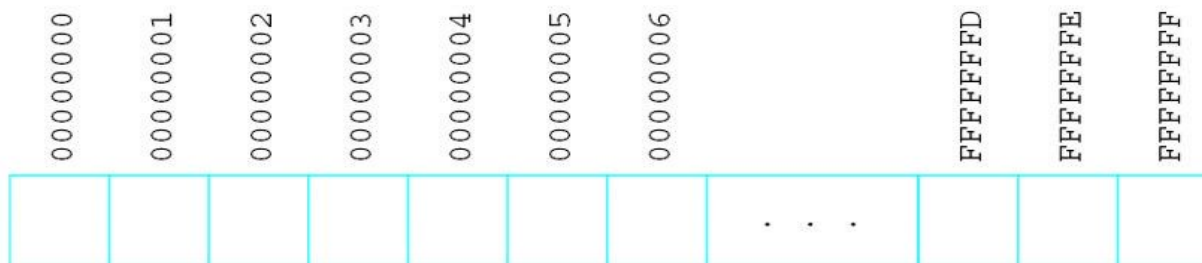
- 8086~80286

- 实模式

- 直接访问物理地址，处理器只能寻址 **1 MB** 的内存空间
 - 处理器一次只能运行一个程序，随时可被中断

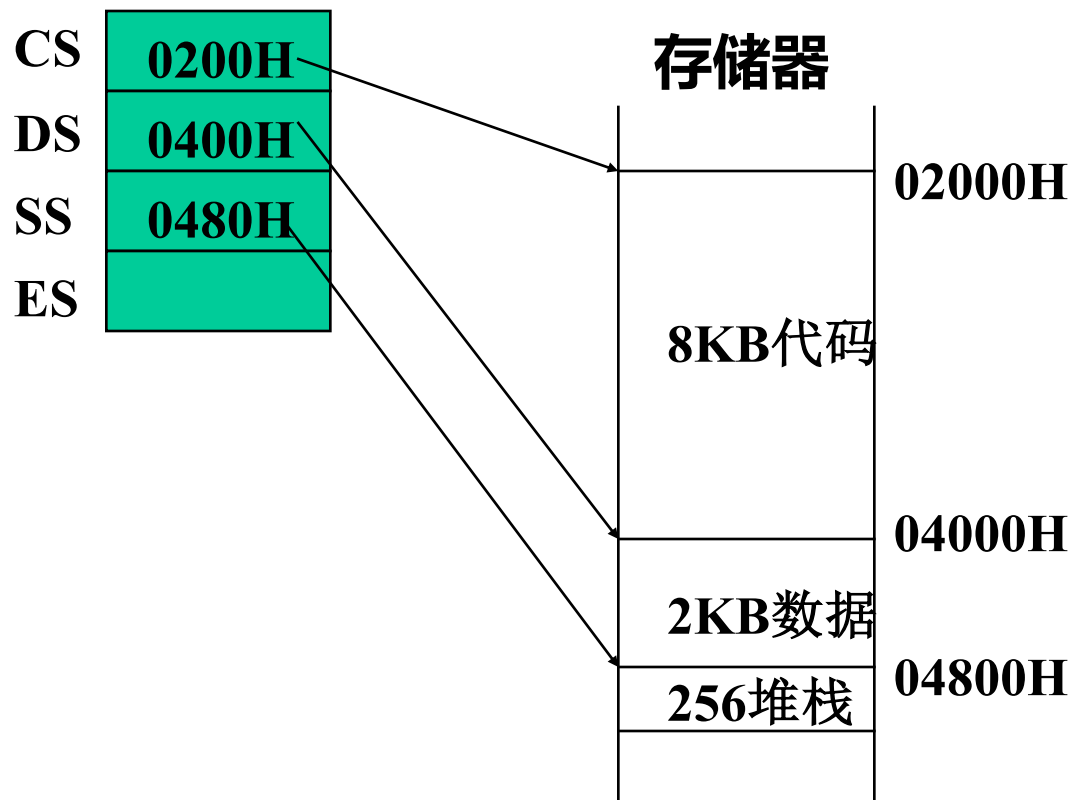
- 80386

- 实模式：80386提供实模式来兼容先前的8086
 - 保护模式：程序访问虚拟地址，OS内核解析为物理地址



Memory

- 段寄存器（为什么内存要分段？）
- 代码段**CS**
 - 对应执行的指令，起始地址对应第一条可执行的指令
- 数据段**DS**
 - 对应程序的数据、常量等
- 堆栈段**SS**
 - 对应暂存任何数据，包括子程序所用的数据
- 扩展段**ES**
 - 对应于特殊的数据和操作使用



Memory

- 内存分段管理

- 由于**8086**中的地址寄存器都是**16**位的，用户不能直接使用**20**位的物理地址，编程时需要使用**逻辑地址**来寻址存储单元。
- 逻辑地址由两个**16**位数构成，其形式为：
段的起始地址 ： 段内的偏移地址
(**16**位段地址) ： (**16**位偏移量)

Memory

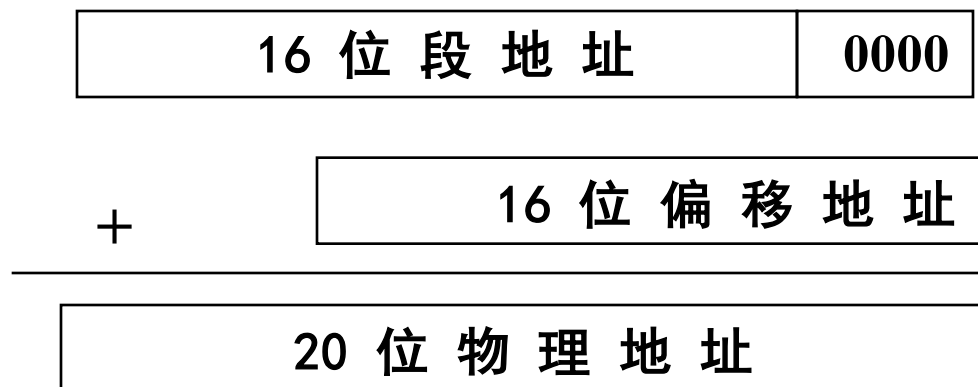
- 存储空间分为多个**逻辑段**（段——Segment）：
 - 段的**20**位的起始地址（**xxxxxH**）其低**4**位必须为**0**（**xxxx0H**），所以可以将它们省略，然后用**1**个**16**位数来表示段的首地址。
 - 每段长度限 $2^{16}=64\text{KB}$ ，所以段内偏移地址可以用**1**个**16**位数来表示（**xxxxH**）；
 - 所以有：
段的起始地址 ： **段内的偏移地址**
（16位段地址） ： **（16位偏移量）**

Memory

- 物理地址和逻辑地址
 - 每个存储单元都有一个唯一物理地址（00000H～FFFFFH），20位，该地址在指令执行时由地址加法器形成，并进行硬件寻址。
 - 地址加法器的具体做法：段地址左移4位，然后加上偏移地址就得到20位物理地址。
 - 用户编程时采用逻辑地址，其形式为：
段的首地址：段内偏移地址
它们由两个16位的无符号数构成。
 - 逻辑地址“1460H:100H” = 物理地址14700H
 $14600H + 100H = 14700H$

Memory

逻辑地址 **vs.** 物理地址



Memory

- 段寄存器和逻辑段
 - 典型的程序有三个段：代码段、数据段和堆栈段
 - 8086有4个16位的段寄存器：
 - CS**（代码段寄存器）用来指明代码段的首地址
 - SS**（堆栈段寄存器）指明堆栈段的首地址
 - DS**（数据段寄存器）指明数据段的首地址
 - ES, FS, GS**（附加段寄存器）指明附加段的首地址

Memory

- CS: $IP = CS * 16 + IP$, 当前指令地址
- DS: $[x] = DS * 16 + x$, 当前数据地址
- SS: $SP = SS * 16 + SP$, 当前栈顶地址
 - 需自己保证push和pop不越界
- 80x86不允许直接将立即数送入段寄存器
 - `Mov ds, 1000H`非法
 - `Mov ax, 1000H` `mov ds, ax`合法

程序员可见状态

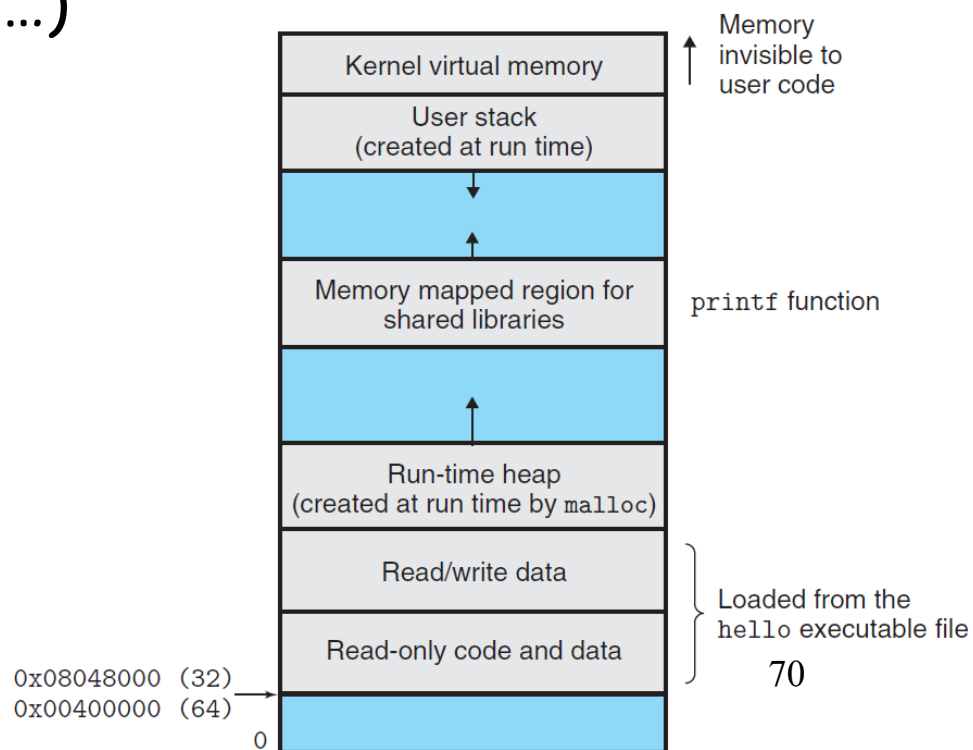
- C语言程序员
 - C语句
 - 各种计算
 - 程序流程（循环、分支、函数调用等）
 - 常量、变量（内存）//寄存器不可见
 - 全局变量/静态变量：数据段
 - 局部变量：堆栈段（注意作用域）
 - 操作系统提供的系统调用
 - **Malloc**（堆内存），**free**
 - **Printf**等，控制外设（输入输出设备）
 - **Read/write**，访问文件系统（**I/O**设备）
 - **Send/receive**，访问网卡

程序员可见状态

- 汇编程序员
 - 汇编指令
 - 各类运算
 - 跳转指令（`jmp`, `jcc`->分支、循环；子过程调用`call`）
 - 中断指令（`int`, 与外设打交道/进行系统调用）
 - 寄存器
 - `Eax`, `ebx`, ...等通用和专用寄存器
 - 内存
 - 全局变量：放在数据段
 - 堆栈段：可以直接操作（`push`, `pop`, `move`, ...）
 - 函数调用时自动操作堆栈（保存返回地址等）

程序员可见状态

- 寄存器
 - %rax, %rbx, %rcx, %rdx, %rbp, %rsp, %rsi, %rdi
 - %r8, %r9, %r10, %r11, %r12, %r13, %r14, %r15
 - EFlags (SF, ZF, OF, CF, ...)
 - CS, DS, SS, ES, FS, GS
 - PC (IP)
 - Memory
 - 8086: 物理内存
 - 80386+: 虚拟内存地址



程序员可见状态

x86-64 Integer Registers

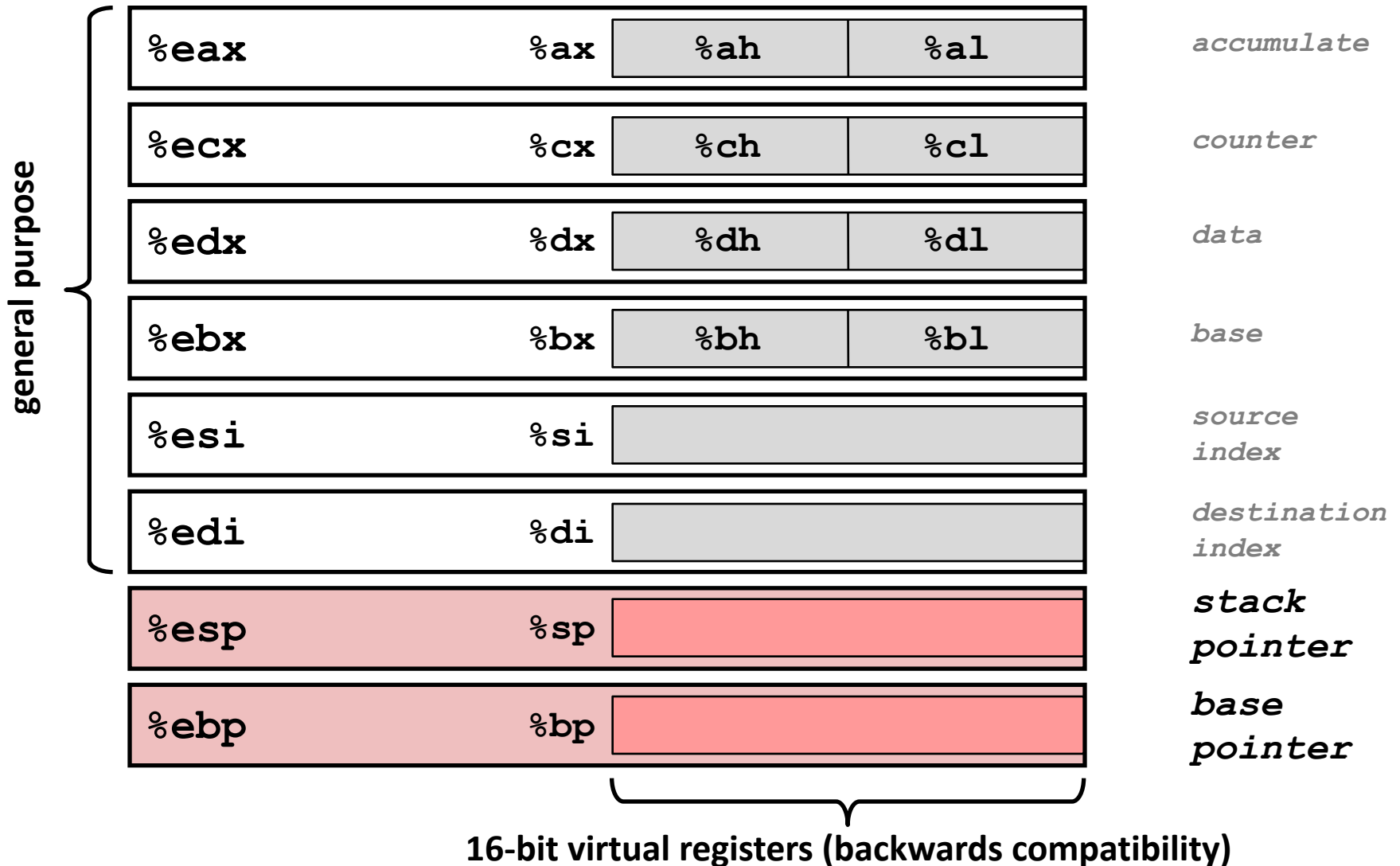
%rax	%eax
%rbx	%ebx
%rcx	%ecx
%rdx	%edx
%rsi	%esi
%rdi	%edi
%rsp	%esp
%rbp	%ebp

%r8	%r8d
%r9	%r9d
%r10	%r10d
%r11	%r11d
%r12	%r12d
%r13	%r13d
%r14	%r14d
%r15	%r15d

- Can reference low-order 4 bytes (also low-order 1 & 2 bytes) ⁷¹

Some History: IA32 Registers

Origin
(mostly obsolete)



8086计算机上，如果希望能够寻址到物理地址0x54000，请问段地址的最大、最小分别是多少？使用十六进制表示

最大：0x [填空1]

最小：0x [填空2]

内存中存放的机器码和对应的汇编指令如下图所示，设CPU初始状态： $CS=2000H$ ， $IP=0000H$ ，请写出指令执行序列。

地址	内存	对应汇编指令	地址	内存	对应汇编指令
10000H	B8	mov ax, 0123H	20000H	B8	mov ax, 6622H
	23			22	
	01			66	
10003H	B8	mov ax, 0000	20003H	EA	jmp 1000:3
	00			03	
	00			00	
10006H	8B	mov bx, ax		00	
	D8			10	
10008H	FF	jmp bx	20008H	89	mov cx, ax
10009H	E3			C1	

练习答案

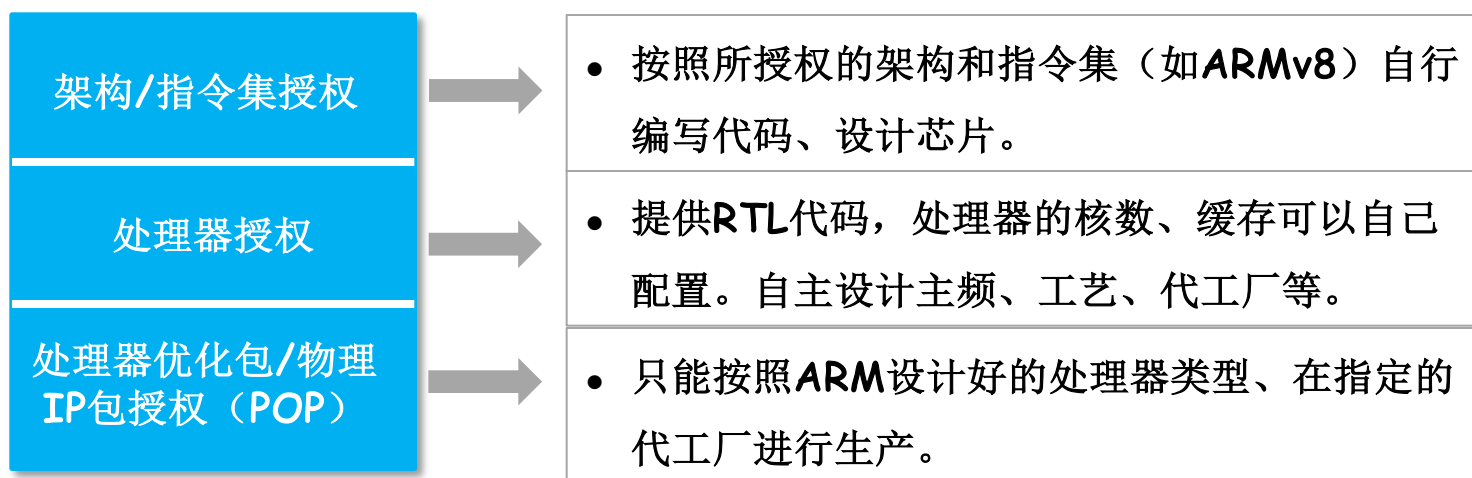
- `Mov ax, 6622H`
- `Jmp 1000:3`
- `Mov ax, 0000`
- `Mov bx, ax`
- `Jmp bx`
- `Mov ax, 0123H`
- →第3步

Outline

- CPU发展
- 自下而上：构建计算机体系
- CPU Architecture
- Memory and Registers
- 处理器架构实例：华为鲲鹏处理器

ARM公司授权体系

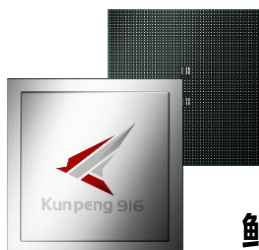
- ARM 目前在全球拥有大约1000个授权合作商、320家伙伴，但是购买架构授权的厂家不超过20家，中国有华为、飞腾获得了架构授权。



ARM服务器处理器的优势

- 低功耗一直以来都是ARM架构芯片最大的优势；
- ARM架构的芯片在成本、集成度方面也有较大的优势；
- 端、边、云全场景同构互联与协同；
- 更高的并发处理效率；
- 多元化的市场供应。

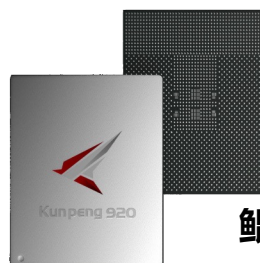
鲲鹏处理器



鲲鹏916

支持多路互联的ARM处理器

- 32核, 2.4 GHz主频
- SPECint性能匹配业界中端, 功耗低至75 W
- 支持4通道DDR4控制器
- 支持PCI-e 3.0和SAS/SATA 3.0
- 集成板载GE/10 GE网络
- 支持2路互联



鲲鹏920

7nm制程, 数据中心ARM处理

- 计算核数提升1倍, 最多64核
- SPECint性能提升超过2倍
- 内存通道数提升1倍, 支持8通道DDR4控制器
- 支持PCI-e 4.0和CCIX
- 集成板载100 GE网络和加密、压缩加速引擎
- 支持2路或4路互联

基于鲲鹏920的华为TaiShan 200服务器



17年的工程工艺积累

40度以上异常高温运行

液冷
散热

无源背板&三重硬盘抗震

高速
互联

可靠
设计

质量
品控

56G/112G板级高速互联

故障率低于业界15%

数据中心

存储密集型

5290
4U72盘存储型



(新产品)

5280
4U40盘存储型



2280
2U2路均衡型



1280
1U2路高密度



(新产品)

计算密集型

2480
2U4路高性能型



(新产品)

X6000
2U4节点高密度



边缘计算

2280E
边缘型



(新产品)

学习推荐

- 华为官方网站

- 在线学习:

- <https://e.huawei.com/cn/talent/#/home>

- 企业业务: <http://enterprise.huawei.com/cn/>

- 技术支持:

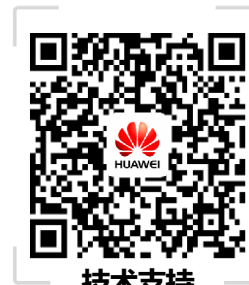
- <http://support.huawei.com>

- 热门工具

- HedEx Lite

- 网络资料工具中心

- 信息查询助手



技术支持



热门工具