

北京科技大学计算机与通信工程学院

计算机体系结构

Computer Architecture

马惠敏 教授

助教： 储华珍 梅若恒

北京科技大学机电信息楼 702A, 605
62334122, 13810187014 mhmpub@ustb.edu.cn

北京科技大学计算机与通信工程学院

计算机体系结构

Computer Architecture

马惠敏 教授

助教： 储华珍 梅若恒

北京科技大学机电信息楼 702A, 605
62334122, 13810187014 mhmpub@ustb.edu.cn

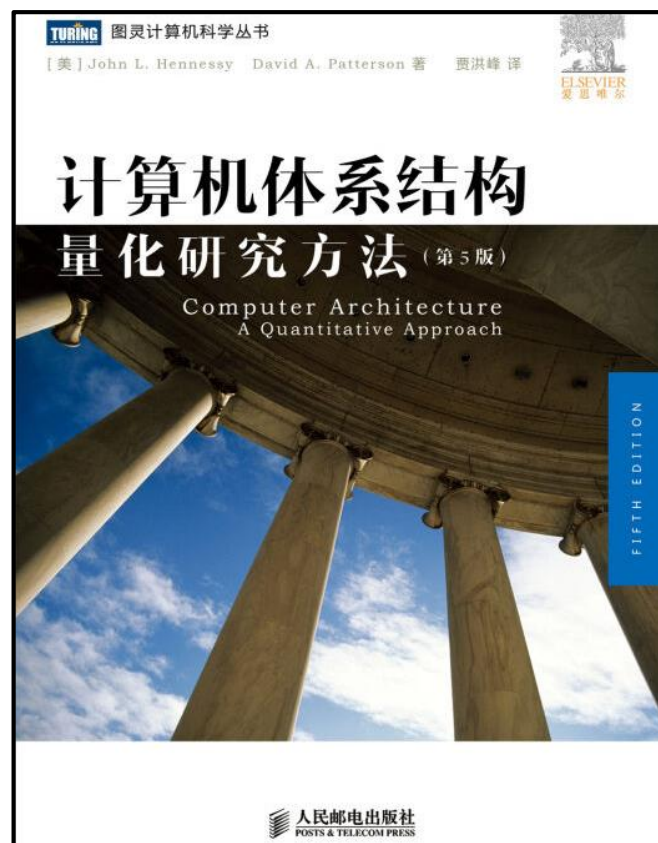
计算机体系结构



- 计算机体系结构：是程序员看到的计算机属性，即概念性结构与功能特性
- 在计算机设计中，一种重要的抽象就是“**硬件与软件之间的接口**”——被赋予特殊的名称：机器指令集体系结构(instruction set architecture, or simply architecture-系统结构)

计算机体系结构

- 指令体系结构允许计算机设计者脱离执行它们的硬件来讨论功能，使得系统设计与实现区分开



课程介绍

1. 课程名称： 计算机体系结构

Computer Architecture

Instruction Set
Architecture

+

Computer
Machine Organization

指令集体系结构

计算机组成+硬件

课程介绍

2. 研究内容

计算机用户所看到的物理计算机的抽象：编写出能在机器上正确运行的程序所必须了解的**计算机属性**

3. 学习目的

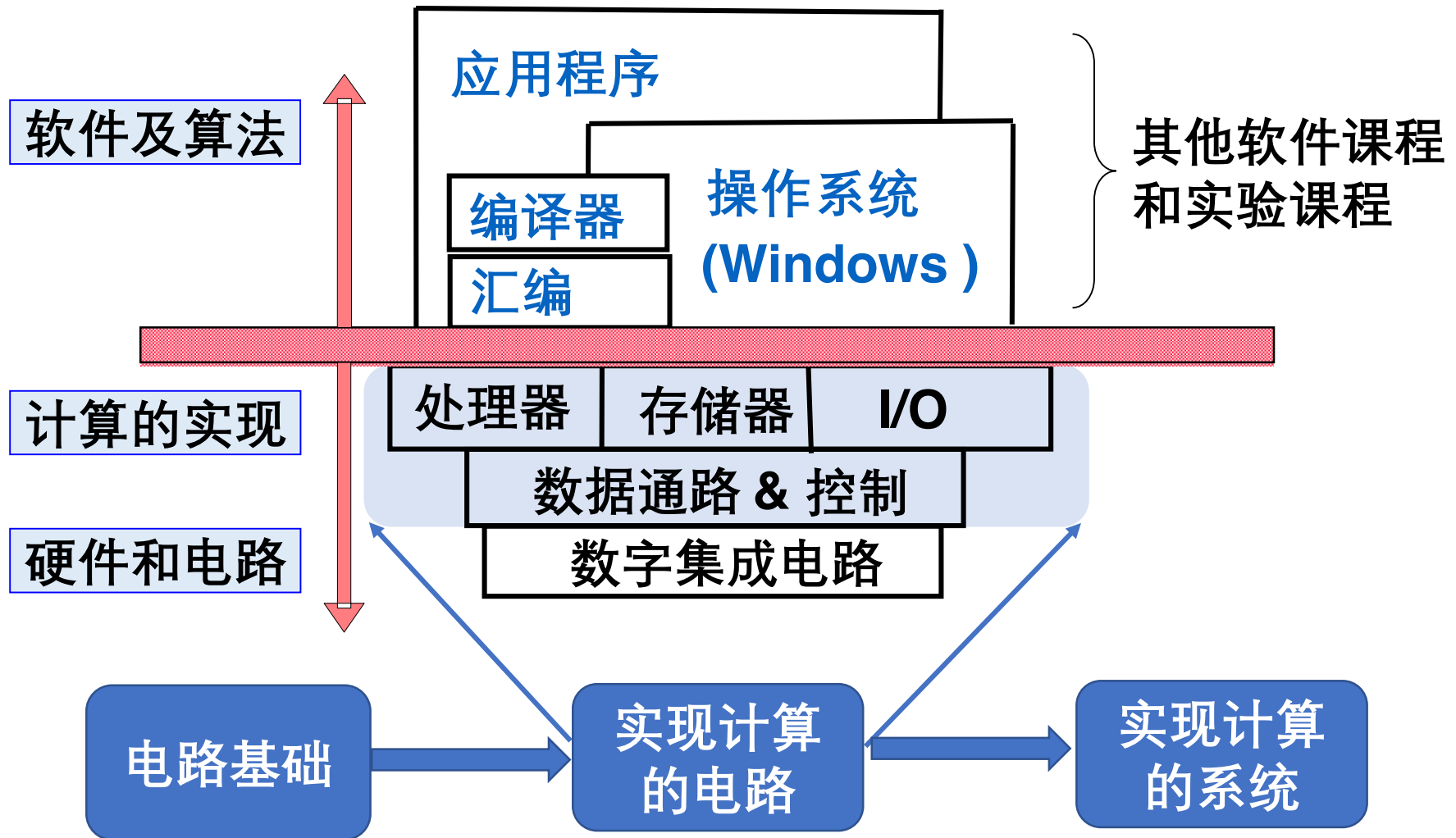
- 建立计算机体系结构的**完整概念**

系统观：计算机系统内部各软件/硬件部分的关联关系与逻辑层次

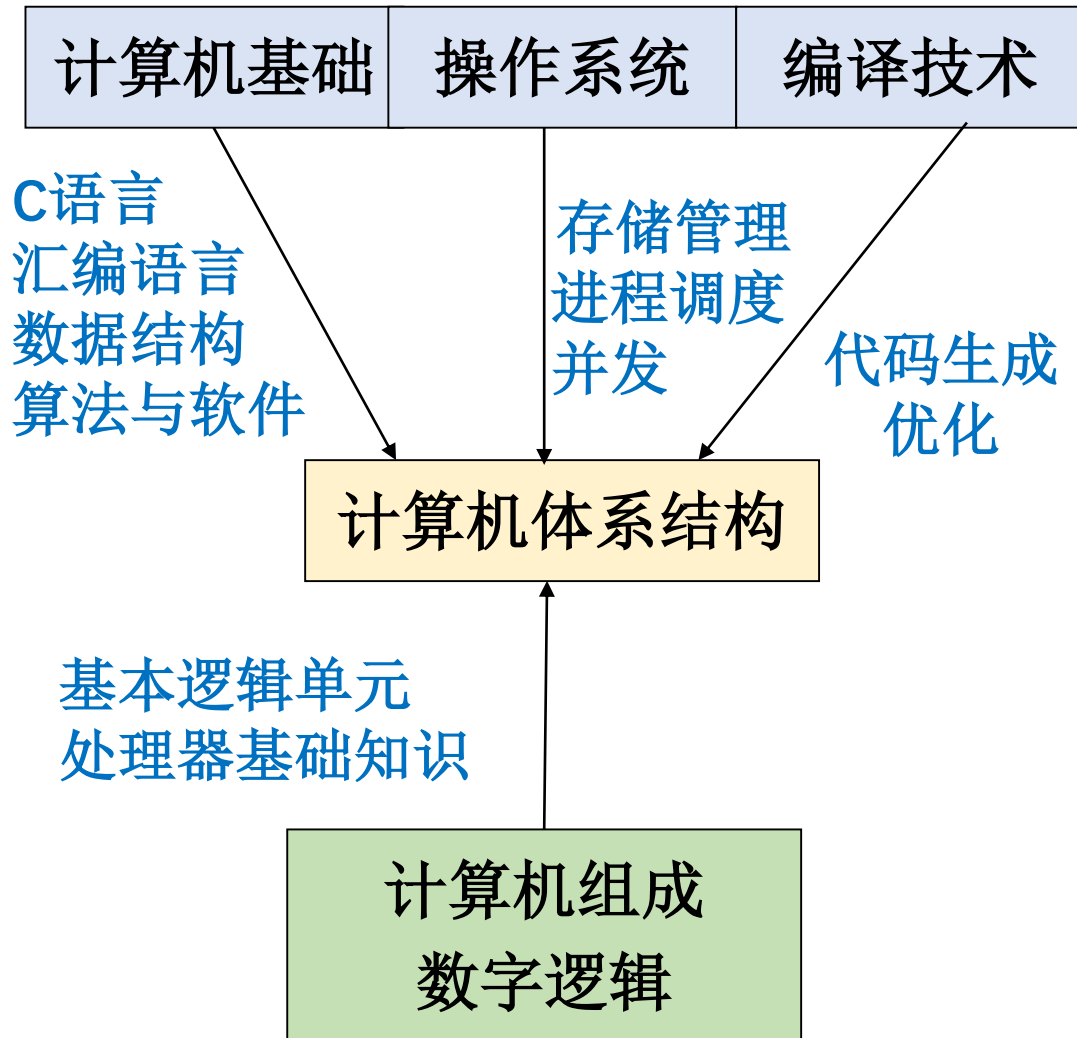
- 学习计算机体系结构的**分析方法和设计方法**

在掌握基本系统原理的基础上，进一步掌握设计、实现计算机硬件、系统软件和应用系统的综合能力。

计算机相关课程体系



计算机体系结构及相关课程



- ✓ 计算机应用需要什么?
- ✓ 操作系统需要哪些功能支持?
- ✓ 优化编译可以利用和实现哪些功能?
- ✓ 我们能够建造什么样的机器?
- ✓ 今后的计算机将会如何?

计算机基础	操作系统	编译技术
-------	------	------

● 计算机基础

```

a = 0;
b = a;
res = 1;
i = x;
while ( --i != 0 )
{
    a=b;
    b=res;
    res=a+b;
}
f = res;
    
```

学期					
英语(1)	高数A(1)	线代	工程制图	高级语言	I
英语(2)	高数A(2)		电路原理	物理A(1)	II
英语(3)			模拟电路	物理A(2)	III
英语(4)	复变函数 概率统计 随机过程		数字电子	汇编语言×	IV
		信号与系统		EDA C#程序语言	V
	信息概论 控制工程	图形学	计算机组成 微机与接口	数据结构	
	人工智能	模式识别 信号处理	系统结构	通信原理 计算机网络	VI
		图像处理	编译原理 嵌入式 并行计算	VLSI 操作系统 数据库	VII
	计算方法		网络与安全	JAVA 软件工程 电子商务	
论文训练					VIII

处理器的架构

- 中央处理器 (Central Processing Unit)
 - 用于解释计算机指令与处理数据
 - 一般的程序中常见操作：
 - 算数运算、逻辑运算、判断比较、数据位移
 - 数据的读入与写回
 - 分支判断或循环等指令跳转
 - 处理器需要支持以上所有功能

```
a = 0; b = a; res = 1;  
i = x;  
while ( --i != 0 )  
{ a=b;  
  b=res;  
  res=a+b;  
}  
f = res;
```

这个程序需要处理器支持哪些功能？

C语言 与 汇编语言

```
a = 0; // a = s1
b = a; // b = s2
res = 1;
i = x;
while ( --i != 0 ) // decrease i
                  // then check if 0
{ a=b;
  b=res;
  res=a+b;
}
```

```
f = res;
```

// note x itself isn't changed

x = 1 2 3 4 5 6 7 8 9 10

f = 1 1 2 3 5 8 13 21 34 55

前两项均为1，从第三项起，每一项都是其前两项的和

f = fib(x) (for $x > 0$) (Fibonacci)

.data

#x's address stored in \$a0

f's address stored in \$a1

.text

move \$s1,\$0 **#\$s1=0**

move \$s2,\$s1 **#\$s2=\$s1**

addiu \$s3,\$0,1 **#\$s3=1**

lw \$s4,0(\$a0) **#\$s4=x**

loop:

addi \$s4,\$s4,-1 **#\$s4=\$s4-1**

beqz \$s4,done **#if(\$s4=0)done**

move \$s1,\$s2 **#\$s1=\$s2**

move \$s2,\$s3 **#\$s2=\$s3**

addu \$s3,\$s2,\$s1 **#\$s3=\$s2+\$s1**

j loop **#goto loop**

done:

sw \$s3,0(\$a1) **#store result in f**

机器语言

- 计算机硬件的基本功能就是执行指令，指令在冯·诺伊曼计算机中由二进制数字进行编码
- 描述完成一个确定任务的指令序列称为**程序**
- 计算机的全部二进制机器指令组成了一种可供人与计算机进行交流的语言，称为**机器语言**

```
00100000000010000000000000000001  
001000010000100000000000000000010  
10101100000010000001111101000000
```

汇编语言

- 使用机器语言编写程序十分困难，于是人们发明了用助记符表示指令的方法。助记符形式的指令的集合组成了**汇编语言**
- 汇编语言的助记符形式的指令必须翻译成机器语言二进制指令才能在计算机上执行，实现这种翻译的程序称为汇编器(assembly)

```
addi $s1, $0, 1  
addi $s1, $s1, 2  
sw  $s1, 8000($0)
```

汇编语言源程序

```
00100000000010000000000000000001  
00100001000010000000000000000010  
10101100000010000001111101000000
```

机器语言源程序

高级语言

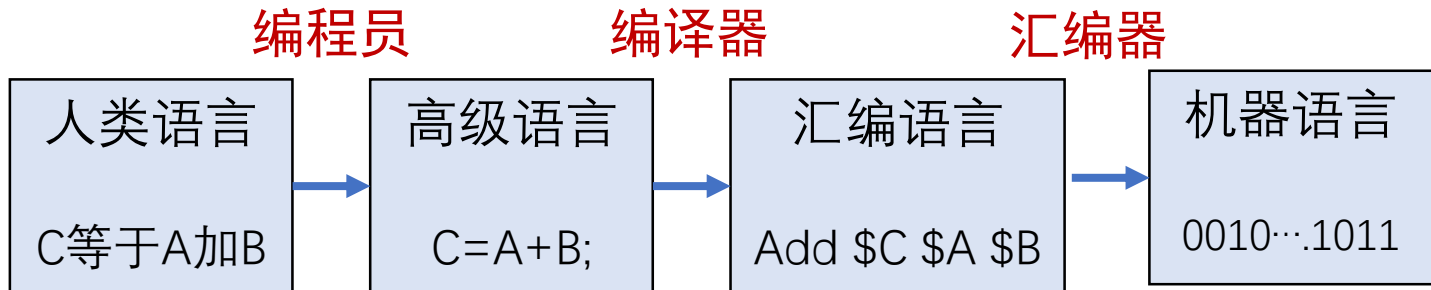
- 汇编语言与机器语言是一一对应的，所以开发效率仍然十分低下，于是人们发明了高级程序设计语言，如FORTRAN、C等等
- 使高级语言程序在只能运行二进制机器指令的计算机上运行，有两种途径：
 - **编译**——将高级语言编写的程序翻译成等价的二进制指令序列来代替，计算机执行等价的机器语言程序
 - **解释**——以高级语言程序作为输入数据，顺序地检查它的每一条语句，并直接执行等价的机器语言指令序列，这种方法称为解释

matlab

代码执行的流程

●从程序员到程序

- 程序员将人类语言描述的问题转化为用C, Java等高级语言描述的程序。
- 编译器将高级语言的程序变为目标平台的汇编语言 (MIPS, X86)
- 汇编器将汇编语言的程序翻译成机器语言的二进制码 (1010 0010 1011)



汇编语言 与 机器语言

C程序:
S1=1+2

```
addi $s1, $0, 1  
addi $s1, $s1, 2  
sw  $s1, 8000($0)
```

汇编语言源程序

汇编器

```
001000000000100100000000000000001  
0010000100101001000000000000000010  
10101100000010010001111101000000
```

机器语言程序

MIPS指令集

add \$8,\$9,\$10

000000	01001	01010	01000	00000	100000
--------	-------	-------	-------	-------	--------

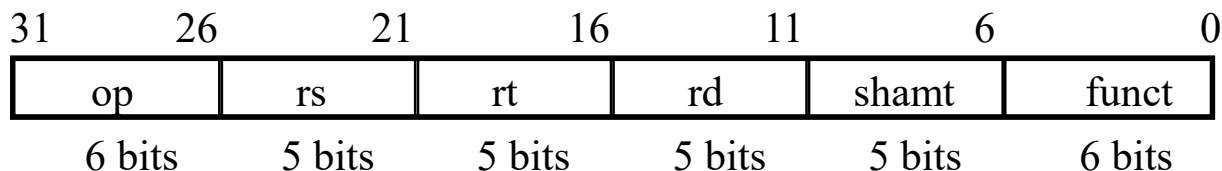
● ADD, SUB, AND, SLT

- addU rd, rs, rt
- subU rd, rs, rt
- and rd, rs, rt
- sltU rd, rs, rt

指令格式，寻址方式，指令

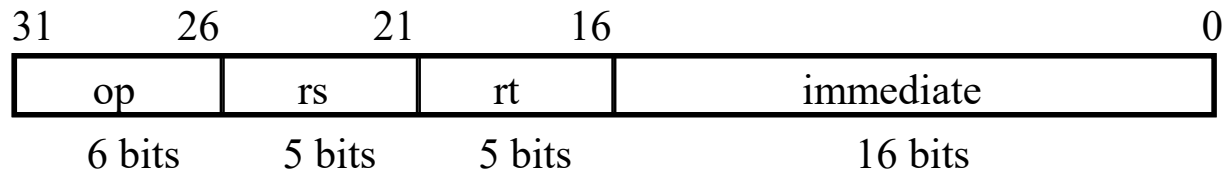
● OR Immediate:

- ori rt, rs, imm16



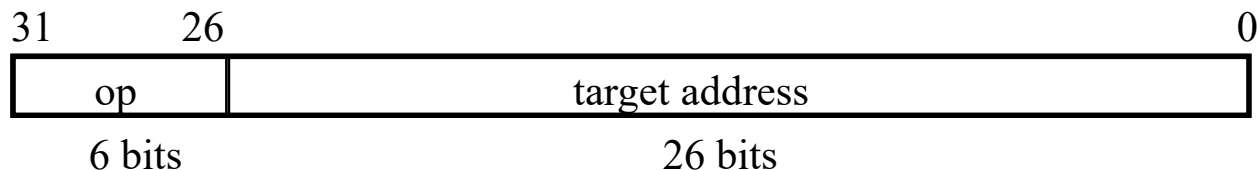
● LOAD and STORE Word

- lw rt, imm16(rs)
- sw rt, imm16(rs)

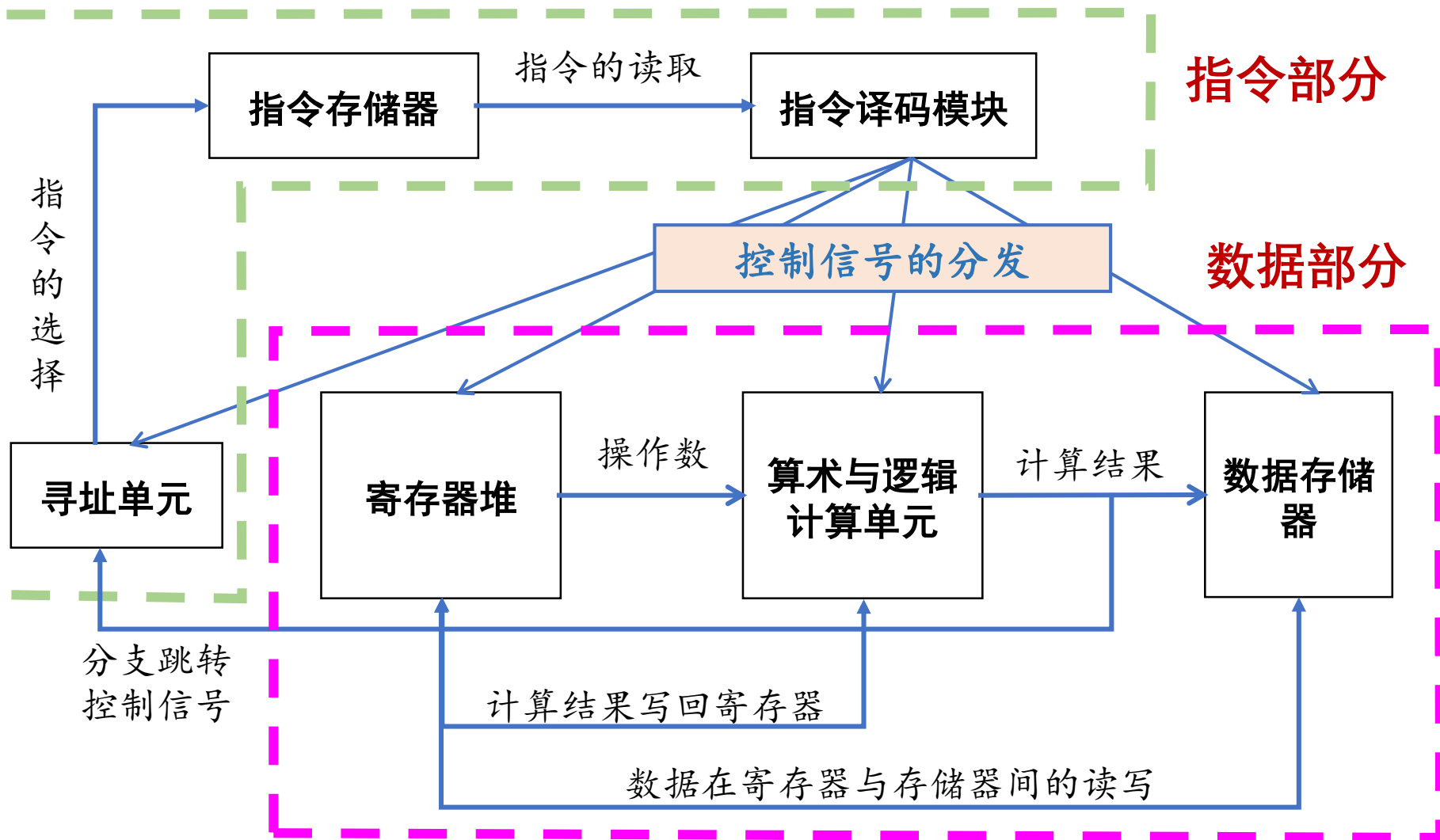


● BRANCH and JUMP:

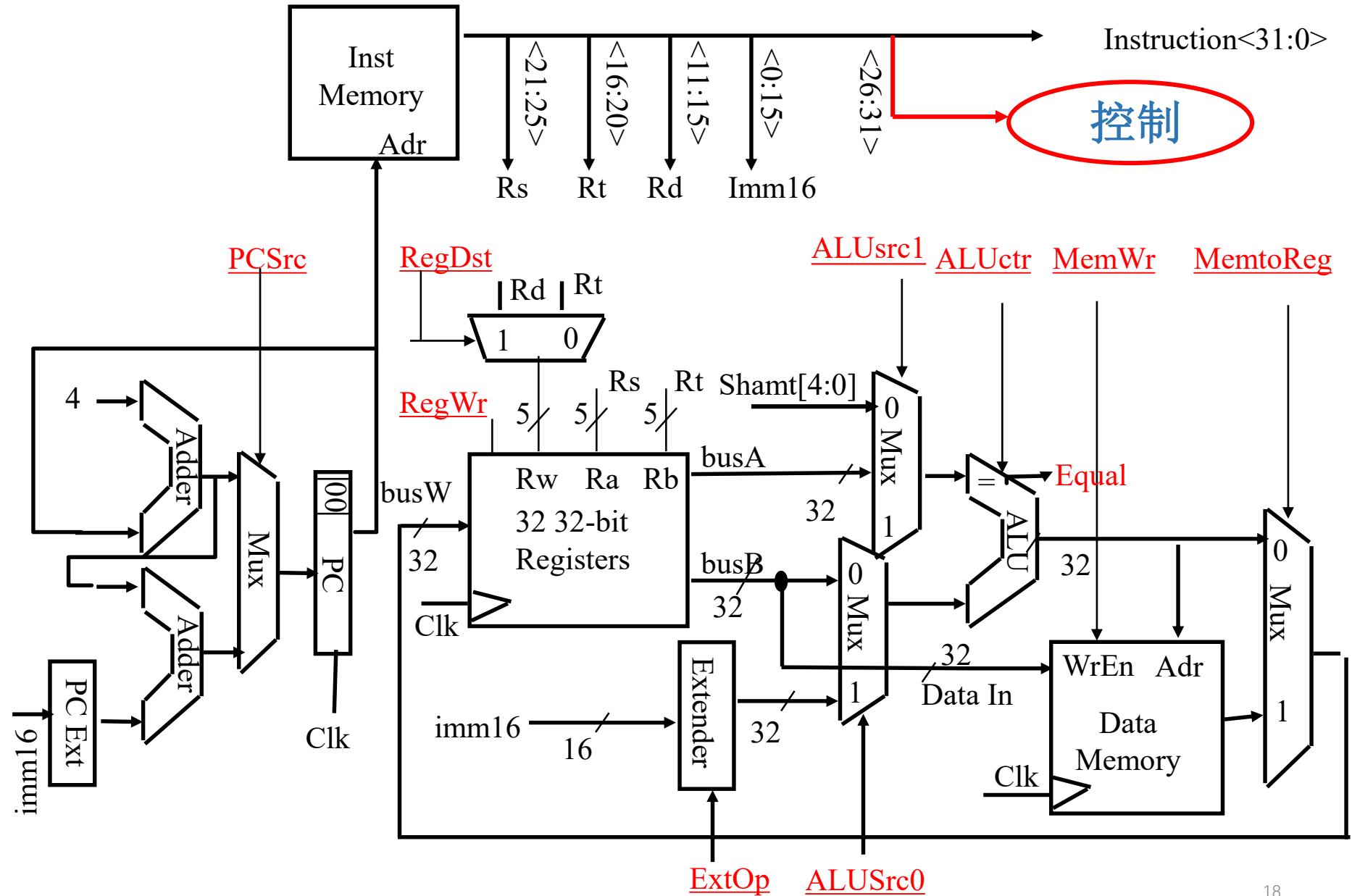
- beq rs, rt, imm16
- j Lable



处理器的架构：数据通路



CPU单周期数据通路

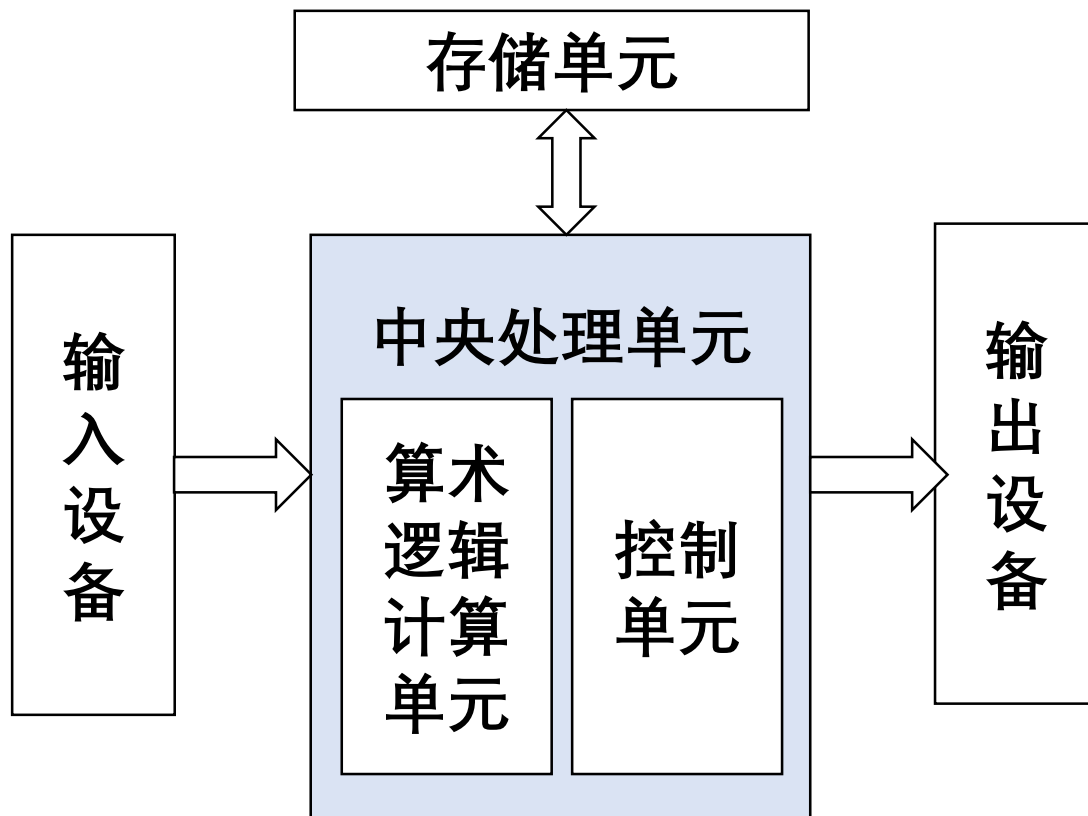


- **计算机组成**：指的是计算机体系结构的逻辑实现，包括物理机器级的数据流和控制流的组成以及逻辑设计
- **学习内容**：物理机器级指令的排序方式与控制方式、各部件的功能以及各部件之间的联系。
- **学习目标**：掌握计算机**五大部件**的组成原理、逻辑实现、设计方法及其互连构成整机系统的技术，以及整机的工作原理

计算机系统：计算机组成原理，计算机体系结构，微机原理与接口

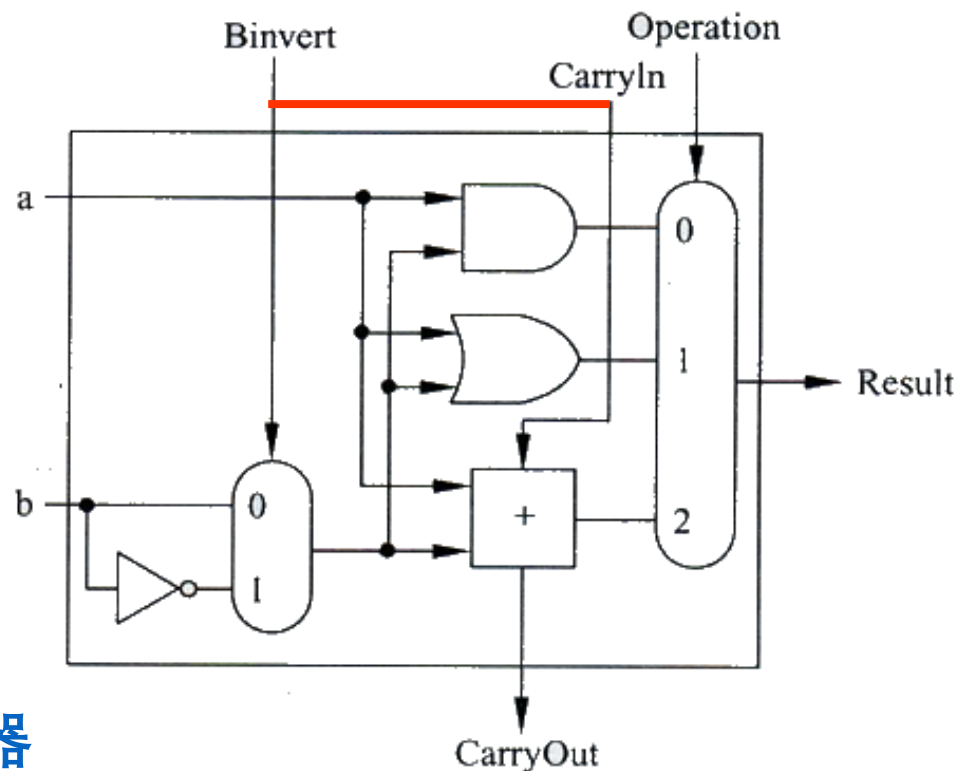
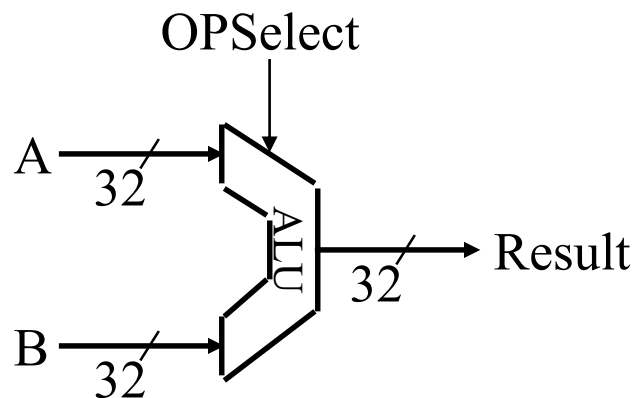
● 计算机五大部件的组成原理、逻辑实现、设计方法及系统连接

- ✓ 运算器
- ✓ 控制器
- ✓ 存储器
- ✓ 输入
- ✓ 输出

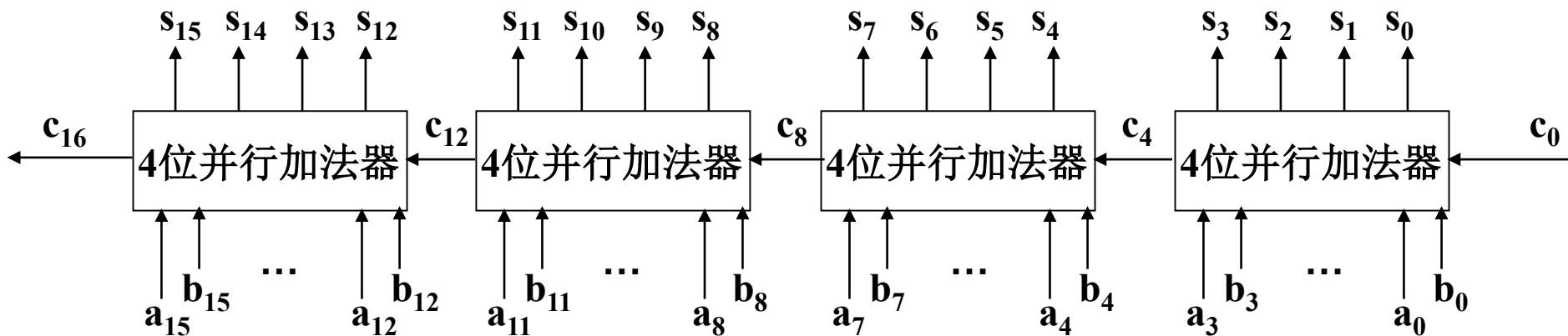


冯·诺依曼架构

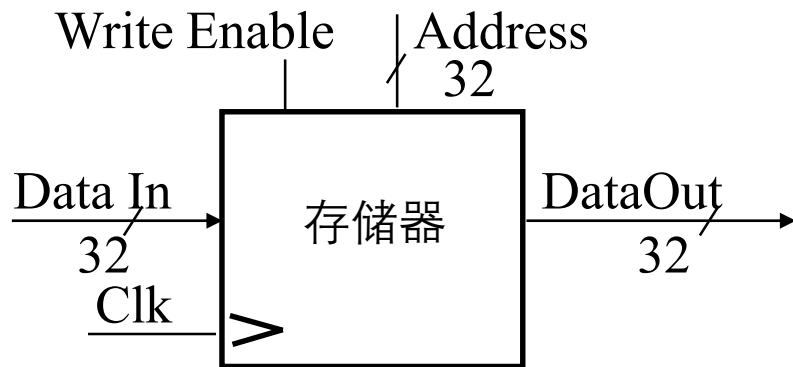
● 运算器



组内并行、组间串行进位加法器

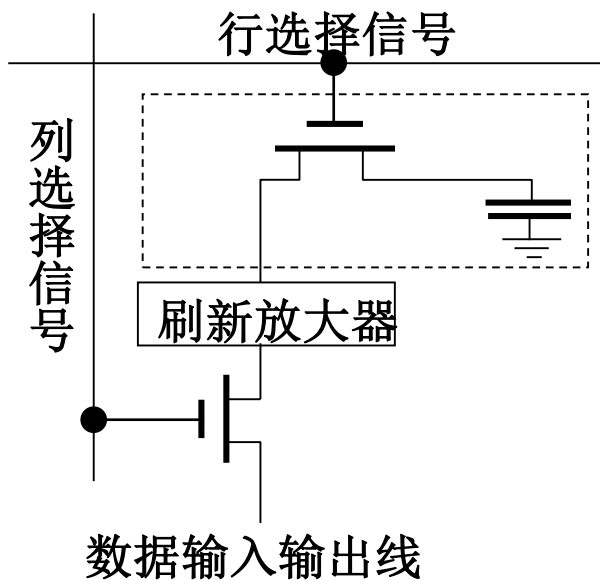


计算机系统：计算机组成原理，计算机体系结构，微机原理与接口

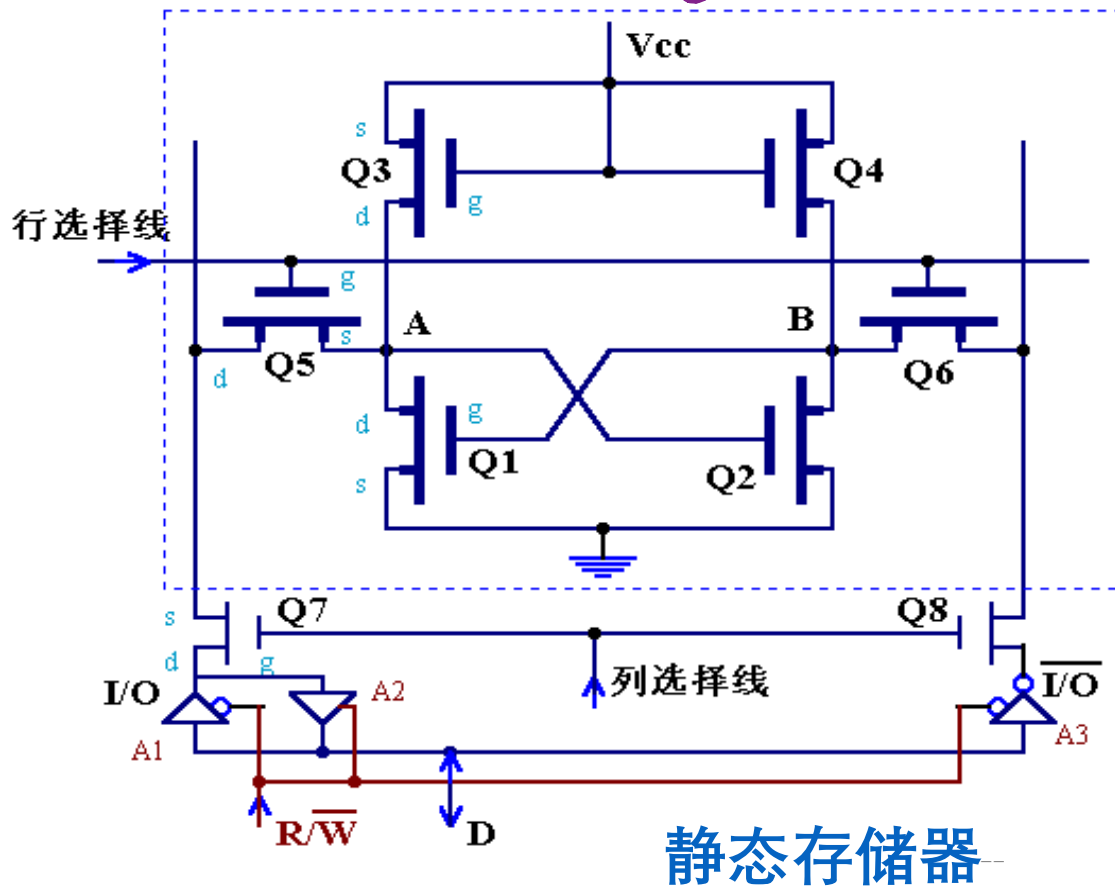


● 存储器

存储更快↓



动态存储器



● 存储器

\$sp → 7fff ffff hex

MIPS 程序和数据的存储器空间使用约定

- 从顶端开始，对栈指针初始化为 7ffffff, 并向下向数据段增长;
- 在底端，程序代码（文本）开始于 00400000;
- 静态数据开始于 10000000;
- 紧接着是由C中malloc进行存储器分配的动态数据，朝堆栈段向上增长

全局指针被设定为易于访问数据的地址，以便使用相对于\$gp的±16位偏移量

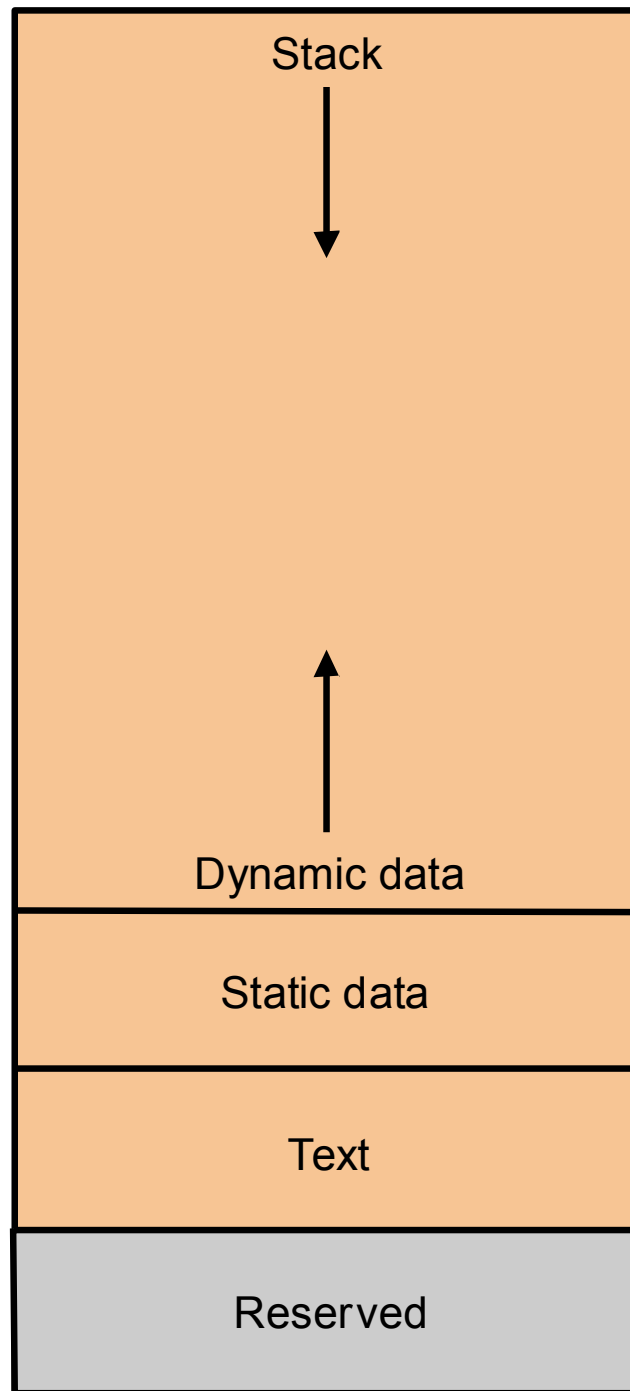
$10000000_{\text{hex}} - 1000ffff_{\text{hex}}$

\$gp → 1000 8000 hex

1000 0000 hex

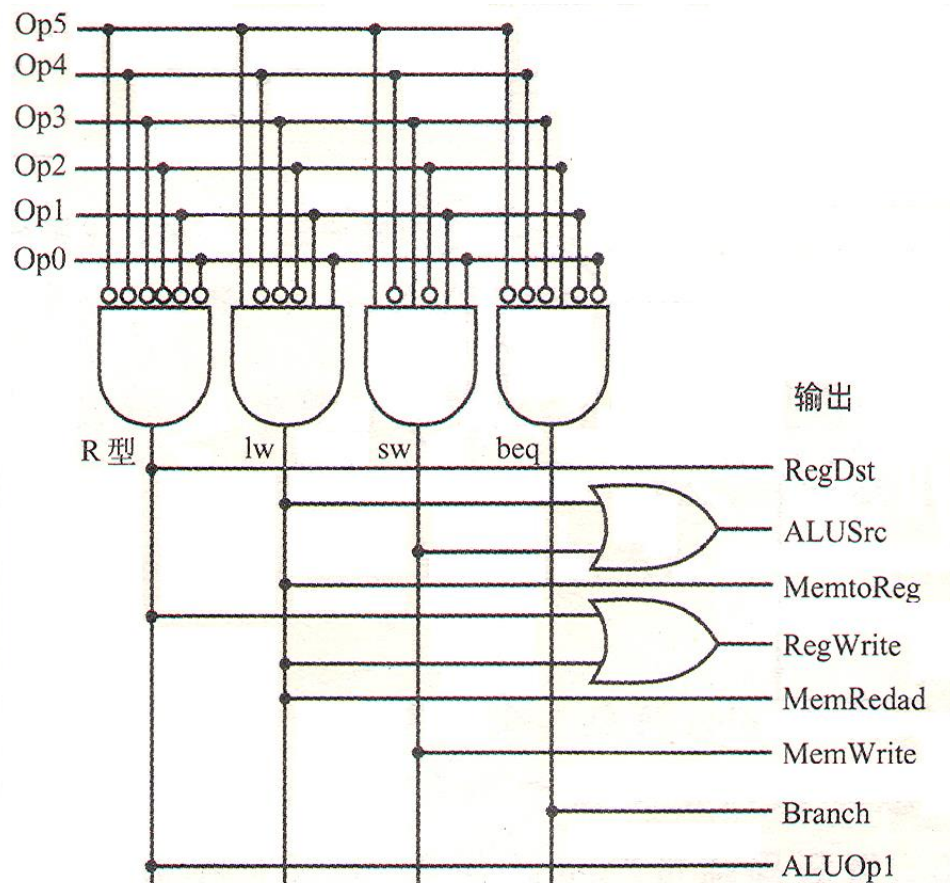
pc → 0040 0000 hex

0



● 控制器

控制	信号	R类型	lw	SW	beq
输入	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
输出	RegDest	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	Regwrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1



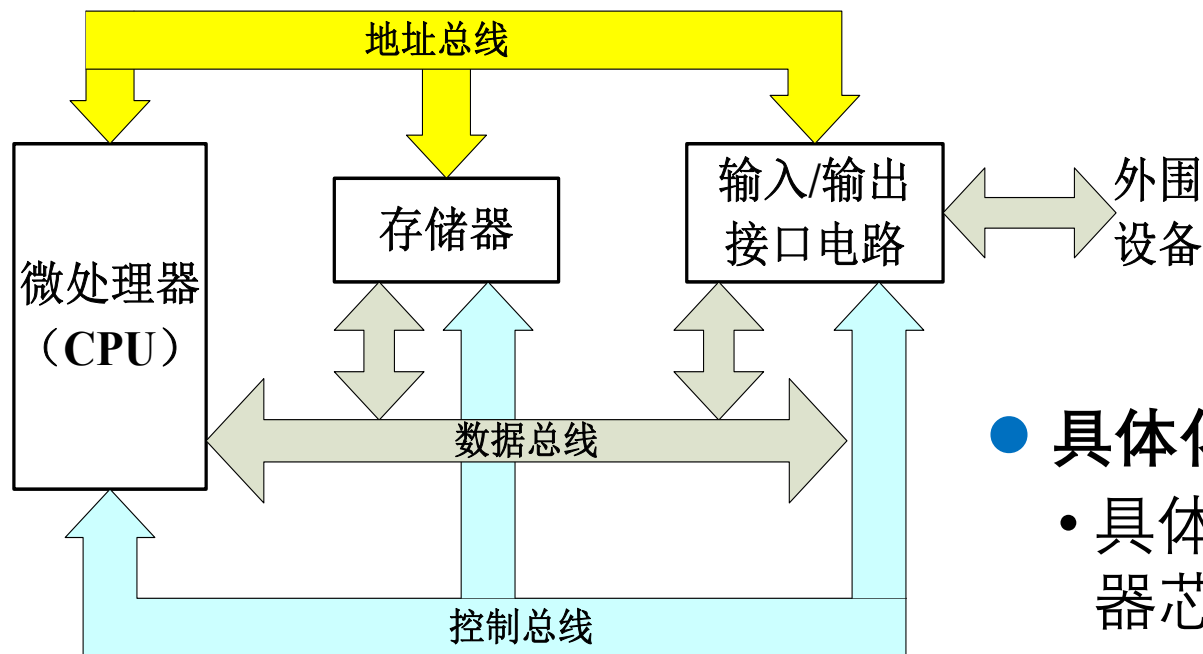
组合逻辑控制单元：用门电路实现主控功能

● 输入/输出

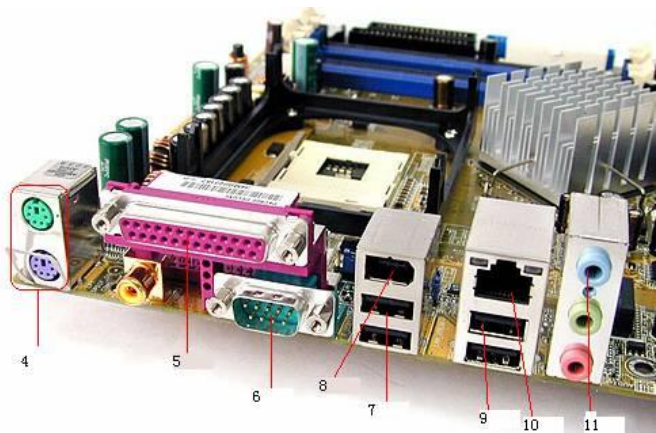


计算机 **I/O** 系统：接口、输入输出设备

计算机系统：计算机组成原理，计算机体系结构，微机原理与接口



微型机的基本结构



● 具体化计算机组成结构

- 具体的CPU芯片、存储器芯片、接口芯片

● 连接各组成部件成整机

- 芯片间信号配合，机器各部件正确理解

● 接近机器语言的编程语言让机器工作

- 汇编语言 (X86)

计算机系统：计算机组成原理，**计算机体系结构**，微机原理与接口

- 指令集结构
- 流水线技术
- 指令集并行
- 存储系统
- 输入输出系统

- ~~互联网络~~
- ~~多处理机~~
- ~~集群系统~~
- ~~多核系统结构~~

计算机体系结构：计算机的软、硬件接口



计算机组成：计算机体系结构的逻辑实现

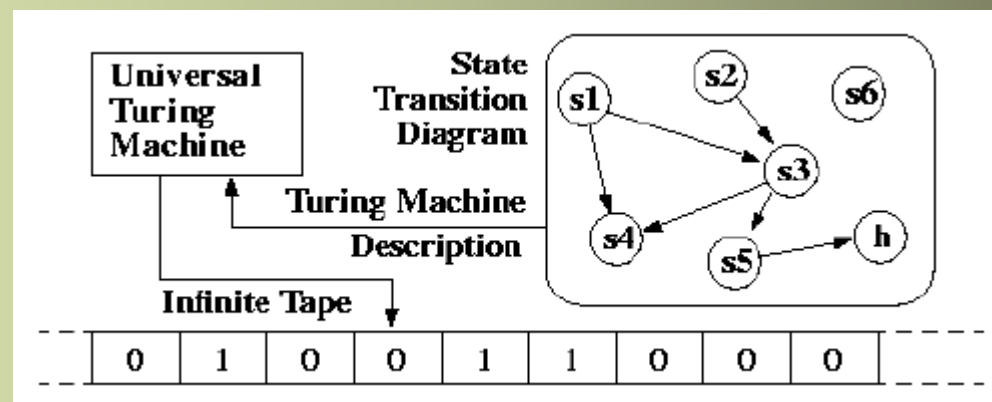


计算机实现：计算机组成的物理实现

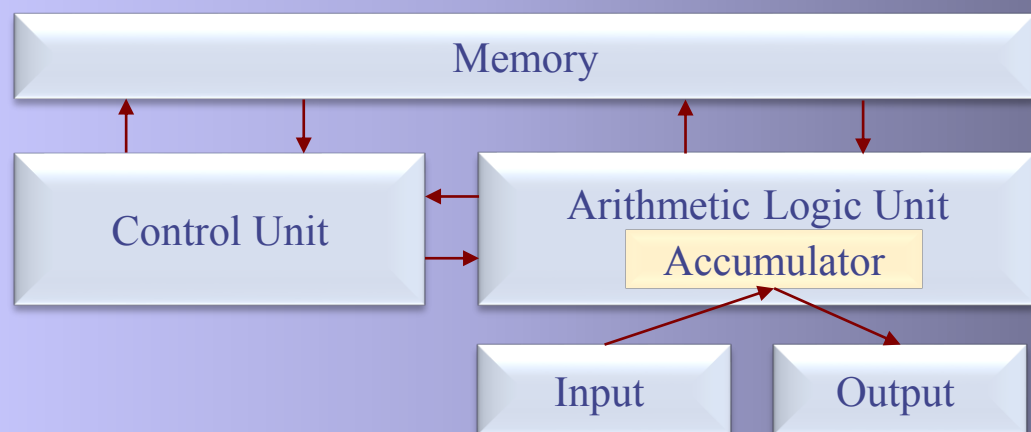
计算模型

图灵机

基于符号处理的
理论模型



电子计算机 冯.诺依曼结构



计算机体系结构教学日历

	内容		作业
第一周	1 基本概念	课程体系及要求	调研GPU的向量运算过程
		定量分析	CPU性能评价, 作业1.6-1.9
第二周	2 指令集	指令格式及CISC	学习使用Mars模拟器
		RICS及MIPS	熟悉MIPS指令与程序
第三周	3 流水线	基本概念	作业： 3.9, 3.11, 3.12, 3.14, 3.16 流水线冲突分析及转发设计 用Modelsim实现流水线CPU
性能指标			
第四周		相关与冲突	
		MIPS流水线实现	
第五周	4 指令级并行	动态调度	作业： 4.4, 4.5, 4.7, 4.9
第六周		分支预测	
		多指令流出	
第七周	5 存储系统	Cache层次体系	效率计算, 地址映像计算,
		Cache设计	作业： 5.8, 5.10, 5.12, 5.14
	6 输入输出	I/O与总线	总线设计, 6.6, 6.8, 6.10
第八周	7 实验课	计算机性能评价	
		流水线实现测试	

教材



第1章 计算机系统结构的基本概念

第2章 计算机指令集结构

第3章 流水线技术

第4章 指令级并行性

第5章 存储系统

第6章 输入输出系统

第7章 互连网络

第8章 多处理机

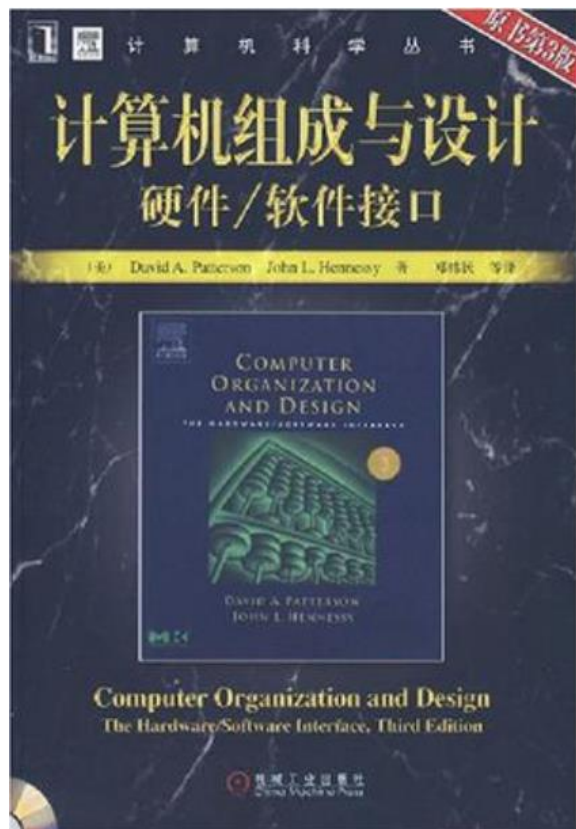
第9章 机群系统

第10章 多核系统结构与编程

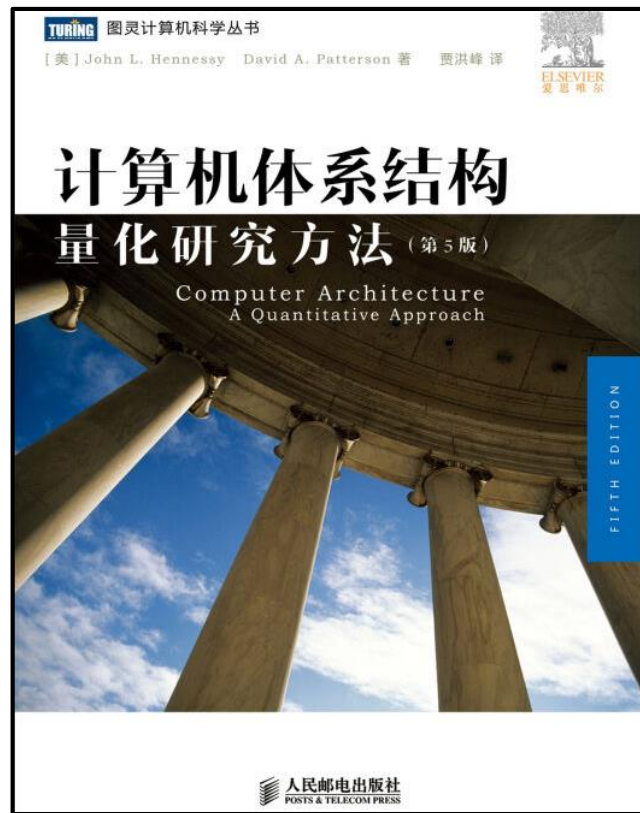
参考书

David A. Patterson and John L. Hennessy

- Computer Organization and Design: the Hardware/Software Interface 3 Ed.
- Computer Architecture: A Quantitative Approach 5 Ed.



计算机组成与设计—硬件/软件接口
郑伟民等译，机械工业出版社



计算机体系结构：量化研究方法
贾洪峰译，人民邮电出版社

参考书

- **Advanced Computer Architecture Parallelism Scalability Programmability**, Kai Hwang,
中文：高等计算机体系结构：并行性 可扩展性 可编程性，
清华大学出版社、广西科学技术出版社
- **Computer Organization and Architecture: Design for Performance**. William Stalling, Prentice_Hall International, Inc.
中文：计算机组织与结构：性能设计(第5版) 电子工业出版社
- **Computer Systems: A Programmer's Perspective (3 Ed.)**. Randal E. Bryant, David R. O'Hallaron. 龚奕利，贺莲译。
中文：深入理解计算机系统. 机械工业出版社，2016年

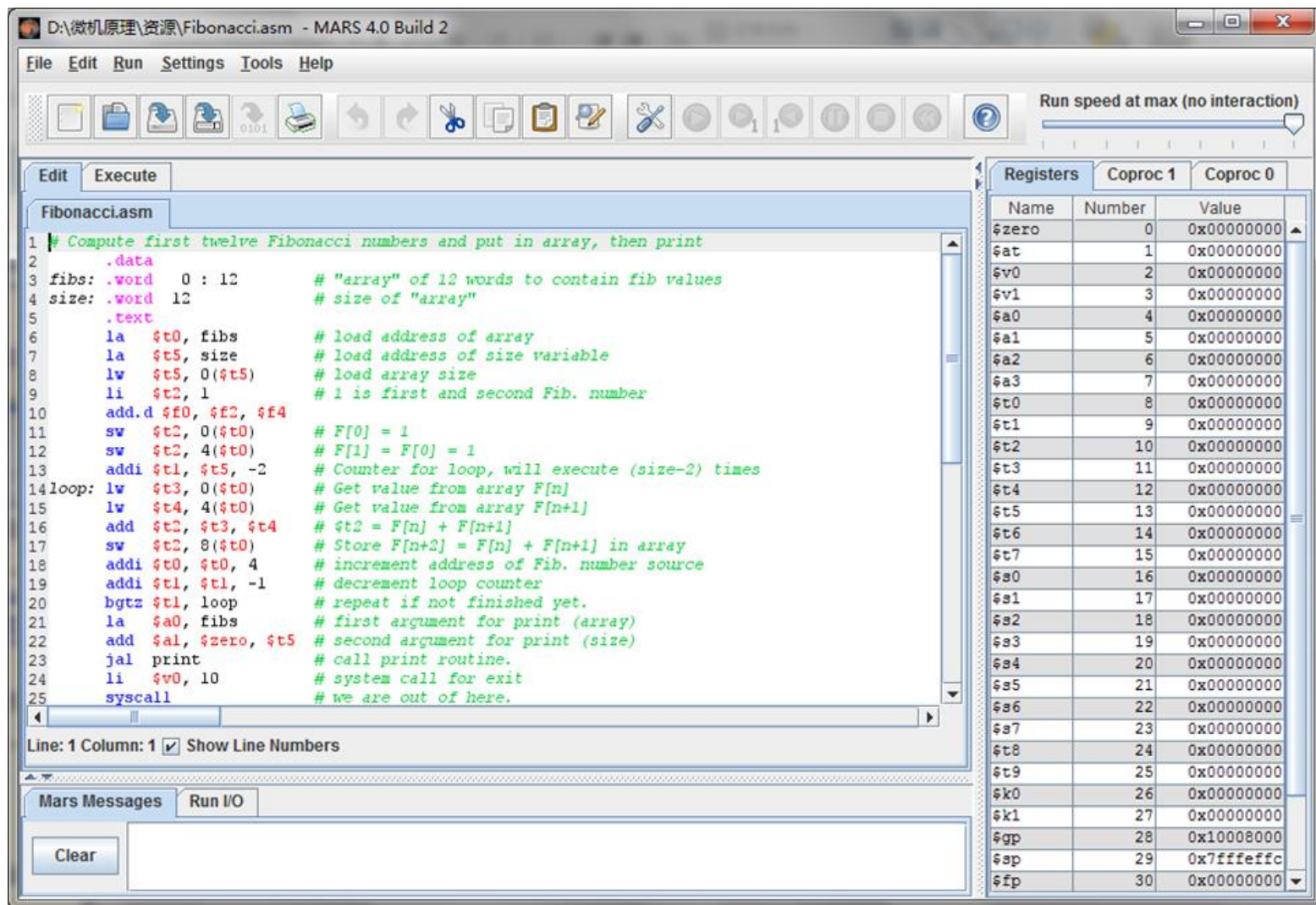
参考MIT课程 Computer Architecture

<https://6004.mit.edu/web/spring19/resources/lectures>

参考智慧树网上课程 <http://www.zhihuishu.com>，网页右上角点击
【登录】选择学号登录，输入学校、大学学号及初始密码123456。

- MARS 是MIPS Assembler and Runtime Simulator ， 能够运行和调试MIPS汇编语言程序，

<http://courses.missouristate.edu/KenVollmar/MARS/>



- SPIM是MIT教学版的MIPS模拟器，能够运行和调试MIPS汇编语言程序，支持Unix、Windows等多个操作系统平台，
<http://pages.cs.wisc.edu/~larus/spim.html>

The screenshot shows the PCSpim MIPS simulator interface. The title bar is 'PCSpim'. The menu bar includes 'File', 'Simulator', 'Window', and 'Help'. Below the menu is a toolbar with icons for file operations and simulation control. The main window is divided into several sections:

- Registers:** Displays the current state of MIPS registers.

PC = 0040000c	EPC = 00400000	Cause = 00000024	BadVAddr = 00000000
Status = 3000ff10	HI = 00000000	LO = 00000000	

 Below this, it lists General Registers R0 through R26 with their current values.
- Assembly Code:** A list of instructions with their addresses and comments.

[0x00400000]	0x3c101001	lui \$16, 4097 [fibs]	; 7: la \$s0, fibs
[0x00400004]	0x3c011001	lui \$1, 4097 [size]	; 8: la \$s5, size
[0x00400008]	0x3435004c	ori \$21, \$1, 76 [size]	
[0x0040000c]	0x8eb50000	lw \$21, 0(\$21)	; 9: lw \$s5, 0(\$s5)
[0x00400010]	0x34120001	ori \$18, \$0, 1	; 21: li \$s2, 1
[0x00400014]	0xae120000	sw \$18, 0(\$16)	; 22: sw \$s2, 0(\$s0)
- DATA:** A memory dump showing values at various addresses.

[0x10000000]...[0x1001004c]	0x00000000
[0x1001004c]	0x00000013
[0x10010050]	0x20776f48 0x796e616d 0x62694620 0x63616e6f
[0x10010060]	0x6e206963 0x65626d75 0x74207372 0x6567206f
[0x10010070]	0x6172655e 0x203f6574 0x3c203228 0x2078203d
- Status Bar:** At the bottom, it shows 'For Help, press F1' and the current register values: 'PC=0x0040000c EPC=0x00400000 Cause=0x00000024'.

作业：

1. 为什么说计算机体系结构的核心是指令集体系结构、计算机组成、硬件实现的总和？
2. 分析CPU单周期数据通路的组成和add、load两条指令的数据流。
3. 调研GPU的向量运算过程。

助教联系方式:

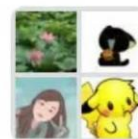
储华珍: 18811509969

chuhuazhen@163.com

梅若恒: 18811501228

blackcoldbird@qq.com

班微信群



计算机体系结构课程
2021春

