清華大學
Tsinghua University

第一单元　第二讲

计算机的指令系统

刘卫东

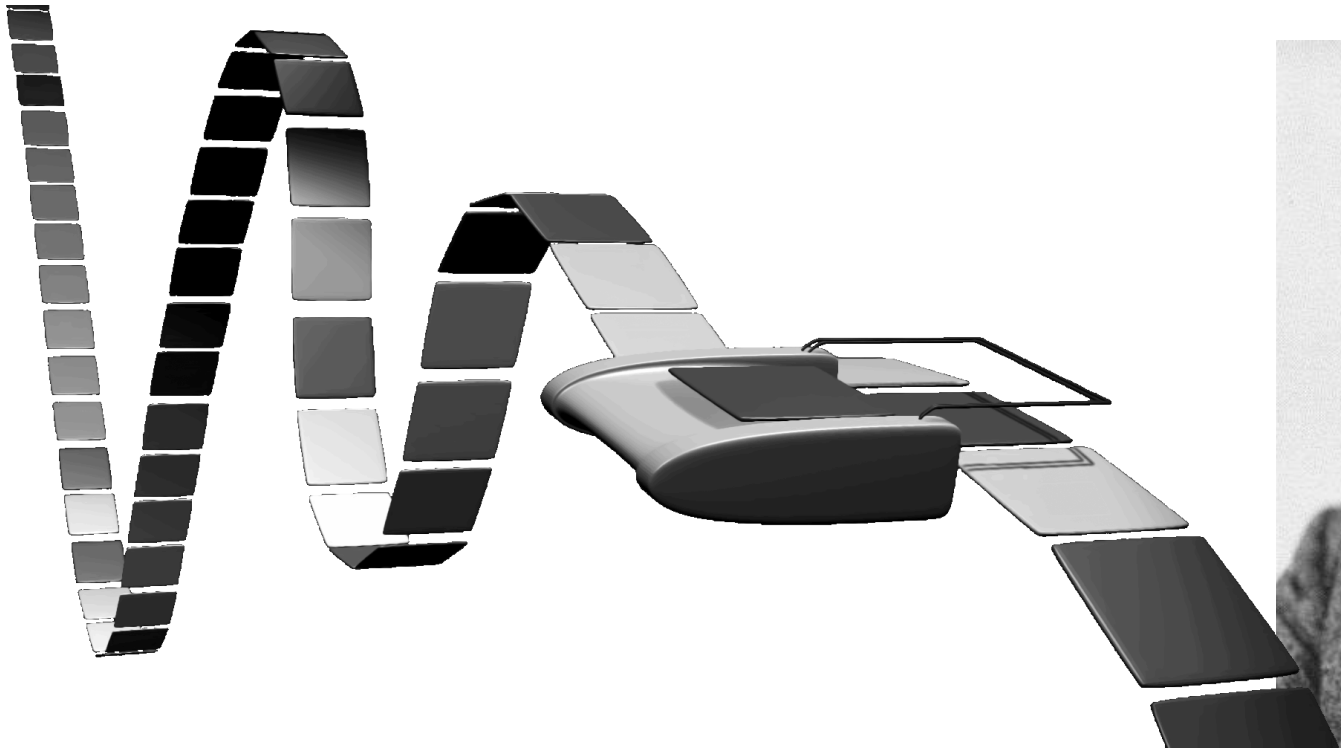
计算机科学与技术系

本讲概要

清華大學
Tsinghua University

- 计算机程序及分类
- 指令系统基本知识
- MIPS指令系统简介
- THCO MIPS指令系统
- THINPAD指令模拟器

图灵和图灵机



Turing Machine

1937

Alan Turing

计算机程序

- **Computer programs** (also **software programs**, or just **programs**) are instructions for a computer. A computer requires programs to function, and a computer program does nothing unless its instructions are executed by a central processor. Computer programs are either executable programs or the source code from which executable programs are derived (e.g., compiled).

- 程序员和计算机硬件之间交互的语言

- 计算机程序分类
  - 高级语言
  - 汇编语言
  - 机器语言

程序举例

C语言程序

```
main()
{ int fibo[10];
  int  i;
  fibo[0]=1; fibo[1] =1;
  for ( i =2 ; i<10; i++)
    fibo[i]= fibo[i-1]+
             fibo[i-2];
  }
```

Fibo

1
1
2
3
5
8
13
21
34
55

程序举例

清華大學 Tsinghua University

机器语言

6901: 0110100100000001

6A01: 0110101000000001

6B80

3360

6C09

DB20

DB41

E145

E149

4B02

程序举例

清華大學
Tsinghua University

汇编语言

```
LI    R1    1
LI    R2    1
LI    R3    80
SLL R3    R3    0
LI    R4    9
SW   R3    R1     0
SW   R3    R2     1
ADDU R1   R2   R1
ADDU R1   R2   R2
ADDIU R3   2
ADDIU R4   FF
BNEZ    R4    F9
```

程序举例

清華大學
Tsinghua University

汇编语言

```
LI    R1    1          ;将R1寄存器赋值为1
LI    R2    1          ;将R2寄存器赋值为1
LI    R3    80         ;将R3寄存器赋值为80h
SLL R3    R3    0      ;R3逻辑左移8位为8000h
LI    R4    9          ;将R4寄存器赋值为9，规定循环次数为9
SW  R3    R1    0      ;将R1的值写入[R3+0]内存处
SW  R3    R2    1      ;将R2的值写入[R3+1]内存处
ADDU R1  R2  R1    ;R1=R1+R2
ADDU R1  R2  R2    ;R2=R1+R2
ADDIU R3  2        ;R3=R3+2
ADDIU R4  FF       ;R4=R4-1
BNEZ   R4   F9     ;跳转到指令（SW R3 R1 0）处，F9为偏移量-7
```

高级语言

清華大學
Tsinghua University

高级语言又称算法语言，它的实现思路，不是过分地“靠拢”计算机硬件的指令系统，而是着重面向解决实际问题所用的算法，瞄准的是如何使程序设计人员能够方便地写出处理问题和解题过程的程序，力争使程序设计工作的效率更高。

用高级语言语言设计出来的程序，需要经过编译程序先翻译成机器语言程序，才能在计算机的硬件系统上予以执行，个别的选用解释执行方案。

高级语言的程序通用性强，在不同型号的计算机之间更容易移植。对高级语言进行编译、汇编后得到机器语言在计算机上运行。

汇编语言及机器语言

清華大學 Tsinghua University

汇编语言是对计算机机器语言进行符号化处理的结果,再增加一些为方便程序设计而实现的扩展功能。

在汇编语言中，可以用英文单词或其缩写替代二进制的指令代码，更容易记忆和理解；还可以选用英文单词来表示程序中的数据(常量、变量和语句标号)，使程序员不必亲自为这些数据分配存储单元，而是留给汇编程序去处理,达到基本可用标准。

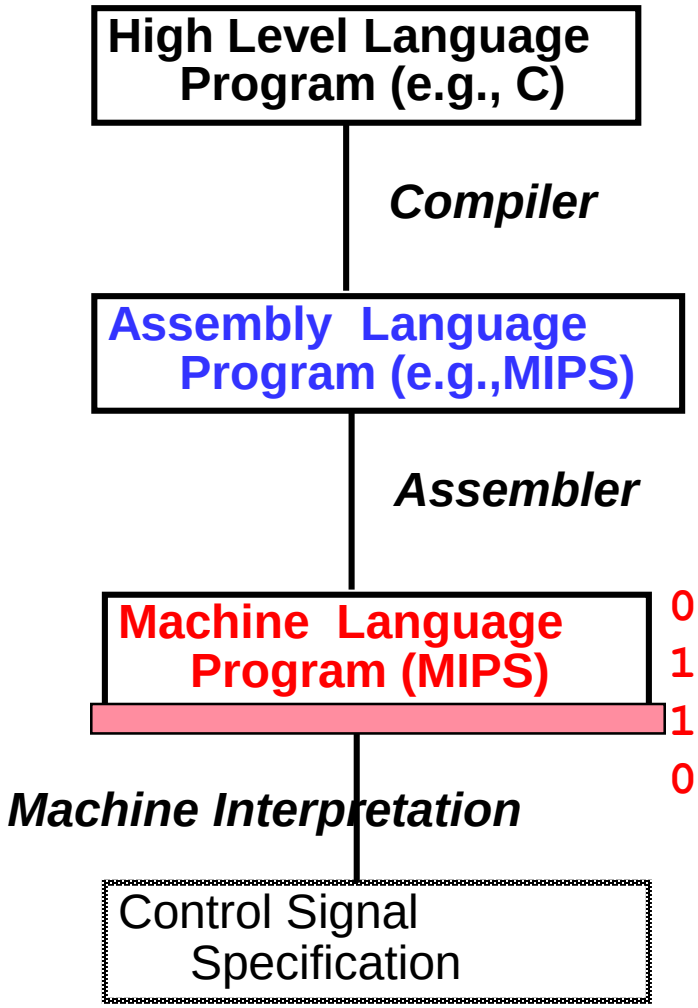
若在此基础上，能够在支持程序的不同结构特性（如循环和重复执行结构，子程序所用哑变元替换为真实参数）等方面提供必要的支持，使该汇编语言的实用程度更高。

汇编程序要经过汇编器翻译成机器语言后方可运行

机器语言是计算机硬件能直接识别和运行的指令的集合,是二进制码组成的指令，用机器语言设计程序基本不可行。

程序的最小单元是指令，同时，指令也是计算机硬件执行程序的最小单位。

计算机程序分类




```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw $t0,  0($2)
lw $t1,  4($2)
sw $t1,  0($2)
sw $t0,  4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

Von Neumann结构计算机

清華大學 Tsinghua University

- 存储程序计算机
  - 程序由指令构成
  - 程序功能通过指令序列描述
  - 指令序列在存储器中顺序存放
- 顺序执行指令
  - 用PC指示当前被执行的指令
  - 从存储器中读出指令执行
  - PC指向下一条指令

指令和指令系统

清華大學
Tsinghua University

- 计算机系统由硬件和软件两大部分组成。硬件指由中央处理器、存储器以及外围设备等组成的实际装置。软件是为了使用计算机而编写的各种系统的和用户的程序，程序由一个序列的计算机指令组成。
- 指令是计算机运行的最小的功能单元，是指挥计算机硬件运行的命令，是由多个二进制位组成的位串，是计算机硬件可以直接识别和执行的信息体。指令中应指明指令所完成的操作，并明确操作对象。
- 一台计算机提供的全部指令构成该计算机的指令系统。指令用于程序设计人员告知计算机执行一个最基本运算、处理功能，多条指令可以组成一个程序，完成一项预期的任务。

指令系统地位

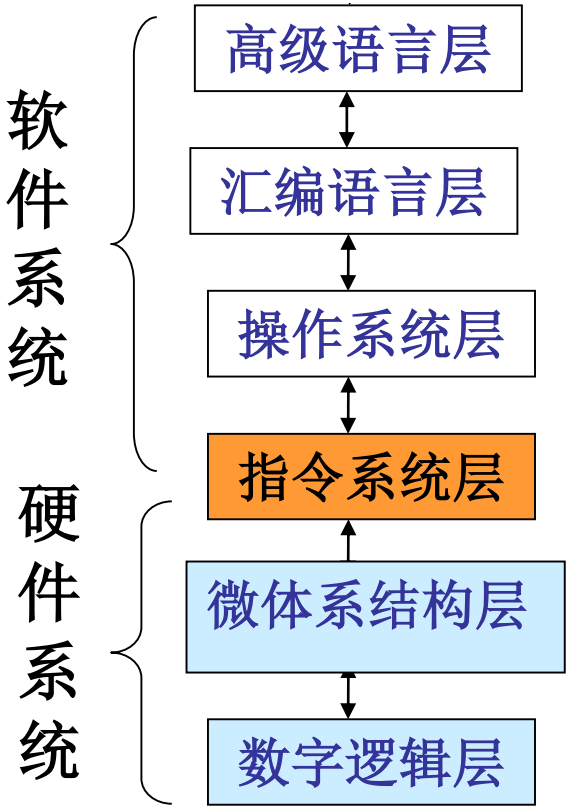
可以从6个层次分析和看待计算机系统的基本组成。

指令系统层处在硬件系统和软件系统之间，是硬、软件之间的接口部分，对两部分都有重要影响。

硬件系统用于实现每条指令的功能，解决指令之间的衔接关系；

软件由按一定规则组织起来的许多条指令组成，完成一定的数据运算或者事务处理功能。

指令系统优劣是一个计算机系统是否成功的关键因素。



计算机系统的层次结构

指令功能分类

- 数据运算指令
  - 算术运算、逻辑运算
- 数据传输指令
  - 寄存器之间、主存/寄存器之间
- 输入/输出指令
  - 与输入/输出端口的数据传输
- 控制指令
  - 转移指令、子程序调用/返回
- 其它指令
  - 停机、开/关中断、空操作、特权、置条件码

指令格式

- **指令格式**：指令字中**操作码**和**操作数地址**的二进制位的分配方案

操作码：指明本条指令的操作功能，
每条指令有一个确定的操作码

操作数地址：说明操作数存放的地址，有时是操作数本身

- **指令字**：完整的一条指令的二进制表示
- **指令字长**：指令字中二进制代码的位数

机器字长：计算机能直接处理的二进制数据的位数

指令字长（字节倍数）＝0.5、1、2…个机器字长

定长指令字结构　变长指令字结构

定长操作码　　扩展操作码

寻址方式

清華大學 Tsinghua University

寻址方式（又称编址方式）指的是确定本条指令的操作数地址及下一条要执行的指令地址的方法。

不同的计算机系统，使用数目和功能不同的寻址方式，其实现的复杂程度和运行性能各不相同。有的计算机寻址方式较少，而有些计算机采用多种寻址方式。

通常需要在指令中为每一个操作数专设一个地址字段，用来表示数据的来源或去向的地址。在指令中给出的操作数（或指令）的地址被称为形式地址，使用形式地址信息并按一定规则计算出来或读操作得到的一个数值才是数据（或指令）的实际地址。在指令的操作数地址字段，可能要指出：

① 运算器中的累加器的编号或专用寄存器名称（编号）

② 输入/输出指令中用到的 I/O 设备的入出端口地址

③ 内存储器的一个存储单元（或一 I/O设备 ）的地址

评价计算机性能的指标

清華大學 Tsinghua University

- 吞吐率
  - 单位时间完成的任务数量
- 响应时间
  - 完成任务的时间
- 衡量性能的指标
  - MIPS
  - CPI
  - CPU time
  - CPU Clock
- 综合测试程序（测试床）

MIPS指令系统

- MIPS：
  - Microprocessor without interlocked piped stages
  - 无内部互锁流水级的微处理器
  - RISC芯片
  - 由John L. Hennessy设计
- MIPS：
  - Million Instructions Per Second
  - 计算机性能指标之一

MIPS处理器

清華大學
Tsinghua University

Year	Model - Clock Rate (MHz)	Instruction Set	Cache (I+D)	Transistor Count
1987	R2000-16	MIPS I	External: 4K+4K to 32K+32K	115 thousand
1990	R3000-33		External: 4K+4K to 64K+64K	120 thousand
1991	R4000-100	MIPS III	8K+8K	1.35 million
1993	R4400-150		16K+16K	2.3 million
	R4600-100		16K+16K	1.9 million
1995	Vr4300-133		16K+8K	1.7 million
1996	R5000-200	MIPS IV	32K + 32K	3.9 million
	R10000-200		32K + 32K	6.8 million
1999	R12000-300		32K + 32K	7.2 million
2002	R14000-600		32K + 32K	7.2 million

MIPS指令格式

清華大學 Tsinghua University

寄存器型

op	rs	rt	rd	shamt	funct

例如： add $1, $2, $3　　$3←$1+$2

立即数型

op	rs	rt	address / immediate

例如： lw $1, $2, 100　　$2←M[$1+100]

addi $1, $2, 100　　$2←$1+100

转移型

op	target

例如： j 8000　　转移到 PC[31..28] | 8000×4

所有的指令都是32位字长。有 3种指令格式，即寄存器型、立即数型和转移型。

操作数寻址方式有基址加16位位移量的访存寻址、立即数寻址及寄存器寻址 3种。

MIPS指令系统（续）

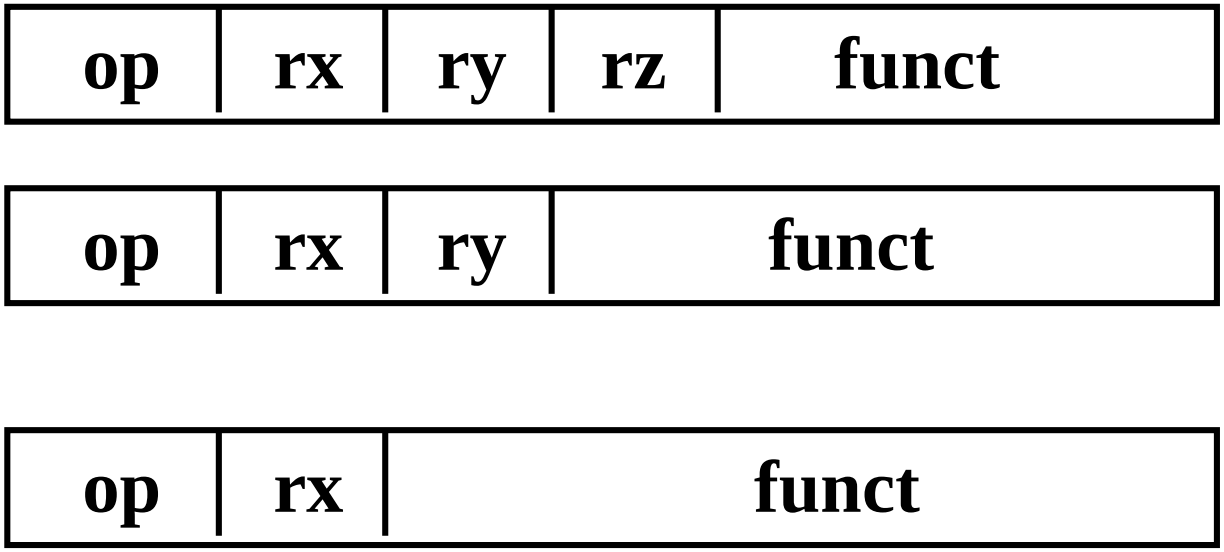
- MIPS 64
  - 面向64位处理器的指令系统
- MIPS 16e
  - 16位字长的MIPS指令集
  - 主要用于嵌入式系统
  - 本课程实验借用了其指令格式，在16位教学机上实现

THCO MIPS指令系统

- 采用与MIPS 16e兼容的指令格式
  - 16位固定字长
  - 操作码位置及长度固定
  - 寻址方式简单
- 共设计有44条指令
  - 可根据需要进行扩展
- 作为本课程教学实验的指令系统

THCO MIPS指令格式

- R型（21条）：



SLL、SRL、SRA、SLLV、SRLV、SRAV、MTSP、MOVE、ADDU、SUBU、MFPC、SLT、SLTU、CMP、NEG、AND、OR、XOR、NOT、MFIH、MTIH

THCO MIPS指令格式

- I型（14条）：

SW_RS、SW_SP、SW、LW_SP、LW、ADDIU3、ADDSP3、ADDSP、ADDIU、LI、SLTI、SLTUI、CMPI、INT

THCO MIPS指令格式

清華大學 Tsinghua University

- B型（5条）：

B、BEQZ、BNEZ、BTEQZ、BTNEZ

THCO MIPS指令格式

- J型（4条）：

JR、JRRA、JALR、NOP

THCO MIPS指令系统

清華大學 Tsinghua University

指令格式	汇编语句	操作数个数	指令类型	功能说明
00110 rx ry imm 00	SLL rx, ry, imm	3	R型指令	rx←ry<<immediate(L)
00110 rx ry imm 10	SRL rx ry imm	3		rx←ry>>immediate(L)
00110 rx ry imm 11	SRA rx ry imm	3		rx←ry>>immediate(A)
11101 rx ry 001 00	SLLV rx ry	2		ry ← ry<<rx
11101 rx ry 001 10	SRLV rx ry	2		ry ← ry>>rx(L)
11101 rx ry 001 11	SRAV rx ry	2		ry ← ry>>rx(A)
01100 100 rx 000 00	MTSP rx	1		SP ← rx
01111 rx ry 000 00	MOVE rx ry	2		rx ← ry
11100 rx ry rz 01	ADDU rx ry rz	3		rz ← rx + ry
11100 rx ry rz 11	SUBU rx ry rz	3		rz ← rx - ry
11101 rx 010 000 00	MFPC rx	1		rx ← PC
11101 rx ry 000 10	SLT rx ry	2		rx < ry时， T ← 1
11101 rx ry 000 11	SLTU rx ry	2		rx < ry时T ← 1(无符号)
11101 rx ry 010 10	CMP rx ry	2		rx = ry时， T ← 0
11101 rx ry 010 11	NEG rx ry	2		rx ← 0 - ry
11101 rx ry 011 00	AND rx ry	2		rx ← rx & ry
11101 rx ry 011 01	OR rx ry	2		rx ← rx || ry

THCO MIPS指令系统

清華大學 Tsinghua University

指令格式	汇编语句	操作数个数	指令类型	功能说明
11101 rx ry 011 10	XOR rx ry	2	R	rx ← rx ⊕ ry
11101 rx ry 011 11	NOT rx ry	2	型	rx← ~ ry
11110 rx 000 000 00	MFIH rx	1	指	rx← IH
11110 rx 000 000 01	MTIH rx	1	令	IH ← rx
01100 010 imm	SW-RS imm	1		MEM[SP + imm]←RA
11010 rx imm	SW-SP rx imm	2		MEM[SP + imm]←rx
11011 rx ry imm	SW rx ry imm	3		MEM[rx + imm]←ry
10010 rx imm	LW_SP rx imm	1		rx ←MEM[SP + imm]
10011 rx ry imm	LW rx ry imm	3	I	ry ← MEM[rx + imm]
01000 rx ry 0 imm	ADDIU3 rx ry imm	3	型	ry ← rx + sign_ext(imm)
00000 rx imm	ADDSP3 rx imm	2	指	rx ←SP + sign_ext(imm)
01100 011 imm	ADDSP imm	1	令	SP ←SP+ sign_ext(imm)
01001 rx imm	ADDIU rx imm	2		rx ← rx + sign_ext(imm)
01101 rx imm	LI rx imm	2		rx ← + sign_ext(imm)
01010 rx imm	SLTI rx imm	2		rx < s_ext(imm)时 T ← 1
01011 rx imm	SLTUI rx imm	2		rx < z_ext(imm)时 T← 1
01110 rx imm	CMPI rx imm	2		rx = imm 时 T ←0

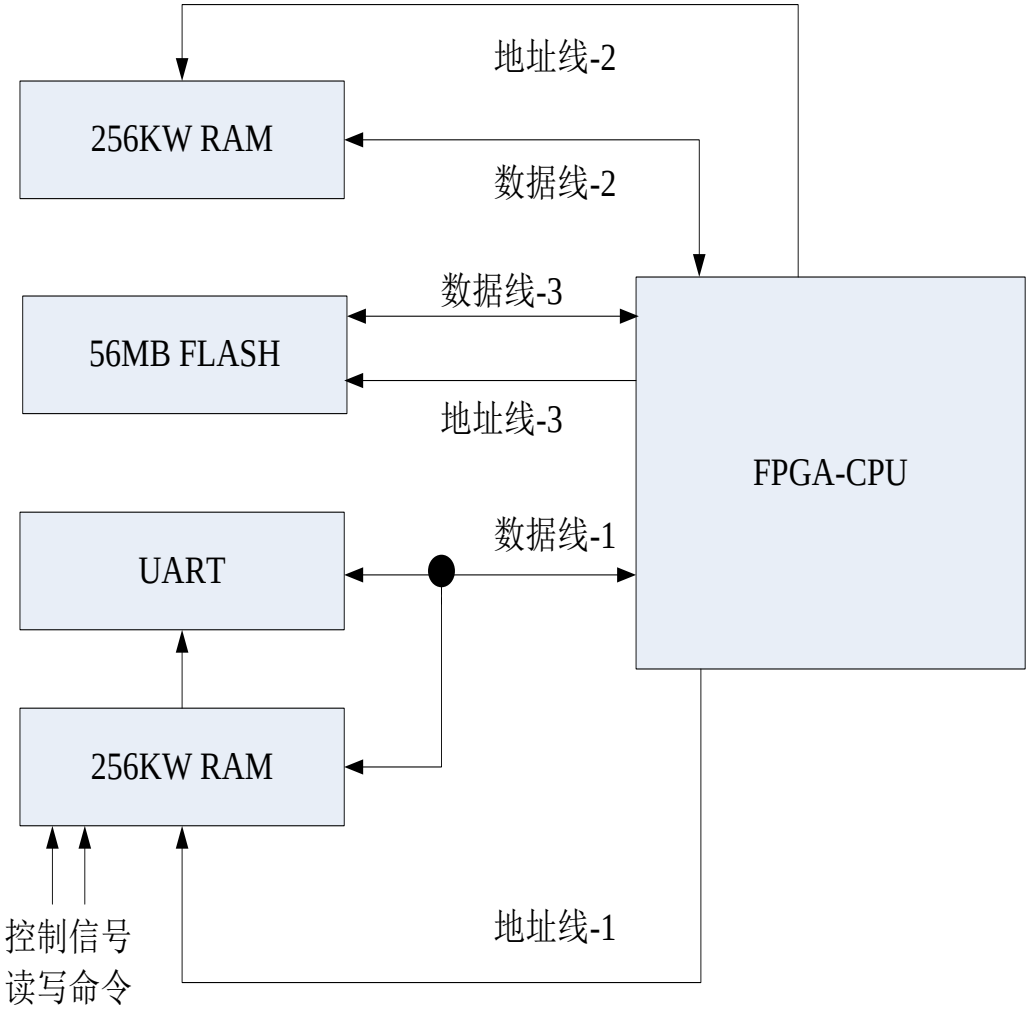
THCO MIPS指令系统

清華大學 Tsinghua University

指令格式	汇编语句	操作数个数	指令类型	功能说明
00010 imm	B imm	1	B型指令	PC←PC+ sign_ext(imm)
00100 rx imm	BEQZ rx imm	2		rx=0 时 跳转
00101 rx imm	BNEZ rx imm	2		rx !=0 时跳转
01100 000 imm	BTEQZ imm	1		T =0 时跳转
01100 001 imm	BTNEZ imm	1		T !=0时 跳转
11101 rx 00000000	JR rx	1	J型指令	PC←rx
11101 000 00100000	JRRA	0		PC ←RA
11101 rx 11000000	JALR rx	1		PC←rx RA ←PC
11111 00000 imm	INT imm	1		软中断
00001 000 00000000	NOP	0		空指令

THINPAD的硬件组成

- 机器字长16位
- CPU
  - 8个通用寄存器
  - 16位字长
  - PC、SP等专用寄存器
- 主存
  - 256KW RAM+ 256KW RAM
  - 按字编址
- I/O
  - 与主存共享地址空间，双串口（BF00/BF01、BF02/BF03）
- 总线
  - 双16位地址总线，双16位数据总线
  - 独立的访问FLASH总线



指令系统实现

- 存储器及I/O
  - 未设置单独的I/O指令，统一地址空间
  - 总线连接（地址、数据）
- DataPath
  - ALU
    - add\sub\and\or\
  - PC
  - SP
  - Register files
  - T
  - RA
  - IH

功能区	地址段	说明
系统程序区	0x0000~0x3FFF	存放监控程序
用户程序区	0x4000~0x7FFF	存放用户程序
系统数据区	0x8000~0xBEFF	监控程序使用的数据区
Com1数据端口/命令端口	0xBF00~0xBF01	串口的端口
Com2数据端口/命令端口	0xBF02~0xBF03	第2个串口的端口
预留给其他接口	0xBF04~0xBF0F	保留
系统堆栈区	0xBF10~0xBFFF	用于系统堆栈
用户数据区	0xC000~0xFFFF	用户程序使用的数据区

THCO MIPS寻址方式

清華大學
Tsinghua University

- 操作数寻址方式
  - 寄存器寻址
  - 立即数寻址
  - 变址寻址

汇编语言程序设计

清華大學
Tsinghua University

- 模拟器介绍
  - 功能：对THCO MIPS指令系统进行指令级功能模拟
  - 实现：高级语言实现
  - 可完成汇编（ea）、反汇编（u）、查看/设置寄存器（r/sr）、单步/连续运行（s/c）、断点（b/db/lb）、重新启动（restart）

程序举例

- 求和： SUM=1+2+3+…+10

```
LI R1 1
LI R3 0
ADDU R1 R3 R3
ADDIU R1 1
SLTI R1 B
BTNEZ FC
NOP
```

程序举例

清華大學 Tsinghua University

Fibonacci数列

```
LI R1 1          ;将R1寄存器赋值为1
LI R2 1          ;将R2寄存器赋值为1
LI R3 80         ;将R3寄存器赋值为80h
SLL R3 R3 0      ;R3逻辑左移8位，变为8000h
LI R4 9          ;将R4寄存器赋值为9，规定循环次数为9
SW R3 R1 0       ;将R1的值写入[R3+0]内存处
SW R3 R2 1       ;将R2的值写入[R3+1]内存处
ADDU R1 R2 R1    ;R1=R1+R2
ADDU R1 R2 R2    ;R2=R1+R2
ADDIU R3 2       ;R3=R3+2
ADDIU R4 FF      ;R4=R4-1
BNEZ R4 F9       ;跳转到指令（SW R3 R1 0）处
```

小结

- 计算机程序语言
  - 高级语言
  - 汇编语言
  - 机器语言
- 指令系统
  - 硬件/软件接口
  - 指令功能/指令格式
- THCO MIPS指令系统

阅读和思考

- 阅读
  - 教材：第2章 指令系统
  - 实验指导书(指令系统、模拟器)
- 思考
  - 指令系统的作用和地位?
  - 为实现THCO指令系统,ALU应该具备哪些功能?
  - 数据在计算机内如何表示?应表示哪几类数据?
- 练习
  - Project1（实验指导书5.1）
  - 分析THCO MIPS指令系统的特点
  - 读监控程序源码(结合Term程序)（实验指导书5.2）