

微机原理与接口技术

MIPS指令操作数

华中科技大学 左冬红



MIPS指令操作数——寄存器操作数

• 整数操作类指令

寄存器为所有算术、逻辑运算类指令的主要操作数来源

编号	名称	使用规则或特殊用途
\$0	\$zero	常数0(constant value 0)
\$1	\$at	汇编程序用来保存超出16位数表示范围的数据
\$2~\$3	\$v0~\$v1	函数调用返回值(values for results and expression evaluation)
\$4~\$7	\$a0~\$a3	函数调用参数(arguments)
\$8~\$15	\$t0~\$t7	临时寄存器(temporary)
\$16~\$23	\$s0~\$s7	存储寄存器(saved), C语言中定义的变量可以保存在这些寄存器中。同时这些寄存器也可以保存存储单元的起始地址(基地址)
\$24~\$25	\$t8~\$t9	临时寄存器(temporary)
\$28	\$gp	全局指针(global pointer)
\$29	\$sp	栈指针(stack pointer)
\$30	\$fp	帧指针(frame pointer)
\$31	\$ra	返回地址(return address)

寄存器操作数示例

寄存器\$*s*0、\$*s*1、\$*s*2、
\$*s*3、\$*s*4分别存放变
量*a*、*b*、*c*、*d*、*e*的值

```
add t0,c,b  
add t1,d,e  
sub a,t0,t1
```



```
add $t0,$s2,$s1  
add $t1,$s3,$s4  
sub $s0,$t0,$t1
```

MIPS指令操作数——存储器操作数

- 一般情况下，基本类型数据都采用寄存器存储
- 存储器存储复杂类型数据或寄存器不够情况下的数据
- MIPS数据存储要求边界对齐
 - 半字数据存储地址需为偶数
 - 字数据存储地址需为4的整数倍

int数据存储地址为0x12?

不行！！

MIPS指令操作数——存储器操作数

如何访问复杂类型数据?

`a[i]`

`a.data`

`a->data`

首地址加偏移地址形式构成复杂类型数据中各元素的地址

学习目标：若已知首地址、数据中各元素的类型以及排列关系，能获得任意元素的地址表达式

数组在存储器中的存储方式如何?

依照索引顺序存储

存储器操作数地址构成示例

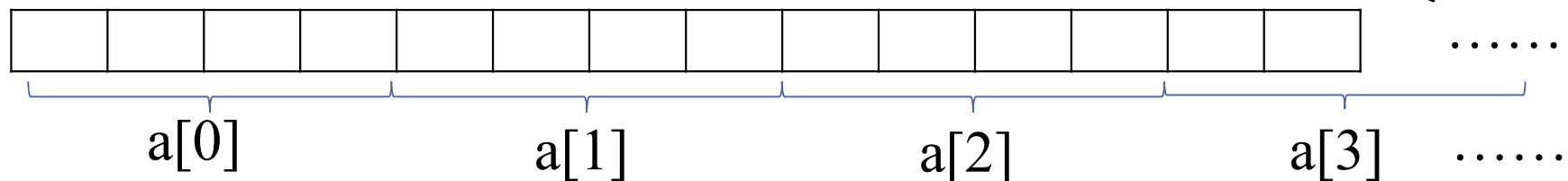
`int a[12];`

a表示什么含义?

每个元素占多少个存储空间?

a[n]的地址如何表示? $n=0\sim11$

低地址a



a[i]的地址与i之间满足什么关系?

$\&a[i] = a + 4 \times i$

存储器操作数

a 存储在 \$s1 中

$\&a[2] = a + 4 \times 2$



$8(\$s1)$

注意书写格式

a 存储在 \$s1, i 存储在 \$a0

$\&a[i] = a + 4 \times i$



$4 \times \$a0 + (\$s1)$

不存在此寻址方式！！

错误！！

需由多条指令实现地址计算

存储器操作数示例

1) $g = h + A[8];$

变量g和h分别保存在寄存器\$*s*0和\$*s*1中

2) $A[12] = h + A[8];$

A保存在寄存器\$*s*2中

A[8]

32(\$*s*2)

A[12]

48(\$*s*2)

$g = h + A[8]$

add \$*s*0,\$*s*1,32(\$*s*2)

错误!!

运算类指令不能直接操作存储器操作数，需首先将存储器操作数装载入寄存器之后才能运算，运算结果也不能直接写入存储器操作数，需运算之后，再存储到存储器操作数

存储器操作数示例

1) $g = h + A[8];$

变量g和h分别保存在寄存器\$*s0*和\$*s1*中

2) $A[12] = h + A[8];$

A保存在寄存器\$*s2*中

A[8]

32(\$*s2*)

装载字: lw

A[12]

48(\$*s2*)

存储字: sw

$g = h + A[8]$

lw \$*t0*, 32(\$*s2*)
add \$*s0*, \$*s1*, \$*t0*

$A[12] = h + A[8]$

lw \$*t0*, 32(\$*s2*)
add \$*t0*, \$*s1*, \$*t0*
sw \$*t0*, 48(\$*s2*)

MIPS指令操作数——立即数

立即数即出现在指令中的数据

如lw \$t0,32(\$s2)中的32

addi \$s1,\$s2,40中的40

立即数可支持两种数制：十进制、十六进制

十六进制采用0x作为前缀表示

小结

- 操作数类型
 - 寄存器操作数
 - 运算类指令的操作数
 - 存储器操作数
 - 唯一表现形式 $\text{imm}(\$R)$, 仅出现在装载、存储指令中
 - 复杂数据类型地址表达式计算, 如 `short a[n]` 地址
- 立即数操作数
 - 与指令存储在一起

下一讲: MIPS指令编码