MIPS指令系统简介

主讲人：邓倩妮

上海交通大学

MIPS指令集设计思想和发展历程

- MIPS的全称
  - 无互锁流水段的微处理器
  - Microprocessor without Interlocked Piped Stages
- 主要关注点
  - 减少指令的类型
  - 降低指令复杂度
- 基本原则
  - 简单的 CPU 就是更快速的 CPU.

- 1981年， MIPS I
- MIPS II
- MIPS III
- MIPS IV
- MIPS V
- 嵌入式指令体系MIPS16
- 1999年， MIPS32、MIPS64

MIPS指令系统的特征

- 简单
  - 指令定长，都是32位。
  - 指令数量少、功能简单
  - 指令格式简单，只有三种：
    - 立即数型（I）、转移型（J）、寄存器型（R）
  - 操作数寻址方式少：
    - 基址加16位偏移量的访存寻址
    - 立即数寻址
    - 寄存器寻址

- 使用较多的寄存器

- 只有Load和Store指令可以访问存储器
  - 例如，不支持x86指令的这种操作：ADD AX,[3000H]

- 需要编译器支持
  - 无互锁流水段，依靠编译器进行指令序列的重新安排，减少流水线中出现的冲突

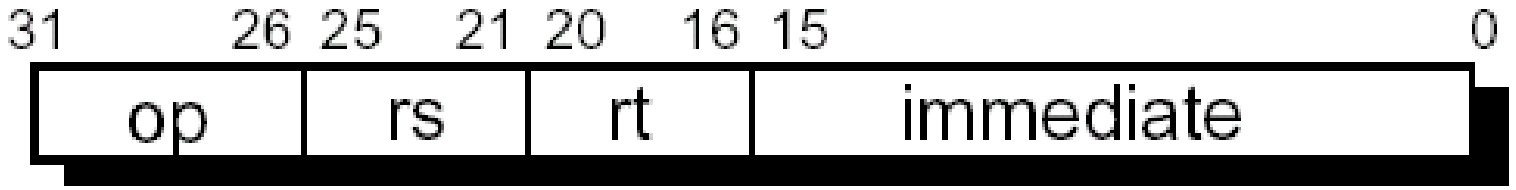
MIPS通用寄存器（32位宽）用途

编号	名称	用途	编号	名称	用途
0	$zero	常量0 （The Constant Value 0）	24-25	$t8-$t9	临时变量 Temporaries
1	$at	汇编器临时变量	26-27	$k0-$k1	为操作系统内核保留
2-3	$v0-$v1	函数返回结果、表达式值	28*	$gp	全局指针 Global Pointer
4-7	$a0-$a3	函数调用时传递的参数	29*	$sp	栈指针 Stack Pointer
8-15	$t0-$t7	临时变量 Temporaries	30*	$fp	结构指针/桢指针 Frame Pointer
16-23*	$s0-$s7	需要保存的临时变量	31*	$ra	返回地址 Return Address

* 在过程调用时需要保存

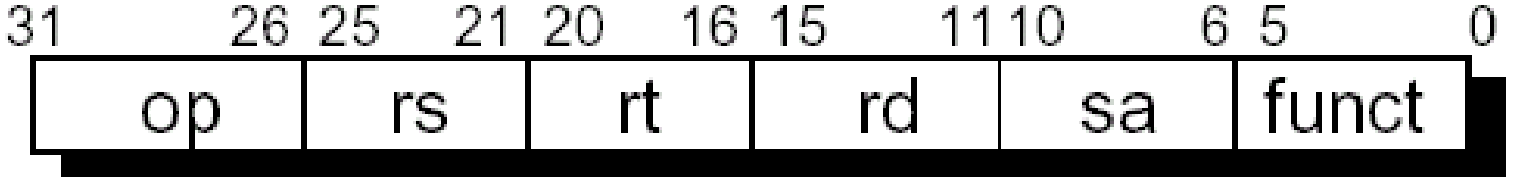
MIPS指令的三种格式

I-Type (Immediate)



J-Type (Jump)

R-Type (Register)

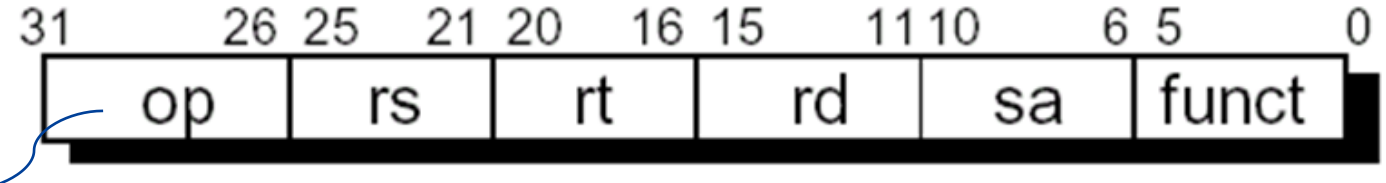


MIPS指令操作码定义

I28~I26 I31~I29	000	001	010	011	100	101	110	111
000	R 格式	bltz/gez	j	jal	beq	bne	blez	bgtz
001	addi	addiu	slti	sltiu	andi	ori	xori	lui
010	TLB 指令	浮点指令						
011								
100	lb	lh	lwl	lw	lbu	lhu	lwr	
101	sb	sh	swl	sw			swr	
110	lwc0	lwc1						
111	swc0	swc1						

MIPS R格式指令扩展操作码定义

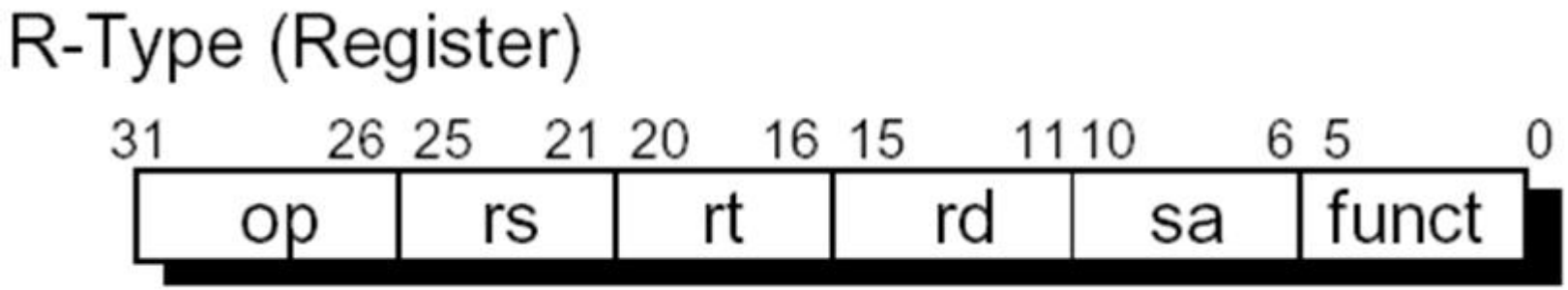
I5~I3 \ I2~I0	000	001	010	011	100	101	110	111
000	sll		srl	sra	sllv		srlv	srav
001	j $r	jalr			syscall	break		
010	mfhi	mthi	mflo	mtlo				
011	mult	multu	div	divu				
100	add	addu	sub	subu	and	or	xor	nor
101			slt	sltu				
110								
111								



R-型: 000000

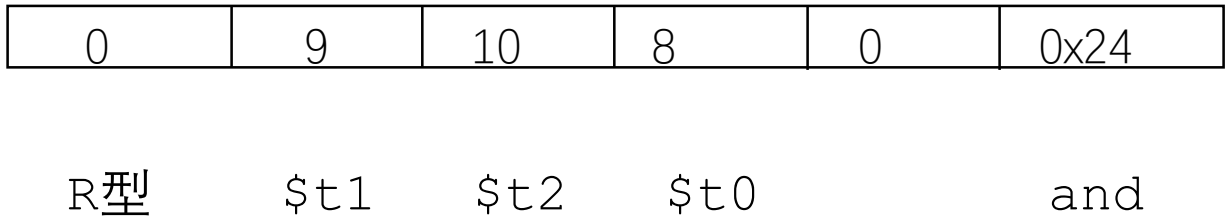
指令扩展操作码:
R型指令的最后6位

先看 R型



MIPS 寄存器指令 (R 型)

- 例如:
- `add $t0, $t1, $t2  #$t0 = $t1 + $t2`
- `sub  $t0, $t1, $t2 #$t0 = $t1 - $t2`
- `and $t0, $t1, $t2  #$t0 = $t1 & $t2`
-
- 指令格式 (R 型)

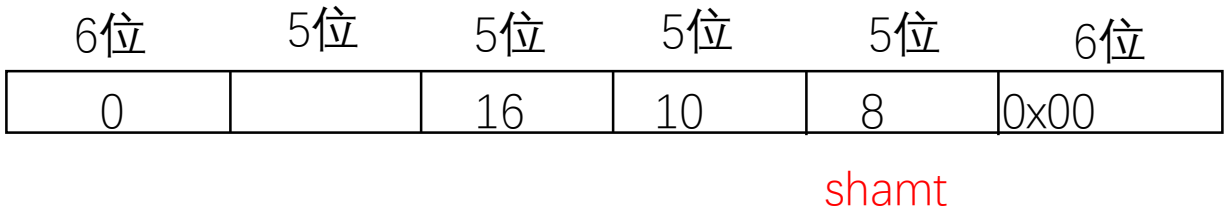


MIPS 移位操作(R型)

- 逻辑左移、逻辑右移
- `sll $t2, $s0, 8   #$t2 = $s0 << 8 bits`

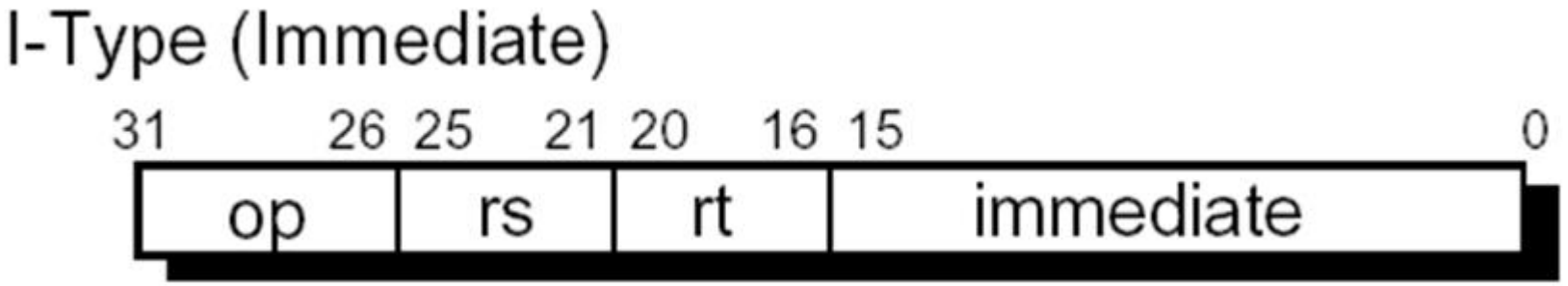
  `srl $t2, $s0, 8   #$t2 = $s0 >> 8 bits`
- 逻辑移位：移位后空出的位置填零

- 指令格式 (R 型)
- `sll $t2, $s0, 8`

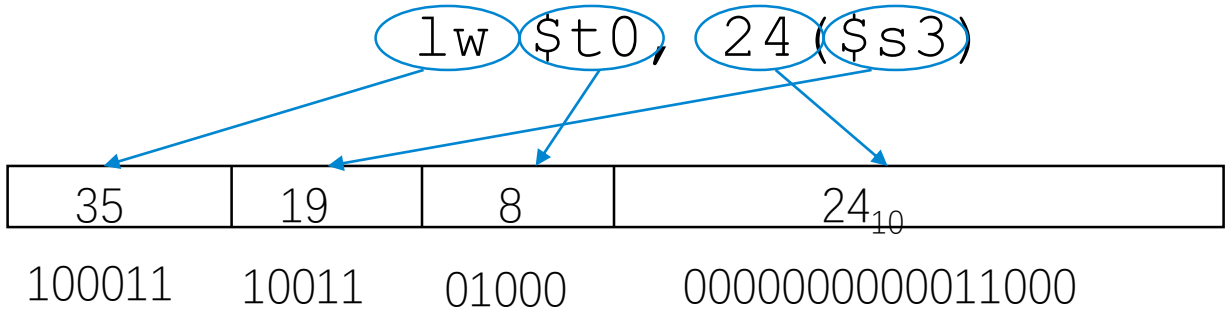


- 注意：5位宽的 shamt 域，足够将一个32位的二进制数值最多移位31（2^5-1）位

I型

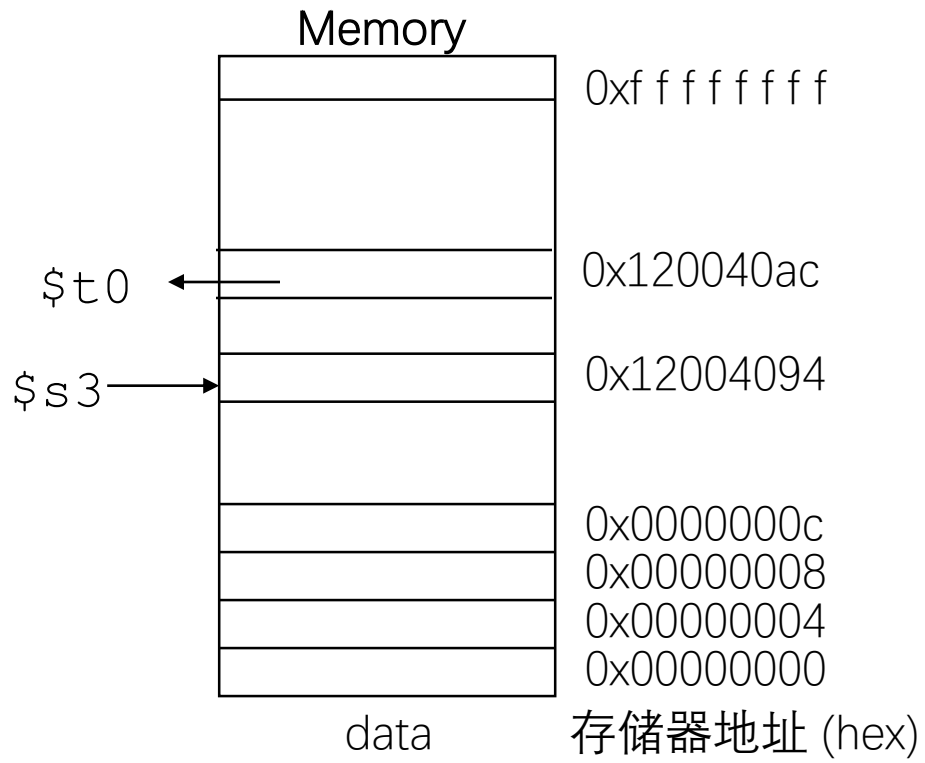


Load/Store 指令格式 (I 型):



$s3 + 24_{10} = 0x12004094+ 0x18
= 0x120040ac

. . . 0001 1000 (**0x18**)
+ . . . 1001 0100 (**0x94**)
. . . 1010 1100 (**0xac**)



MIPS 数据传递指令 （I型）

两种数据传送指令:

- `lw $t0, 4($s3) #load word from memory`
- **`sw $t0, 8($s3) #store word to memory`**

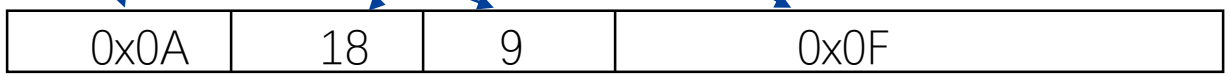
- 数据在存储器和寄存器文件堆中的一个寄存器（5 bit address）之间传送

- 要访问的存储地址– 一个32 位地址 加一个16位偏移量
- 16位偏移量
  - 将访问范围限制在基地址偏移量为 $\pm 2^{15}$ 或 32,768 bytes 的范围

MIPS 立即数指令 (I 型):

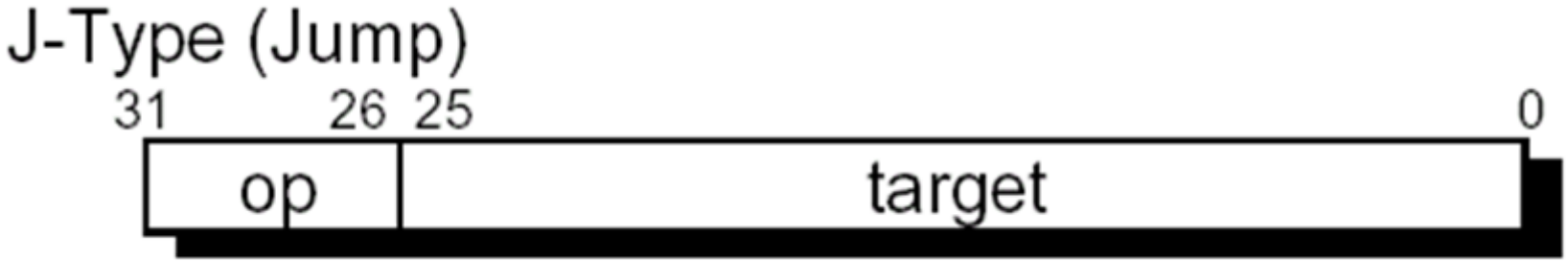
- 典型代码中常常会用到一些值不算大的常数
- 直接在指令中包含这些常数，速度比从存储器或寄存器中获得它们要快

```
andi $t0, $t1, 0xFF00      #$t0 = $t1 & ff00
addi  $sp, $sp, 4          #$sp = $sp + 4
slti $t1, $s2, 15          #$t1 = 1 if $s2<15
```



立即数的范围： $+2^{15}-1$ to -2^{15}

J型



小结

- MIPS的特点
  - 简单、指令条数少、操作数寻址方式简单
- MIPS的指令格式
- 了解了以下类型的指令
  - R型、I型
- 下一节再介绍其他指令