本节主题

MIPS体系结构

北京大学·慕课

计算机组成

制作人：陆俊林

MIPS的设计者 和 RISC的先驱

约翰·亨尼西
John Hennessy
1953年出生

- 1977年，进入斯坦福大学任职
- 1981年，领导RISC微处理器的研究小组
- 1984年，共同创立MIPS计算机系统公司
- 1989年~1999年，先后担任斯坦福大学计算机系统实验室主任、计算机系主任和工程学院院长等
- 2000年起，任斯坦福大学校长

RISC：Reduced Instruction Set Computer，精简指令系统计算机

CISC：Complex Instruction Set Computer，复杂指令系统计算机

IEEE Medal of Honor “for pioneering the RISC processor architecture and for leadership in computer engineering and higher education"

MIPS公司的商业兴衰

- 1984年，MIPS计算机系统公司成立
- 1988年，SGI公司在其计算机产品中采用MIPS处理器
- 1989年，MIPS第一次上市
- 1992年，SGI收购MIPS，更名为MIPS技术公司
- 1998年，MIPS再次上市
- 2012年，Imagination Technologies收购MIPS

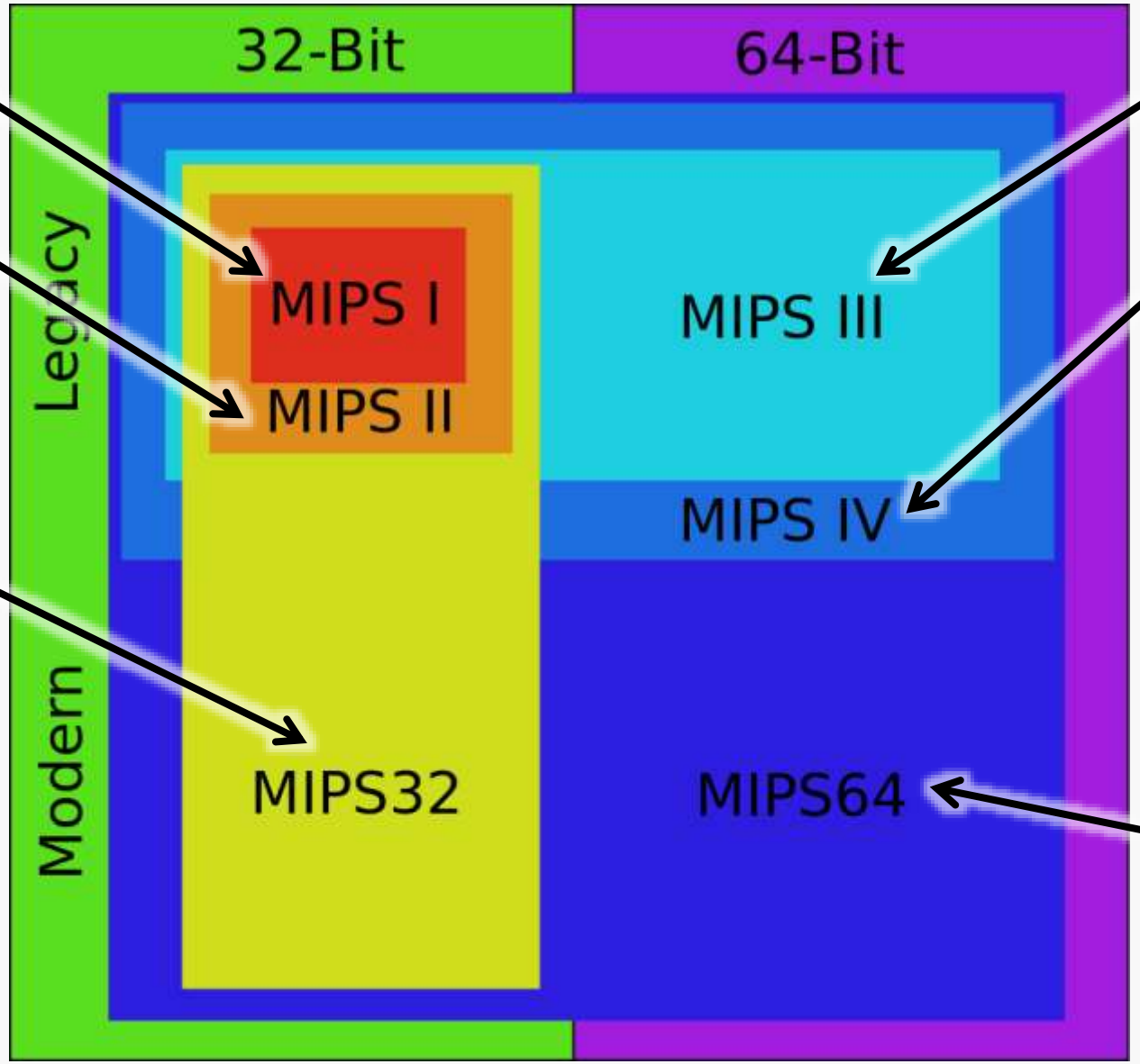
- MIPS处理器广泛应用的领域：
  - 数字电视、机顶盒、蓝光播放器、游戏机、网络设备等

MIPS指令的发展

1985年，R2000

1990年，R3000

1999年
以MIPS II为基础，
增加了MIPS
III/IV/V的部分特性



1992年，R4000
扩展到64位

1994年，R8000

MIPS V
1996年

1999年
以MIPS V为基础

MIPS的设计指导思想

- MIPS的全称
  - 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
  - A simpler CPU is a faster CPU.

MIPS指令的主要特点

- 固定的指令长度（32-bit，即1 word）
  - 简化了从存储器取指令
- 简单的寻址模式
  - 简化了从存储器取操作数
- 指令数量少，指令功能简单（一条指令只完成一个操作）
  - 简化指令的执行过程
- 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]`
- 需要优秀的编译器支持

MIPS指令示例（运算指令）

加法指令

- 格式： `add a, b, c`
- 操作：将b和c求和，结果存入a中

```
add a, b, c
sub a, b, c
mul a, b, c
div a, b, c
```

算术运算

```
and a, b, c
or  a, b, c
```

逻辑运算

```
sll a, b, c
srl a, b, c
```

移位

MIPS指令示例（访存指令）

- 假设
  - A是一个100个字（word）的数组，首地址在寄存器$19中
  - 变量h对应寄存器$18
  - 临时数据存放在寄存器$8

- 那么
  - `A[10]=h+A[3]` 对应的MIPS指令为：

```
lw   $8,12($19)  # t0=A[3]
add  $8,$18,$8   # t0=h+A[3]
sw   $8,40($19)  # A[10]=h+A[3]
```

MIPS的通用寄存器（32个，每个都是32位宽）

编号	名称	用途	编号	名称	用途
0	$zero	The Constant Value 0	24-25	$t8-$t9	Temporaries
1	$at	Assembler Temporary	26-27	$k0-$k1	Reserved for OS Kernel
2-3	$v0-$v1	Values for Function Results and Expression Evaluation	28*	$gp	Global Pointer
4-7	$a0-a3	Arguments	29*	$sp	Stack Pointer
8-15	$t0-$t7	Temporaries	30*	$fp	Frame Pointer
16-23*	$s0-$s7	Saved Temporaries	31*	$ra	Return Address

* Preserved across a call

通用寄存器使用示例

- 以下指令与对应注释中的指令相同

```
 lw   $t0,12($s3)
# lw   $8,12($19)

 add $t0,$s2,$t0
# add $8,$18,$8

 sw   $t0,40($s3)
# sw   $8,40($19)
```

编号	名称	用途
8-15	$t0-$t7	Temporaries
16-23	$s0-$s7	Saved Temporaries

本节小结

MIPS体系结构

北京大学·慕课
计算机组成
制作人：陆俊林