

SSC0611

Arquitetura de Computadores

2ª e 3ª Aulas – Arquitetura MIPS: ISA, Formato das instruções e Modos de endereçamento

Profa. Sarita Mazzini Bruschi
sarita@icmc.usp.br

Arquitetura MIPS

- MIPS: *Microprocessor without interlocked pipeline stages* (microprocessador sem estágios interligados de pipeline)
 - RISC – Patterson e Séquin (1980)
 - MIPS – Hennessy (1981)
 - Desenvolvida pela MIPS Computer System (fundada pelo Hennessy em 1984)
 - Comprada pela Silicon Graphics em 1992, que criou uma subsidiária chamada MIPS Technologies, Inc

ISA (*Instruction Set Architecture*)

- Quantidade e função dos registradores
 - 32 registradores de propósito geral
 - Todos começam com o símbolo \$
- Formato das Instruções
 - 3 tipos
- Modos de endereçamento
 - 5 modos

Registadores do MIPS

\$0	\$zero, \$r0	Sempre zero
\$1	\$at	Reservado para o assembler (assembler temporary)
\$2, \$3	\$v0, \$v1	Primeiro e segundo valores de retorno, respectivamente
\$4, ..., \$7	\$a0, ..., \$a3	Primeiros quatro argumentos para funções
\$8, ..., \$15	\$t0, ..., \$t7	Registadores temporários
\$16, ..., \$23	\$s0, ..., \$s7	Registadores salvos
\$24, \$25	\$t8, \$t9	Mais registadores temporários
\$26, \$27	\$k0, \$k1	Reservados para o Kernel do SO
\$28	\$gp	Global pointer (ponteiro global)
\$29	\$sp	Stack pointer (ponteiro para a pilha)
\$30	\$fp	Frame pointer (ponteiro para o frame)
\$31	\$ra	Return address (endereço de retorno)

Tipos das instruções

- Instruções Aritméticas e Simples
 - add, addi, addu, addiu,
- Instruções de Operandos Lógicos
 - and, andi, or, ori
- Instruções de Uso da Memória
 - lw, lh, lb, sw, sh, sb
- Instruções de Controle de Fluxo
 - beq, bne, j
- Instruções de Comparações
 - slt

Formato das instruções

- 3 formatos:
 - Tipo R (R-format), envolvendo registrador-registrador
 - Tipo I (I-format), envolvendo valor imediato
 - Tipo J (J-format), de desvio

3 Formatos de Instruções

OP	rs	rt	rd	sa	funct	R format
OP	rs	rt	immediate			I format
OP	jump target					J format

Tipo R



- Campos:
 - op: código da operação (opcode)
 - rs: número do primeiro registrador de origem
 - rt: número do segundo registrador de origem
 - rd: número do registrador destino
 - shamt: quantidade de bits a ser deslocado
 - funct: função específica a ser realizada

Tipo R

- Exemplos:

- add \$t2, \$t1, \$t0

000000	01001	01000	01010	00000	100000
Op = 0x0	rs = 9	rt = 8	rd = 10	shamt	f = 0x20

- sub \$s7, \$t8, \$0

000000	11000	00000	10111	00000	100010
Op = 0x0	rs = 24	rt = 0	rd = 23	shamt	f = 0x22

Tipo I

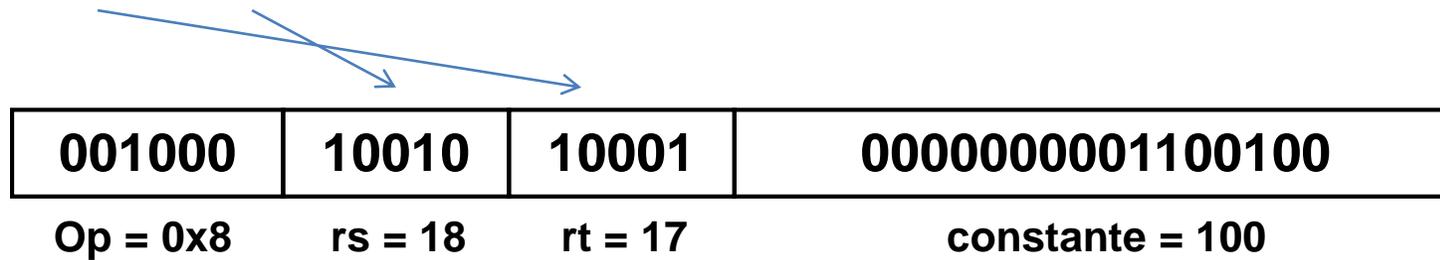


- Campos:
 - op: código da operação (opcode)
 - rs: número do registrador base a ser operado com o valor imediato ou operando
 - rt: número do registrador de destino ou operando
 - constante: -2^{15} a $+2^{15} - 1$
 - endereço:

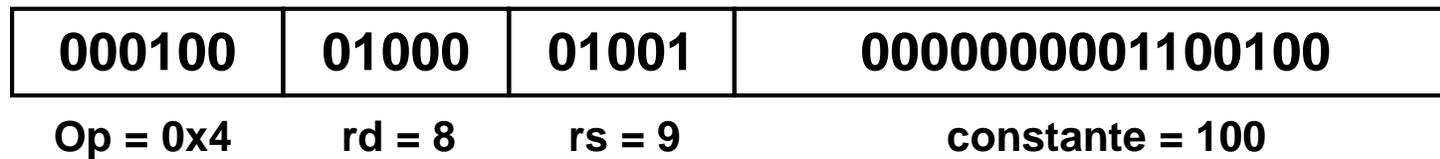
Tipo I

- Exemplos:

- addi \$s1, \$s2, 100



- beq \$t0, \$t1, 100



Tipo I

- Como carregar uma constante de 32 bits?
- Duas instruções:

lui \$t0, 101010101010101010 (load upper immediate)



ori \$t0, \$t0, 1010101010101010 (or immediate)



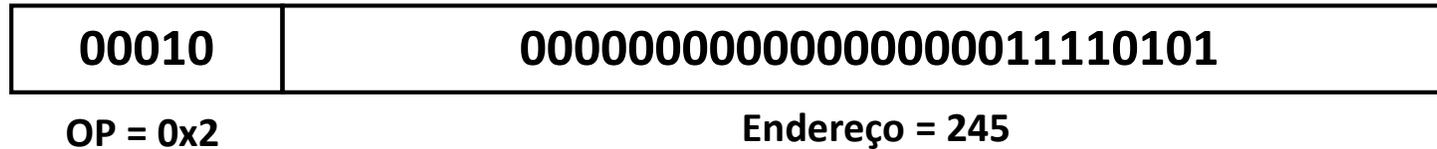
Tipo J



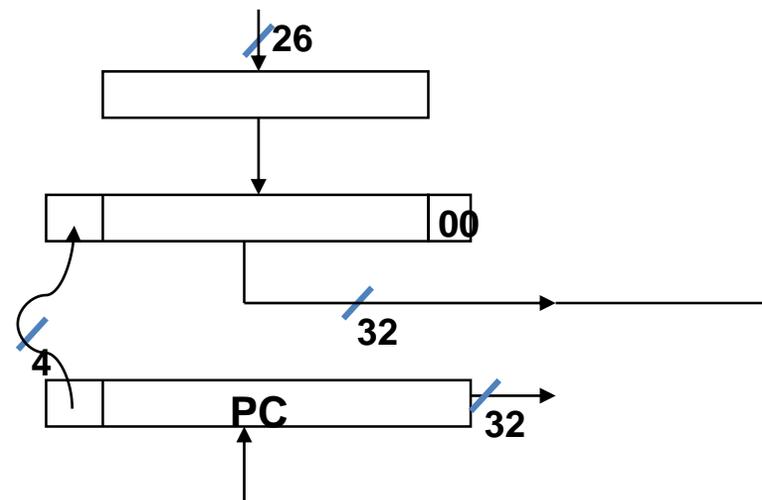
- Campos:
 - op: código da operação (opcode)
 - endereço: endereço destino

Tipo J

- Exemplos:
– j 245



- Como transformar esse valor de 26 bits em um endereço de 32 bits?



Modos de endereçamento

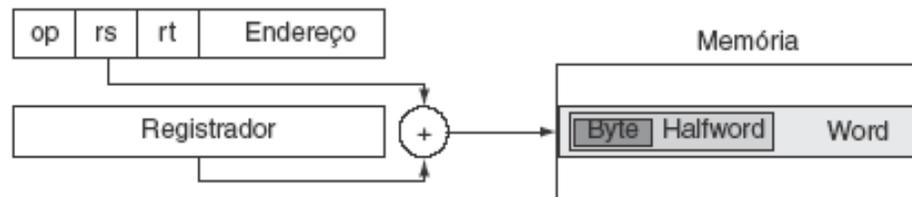
1. Endereçamento imediato



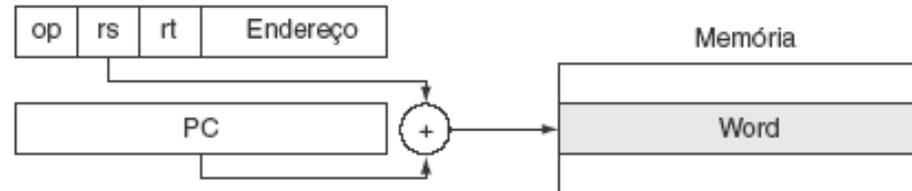
2. Endereçamento em registrador



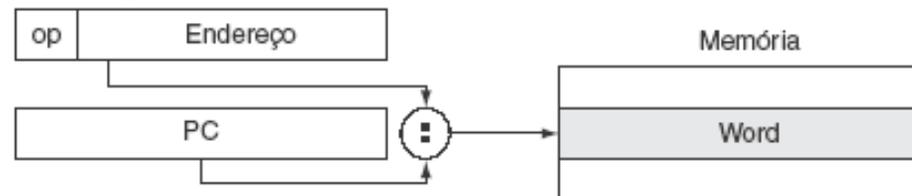
3. Endereçamento de base



4. Endereçamento relativo ao PC



5. Endereçamento pseudodireto



Modos de endereçamento

- Endereçamento imediato
 - `addi $s2, $s1, 100` (Tipo I)
- Endereçamento por registrador
 - `add $t2, $t1, $t0` (Tipo R)
- Endereçamento com base
 - `lw $t0, 8($s1)` (Tipo I)
- Relativo ao PC
 - `beq $t0, $t1, 28` (Tipo I)
- Endereçamento pseudo-direto
 - `j 300` (Tipo J)

Os 4 Princípios de Projeto do MIPS (RISC)

- Simplicidade favorece a regularidade
 - Instruções fixas de 32 bits
 - Pequeno número de formato de instruções
 - O opcode está sempre nos 6 bits mais significativos
- Bons projetos requerem compromissos
 - 3 formatos de instruções
- Menor é melhor
 - Conjunto de instruções limitado
 - Número de registradores no banco de registradores limitado
 - Número de modos de endereçamento limitado
- Torne o caso comum rápido
 - Operandos aritméticos estão no banco de registradores (máquina load-store)
 - Permitir que as instruções tenham operandos imediatos

MIPS Multiciclo

- Etapas:
 1. Busca da instrução
 2. Decodificação da instrução e busca dos registradores
 3. Execução, cálculo do endereço de memória ou conclusão do desvio
 4. Acesso à memória ou conclusão de instrução tipo R
 5. Conclusão da leitura da memória

MIPS Multiciclo

Passo	Instrução tipo R	Instruções de referência à memória	Branches	Jumps
Busca da instrução	$IR \leq \text{Memory}[PC]$ $PC \leq PC + 4$			
Decodificação da instrução e busca dos registradores	$A \leq \text{Reg}[IR[25:21]]$ $B \leq \text{Reg}[IR[20:16]]$ $ALUOut \leq PC + (\text{sign-extend}(IR[15:0]) \ll 2)$			
Execução, cálculo do endereço de memória ou conclusão do desvio	$ALUOut \leq A \text{ op } B$	$ALUOut \leq A + \text{sign-extend}(IR[15:0])$	$\text{If } (A == B)$ $PC \leq ALUOut$	$PC \leq \{PC[31:28],$ $(IR[25:0], 2'b00)\}$
Acesso à memória ou conclusão de instrução tipo R	$\text{Reg}[IR[15:11]] \leq ALUOut$	Load: $MDR \leq \text{Memory}[ALUOut]$ ou Store: $\text{Memory}[ALUOut] \leq B$		
Conclusão da leitura da memória		Load: $\text{Reg}[IR[20:16]] \leq MDR$		