

Relatório MIPS

Alunos: Andresa Bicudo, Beatriz Cabral, Giovanni Santos

Professor: Paulo Carlos Santos

1. Dados do Processador

As instruções implementadas neste processador possuem 32 bits de comprimento e o acesso à memória é feito com granularidade de 4 bytes por acesso. Apesar disso, o endereçamento de memória é orientado a apenas 1 byte que se refere a um endereço. Os endereços de memória também possuem 32 bits justamente para simplificar o hardware de carga e decodificação de instruções.

2. Arquitetura do tipo Load/Store

O MIPS só pode fazer operações lógicas e aritméticas entre registradores ou entre registradores e constantes imediatas. Um registrador recebe um valor armazenado na memória através de uma instrução que só faz a carga. Por outro lado, o conteúdo de um registrador pode ser armazenado na memória através de uma instrução específica para isso.

3. Banco de Registradores

O MIPS possui 32 registradores de uso geral e outros 32 registradores de ponto flutuante. O hardware MIPS não impõe um uso específico para os registradores de uso geral (exceto r0). Ou seja, onde um registrador é necessário, qualquer registrador funcionará. Porém, utilizaremos a seguinte convenção para o uso dos registradores, evoluiu como padrão para a programação MIPS.

Convenção de Uso dos Registradores			
Registradores	Nome	Descrição	Quem Salva
0	\$zero	Sempre retorna 0, não é modificável. A escrita é ignorada.	N. A.
1	\$at	Reservado para uso do montador (<i>assembler temporary</i>).	Rotina que Chama
2-3	\$v0, \$v1	Valores retornados pela sub-rotina.	Rotina que Chama
4-7	\$a0-\$a3	(argumentos) Quatro primeiros parâmetros para a sub-rotina.	Rotina que Chama
8-15	\$t0-\$t7	(temporários) Sub-rotinas pode usar sem salvar.	Rotina que Chama
16-23	\$s0-\$s7	Para variáveis da sub-rotina, precisa restaurar antes de retornar.	Rotina Chamada
24-25	\$t8, \$t9	(temporários) As sub-rotinas podem usar sem salvar.	Rotina que Chama
26-27	\$k0, \$k1	Reservado para uso do manipulador da interrupção/trap.	Rotina que Chama
28	\$gp	Ponteiro Global; usado para acessar variáveis " <i>static</i> " ou " <i>extern</i> ".	Rotina Chamada
29	\$sp	Ponteiro da pilha (<i>stack pointer</i>).	Rotina Chamada
30	\$s8/\$fp	Ponteiro da estrutura (<i>frame pointer</i>) ou nona variável da sub-rotina.	Rotina Chamada
31	\$ra	Endereço de retorno para chamada de sub-rotina (JAL & JALR).	Rotina que Chama

Também existem outros registradores de uso específico, dos quais:

- PC, contador do programa, que armazena o endereço da instrução em execução (passa de PC para PC+4);
- EPC, registrador com o endereço da instrução em que ocorreu a interrupção. Esse é endereço de retorno da interrupção;

4. Endereçamento

Forma como é definido o endereço de destino para uma instrução de desvio. No caso do MIPS, ele se restringe a instruções de leitura e escrita na memória e as instruções de desvio.

O endereçamento pode ser:

- Direto Imediato:
 - Usa como destino um ponto definido no programa (label). O deslocamento é dado pelo valor do imediato (16 bits com sinal) dentro da instrução.
- Indireto:
 - Usa o valor de um registrador como endereço.
- Indexado:
 - Utiliza um endereço base, dado pelo conteúdo de um registrador;
 - Somado a um deslocamento definido na instrução;
 - O deslocamento pode ser negativo.

Abaixo temos alguns exemplos de instruções:

1. Endereçamento Imediato

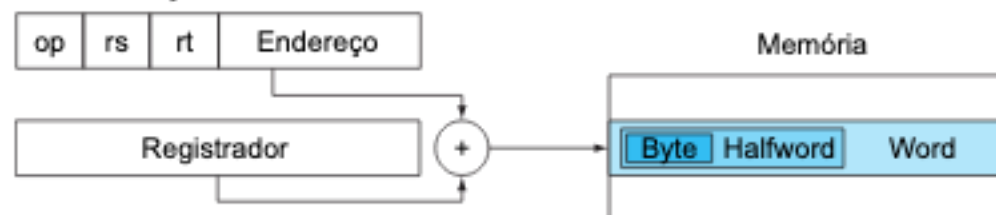


COD 4Ed, Copyright © 2009
Elsevier, Inc. All rights reserved.

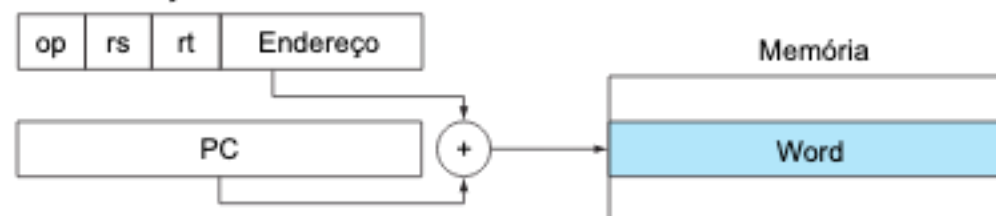
2. Endereçamento em Registrador



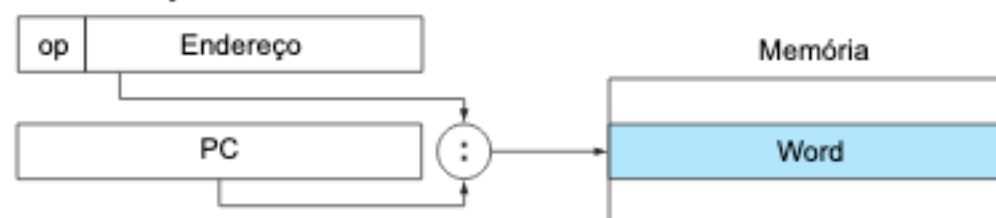
3. Endereçamento de Base



4. Endereçamento Relativo ao PC



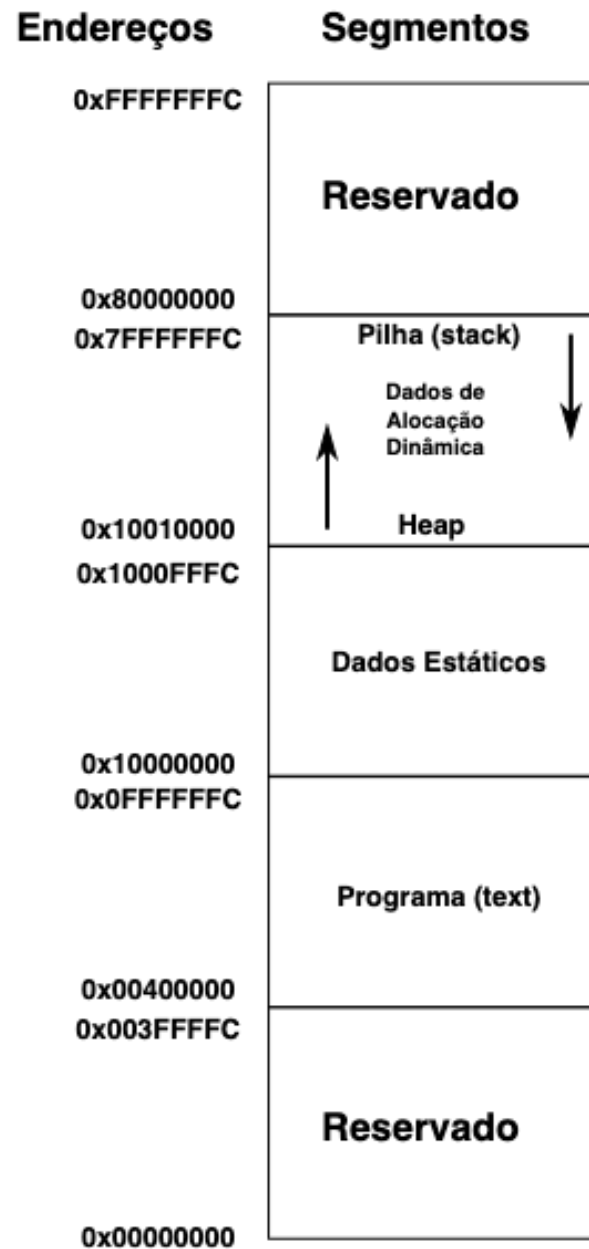
5. Endereçamento Pseudodireto



5. Organização da Memória

O MIPS utiliza os segmentos de memória mostrada abaixo. Na inicialização, os ponteiros para esses segmentos estarão com os seguintes valores:

- O *stack* pointer que aponta para: 0x7FFFFFFC;
- O global pointer que aponta para: 0x10008000;
- O *program counter* aponta para: 0x00400000;



6. Instruções e Pontos de Controle

Instrução	Mnemônico	Código Binário	habMuxPC	habMuxRtRd	habEscReg	habMuxRtlmediato	ULAop	habMuxULAMEM	BEQ	RD	WR
Sem Operação	NOP	000000	0	0	0	0	XX	0	0	0	0
Lê da memória	LW	000001	0	0	1	1	00	1	0	1	0
Escrever na memória	SW	000010	0	0	0	1	00	X	0	0	1
Soma B de A e armazena em A	ADD	000011	0	1	1	0	00	0	0	0	0
Subtrai B de A e armazena em A	SUB	000100	0	1	1	0	01	0	0	0	0
E lógico	AND	000101	0	1	1	0	10	0	0	0	0
OU lógico	OR	000110	0	1	1	0	10	0	0	0	0
Comparação Menor Que	SLT	000111	0	1	1	0	10	0	0	0	0
Compara Igual	BEQ	001000	0	X	X	0	10	X	1	0	0
Desvio incondicional de execução	J	001001	1	1	X	X	XX	X	X	X	X

