



Infra-Estrutura de Hardware

MIPS – Set de instruções

Universidade Federal Rural de Pernambuco
Professor: Abner Corrêa Barros
abnerbarros@gmail.com

Tipos de Instruções

- A fim de manter os princípios de simplicidade e regularidade, todas as instruções do MIPS podem ser agrupadas em três formatos básicos:
 - Instruções registrador-registrador (R-type)
 - Instruções envolvendo valor imediato (I-type)
 - Instruções de desvio (J-type)

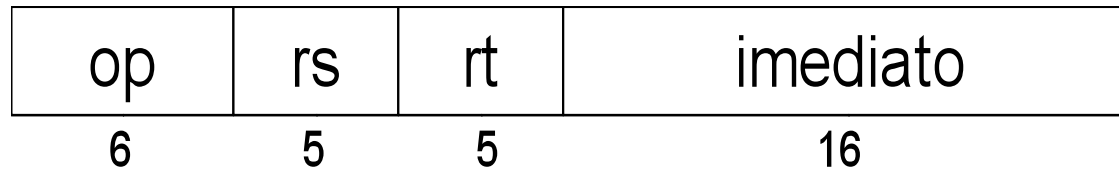
Representação física das Instruções R-type



op	rs	rt	rd	shamt	funct
6	5	5	5	5	6

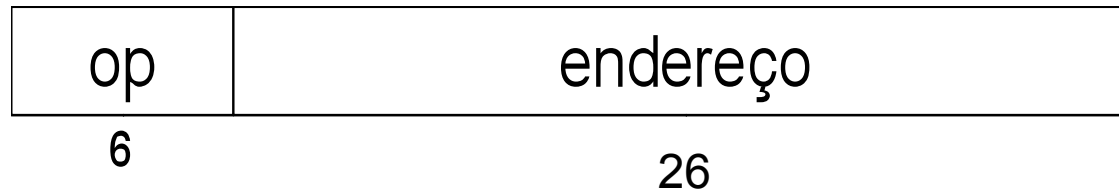
- op: operação básica a ser realizada (opcode)
- rs: primeiro operando
- rt: segundo operando
- rd: registrador de destino (resultado)
- shamt: quantidade de bits a ser deslocado
- funct: função específica a ser executada

Representação física das Instruções I-type



- op: operação básica a ser realizada (opcode)
- rs: registrador base a ser operado com o valor imediato ou operando
- rt: registrador de destino ou operando
- imediato: valor constante a ser operado

Representação física das Instruções J-Type



- op: operação básica a ser realizada (opcode)
- endereço: endereço da instrução para onde o programa deve ser desviado

Instruções Básicas



- Lógicas e Aritméticas

Mnm	Fmt	Operação	Obs
Add	R	$R[rd] = R[rs] + R[rt]$	Operandos com sinal, complemento a 2
Addi	I	$R[rt] = R[rs] + Im$	Im = Inteiro com sinal, complemento a 2
Sub	R	$R[rd] = R[rs] - R[rt]$	Operandos com sinal, complemento a 2
And	R	$R[rd] = R[rs] \& R[rt]$	
Andi	I	$R[rt] = R[rs] \& Im$	
Or	R	$R[rd] = R[rs] R[rt]$	
Ori	I	$R[rt] = R[rs] Im$	
Nor	R	$R[rd] = \sim(R[rs] R[rt])$	

Instruções Básicas

Exemplo

- add \$10, \$9, \$8

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

- addi \$8, \$10, 100

001000	01010	01000	00000000001100100
Op = 0x8	rs = 10	rt = 8	lm = 0x64

Exercícios



- Implemente as seguintes operações em linguagem de máquina
 - Add \$5, \$6, \$7
 - Addi \$16, \$17, 53
 - Sub \$23, \$24, \$0
 - Ori \$4, \$4, 0x45

Exercícios



- Converta o código a seguir em linguagem de montagem do MIPS
 - $a = a + b$
 - $x = y + 3$
 - $a = a \mid 0x41$
 - $b = c \& d$

Exercícios



- Converta as seguintes instruções de linguagem de máquina para linguagem de montagem do MIPS
 - 0x21480064
0010 0001 0100 1000 0000 0000 0110 0100
 - 0x01285022
0000 0001 0010 1000 0101 0000 0010 0010

Instruções Básicas



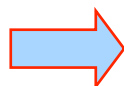
- Desvio e tomada de decisão

Mnm	Fmt	Operação	Obs
Jr	R	$PC = R[rs]$	Desvia para o endereço armazenado em rs
J	J	$PC = \text{endereço}$	Desvia para o endereço indicado * 4
Jal	J	$PC = \text{endereço}$	Desvia para o endereço indicado*4, $R31=PC+4$
Beq	I	If ($R[rs]==R[rt]$) $PC=PC+4+Im$	Desvia para o endereço indicado em $Im*4$ se o conteúdo de rs for igual ao de rt
Bneq	I	If ($R[rs]!=R[rt]$) $PC=PC+4+Im$	Desvia para o endereço indicado em $Im*4$ se o conteúdo de rs for diferente ao de rt

Instruções Básicas

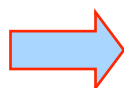
Exemplo

```
While (a!=0){  
    c = a + b;  
    a = a--;  
}
```



```
Inicio: Beq $1, $0, fim  
        Add $3, $1, $2  
        Addi $1, $1, -1  
        J    Inicio  
Fim:
```

```
For (i=0, i !=10, i++){  
    c = a + b;  
}
```



```
Inicio: Add $1, $0, $0  
Loop: Beq $1, 10, fim  
       Addi $1, $1, 1  
       Add $3, $1, $2  
       J    Loop  
Fim:
```

Instruções Básicas



Exemplos

- Jr \$10

000000	01010	00000	00000	00000	001000
--------	-------	-------	-------	-------	--------

Op = 0x0 rs = 10 rt = 0 rd = 0 shamt f = 0x08

- J 100

000010	00000000000000000000000001100100
--------	----------------------------------

Op = 0x2

Im = 0x64

Estratégia adotada nas instruções J e Jal



- A fim de aumentar o alcance das instruções de desvio, que contam com apenas 26 bits no campo do valor imediato, contra os 32 bits do registrador PC, armazena-se neste campo a instrução a ser carregada e não o endereço de memória.
- Desta forma, para descobrir o endereço de memória a ser carregado no PC, deve-se multiplicar o conteúdo do campo Im por 4. O resultado obtido é então carregado para os bits 0 à 27 do PC.

Estratégia adotada nas instruções J e Jal



J 0x8000

Im

00 0000 0000 1000 0000 0000 0000

 0x000 8000



End

0000 0000 0010 0000 0000 0000 0000

 0x002 0000



PC

1000	0000 0000 0010 0000 0000 0010 0100
------	------------------------------------

 0x8002 0024

PC

1000	0000 0000 0010 0000 0000 0000 0000
------	------------------------------------

 0x8002 0000

Instruções Básicas

Exemplos

- Bne \$2, \$3, 100

000101	00010	00011	00000000001100100
--------	-------	-------	-------------------

Op = 0x05

rs = 2

rt = 3

Im = 0x64

- Beq \$2, \$3, 100

000100	00010	00011	00000000001100100
--------	-------	-------	-------------------

Op = 0x4

rs = 2

rt = 3

Im = 0x64

Instruções Básicas

Transferencia de Dados e verificação de conteúdo

Mnm	Fmt	Operação	Obs
Lw	I	$R[rt] = M[R[rs] + Im]$	Carrega registrado com o conteúdo da memória
lui	I	$R[rt] = Im, 0x0000$	Carrega a parte alta do registrador
sw	I	$M[R[rs] + Im] = R[rt]$	Carrega memória com o conteúdo do registrador
slt	R	$R[rd] = (R[rs] < R[rt] ? 1 : 0)$	Carrega o registrador com 1 se $R[rs] < R[rt]$ caso contrário carrega com 0
slti	I	$R[rd] = (R[rs] < Im ? 1 : 0)$	Carrega o registrador com 1 se $R[rs] < Im$ caso contrário carrega com 0

Instruções Básicas

Exemplos

- Lui \$8, 0x8002

0011 11	00000	01000	1000 0000 0000 0010
Op = 0xF	rs = 00	rt = 8	Im = 0x8002

- Lw \$8, 0x30(\$8)

100011	01000	01000	0000 0000 0011 0000
Op = 0x23	rs = 8	rt = 8	Im = 0x30

Instruções Básicas



Exemplos

- Sw \$11, 0x30(\$01)

101011	00001	01011	0000 0000 0011 0000
--------	-------	-------	---------------------

Op = 0x2b rs = 1 rt = 11 lm = 0x30

- Slt \$10, \$9, \$8

000000	01001	01000	01010	00000	101010
--------	-------	-------	-------	-------	--------

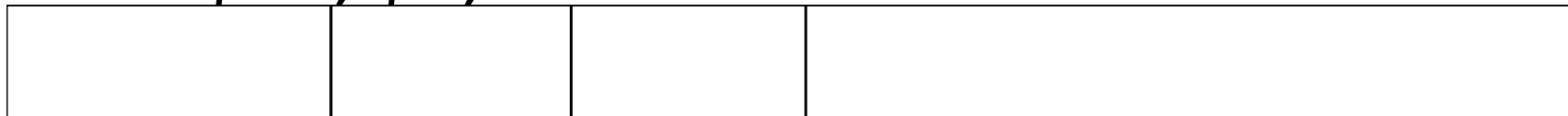
Op = 0x00 rs = 9 rt = 8 rd = 10 shamt f = 0x2a

Instruções Básicas



Exemplos

- Slti \$11, \$9, 0x01



Op =

rs =

rt =

Im =

Exercícios



Implemente o seguinte trecho de código em linguagem de montagem:

```
a = 0x30;  
b = 0x1;  
While (a>5){  
    b = b + 5;  
    a--;  
}
```

a e b estão armazenados respectivamente nas posições 0x8002 0050 e 0x8002 0054 da memória