



Universidade Federal de Pelotas
Centro de Desenvolvimento Tecnológico
Bacharelado em Ciência da Computação
Engenharia de Computação

Arquitetura e Organização de Computadores I

Prática

Aula 3

Revisão, Multiplicação e Divisão

Prof. Guilherme Corrêa
gcorrea@inf.ufpel.edu.br

Prof. Bruno Zatt

Revisão

► MIPS: Registradores

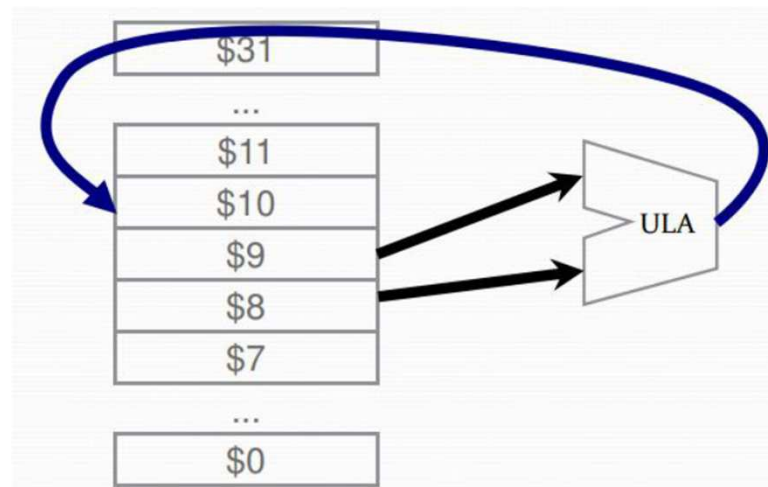
Registrador	Nome	Uso (convenção)
\$0	\$zero	Zero
\$1	\$at	<i>Assembler Temporary</i>
\$2, \$3	\$v0, \$v1	Valor de retorno de subrotina
\$4 – \$7	\$a0 – \$a3	Argumentos de subrotina
\$8 – \$15	\$t0 – \$t7	Temporários (locais à função)
\$16 – \$23	\$s0 – \$s7	Salvos (não alterados na função)
\$24, \$25	\$t8, \$t9	Temporários
\$26, \$27	\$k0, \$k1	Kernel (reservado para SO)
\$28	\$gp	<i>Global Pointer</i>
\$29	\$sp	<i>Stack Pointer</i>
\$30	\$fp	<i>Frame Pointer</i>
\$31	\$ra	Endereço de Retorno

Revisão

► MIPS: Unidade Lógica e Aritmética (ULA)

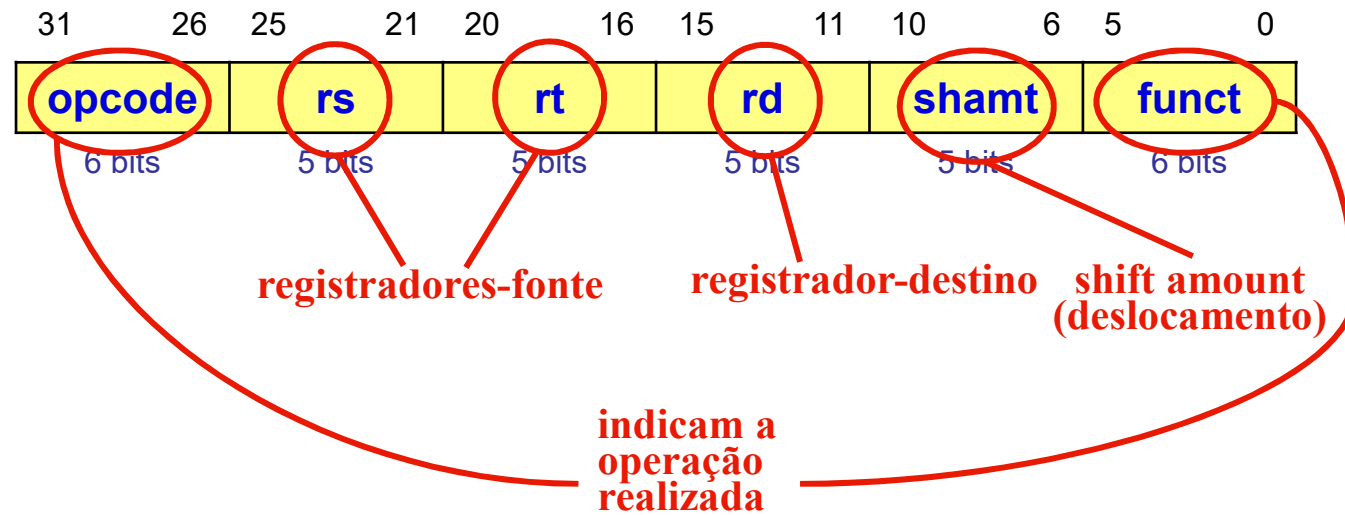
- Circuito responsável pelas operações lógicas e aritméticas
- Exemplo:

and \$10, \$8, \$9



Revisão

► Instruções do Tipo R (Registrador)



Revisão

► Operações Lógicas (and, or, xor, nor)

Tipo R

31	26	25	21	20	16	15	11	10	6	5	0									
opcode						rs			rt			rd			shamt			funct		
6 bits						5 bits			5 bits			5 bits			5 bits			6 bits		

and \$t1, \$zero, \$t3

0H	0H	BH	9H	0H	24H
000000	00000	01011	01001	00000	100100

or \$t0, \$t1, \$t2

0H	9H	AH	8H	0H	25H
000000	01001	01010	01000	00000	100101

xor \$t1, \$t2, \$t3

0H	AH	BH	9H	0H	26H
000000	01010	01011	01001	00000	100110

nor \$t0, \$t0, \$t3

0H	8H	BH	8H	0H	27H
000000	01011	01011	01001	00000	100111

Revisão

► Operações Lógicas (sll, slr)

Tipo R

31	26	25	21	20	16	15	11	10	6	5	0				
opcode						rs		rt		rd		shamt		funct	
6 bits						5 bits		5 bits		5 bits		5 bits		6 bits	

sll \$t0, \$t1, 12

0 _H	0 _H	9 _H	8 _H	12 _H	0 _H
000000	00000	01001	01000	01100	000000

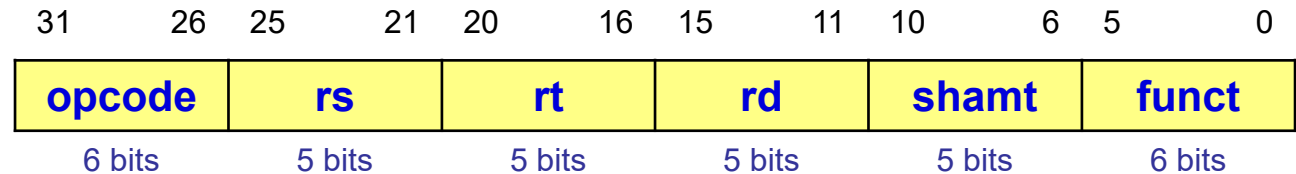
srl \$t5, \$t5, 28

0 _H	0 _H	D _H	D _H	1C _H	2 _H
000000	00000	01101	01101	11100	000010

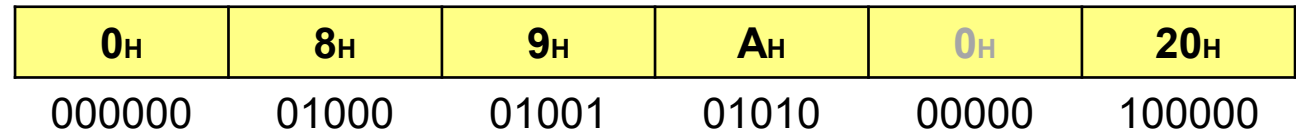
Revisão

► Operações Aritméticas (add, sub)

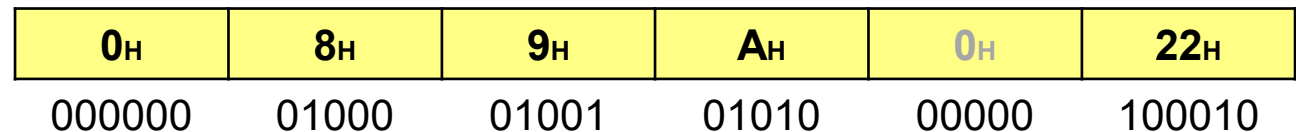
Tipo R



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

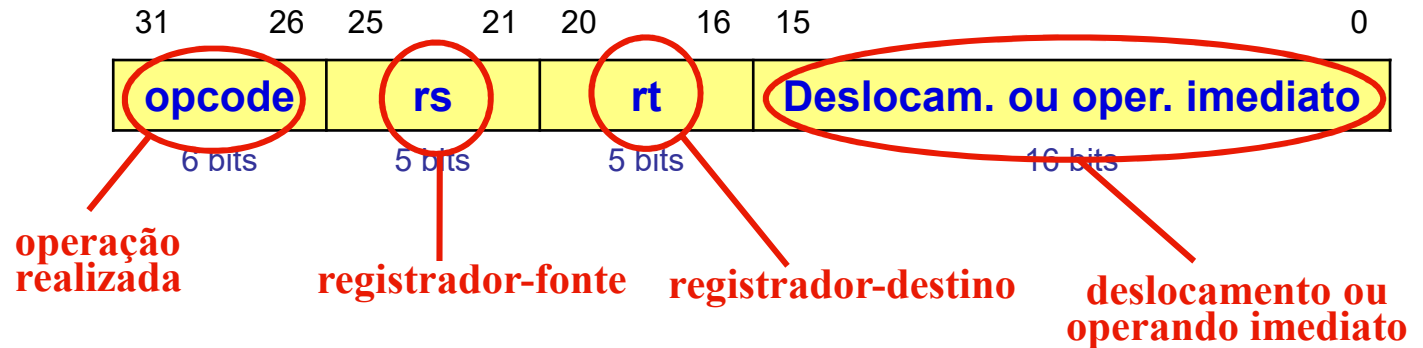


sub \$t2, \$t0, \$t1



Revisão

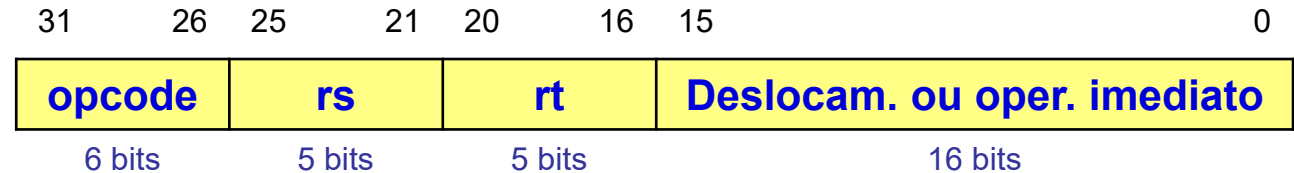
► Instruções do Tipo I (Imediato)



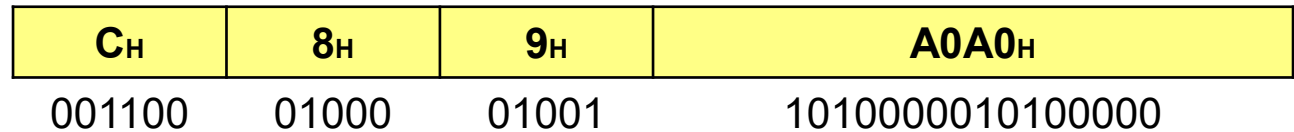
Revisão

► Operações Lógicas (andi, ori, xori)

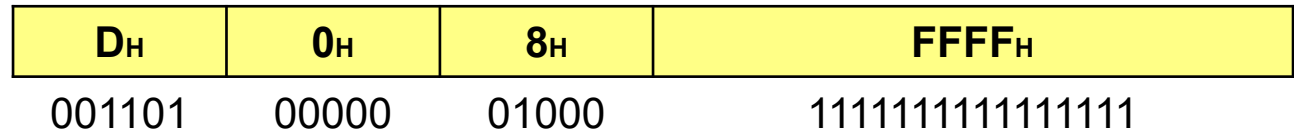
Tipo I



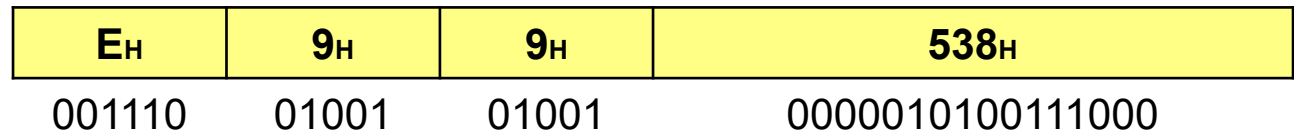
andi \$t1, \$t0, 0xA0A0



ori \$t0, \$zero, 0xFFFF



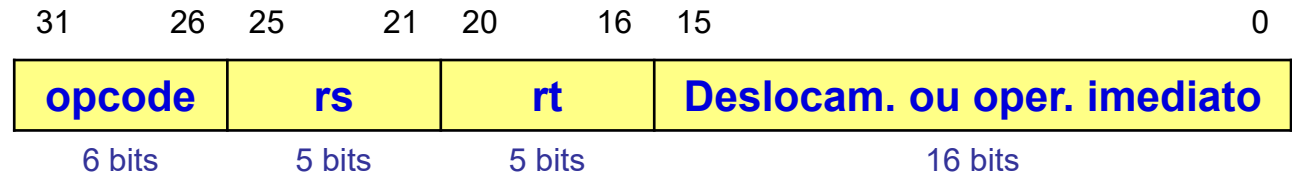
xori \$t1, \$t1, 0x538



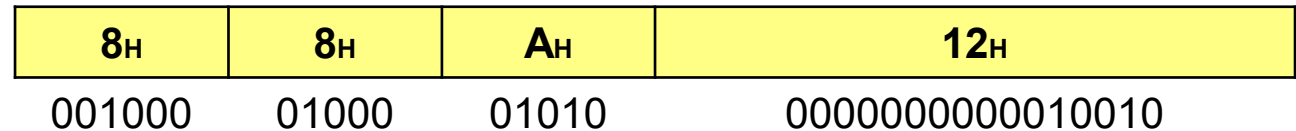
Revisão

► Operações Aritméticas (addi)

Tipo I



addi \$t2, \$t0, 0x12



Operações Aritméticas (parte 2)

► Multiplicação

em negrito: bits significativos

Multiplicando	x	<div>0010_B</div>	rs
Multiplicador		<div>1011_B</div>	rt
<hr/>			
		0010	
	+	0010	
		0000	
		0010	
<hr/>			
Produto		<div>0010110_B</div>	
		hi lo	

	x	<div>1111_B</div>	rs
		<div>1111_B</div>	rt
<hr/>			
		1111	
	+	1111	
		1111	
		1111	
<hr/>			
		<div>11100001_B</div>	
		hi lo	

**O resultado da multiplicação de n bits por m bits
pode ter até $n+m$ bits!**

Operações Aritméticas (parte 2)

► Multiplicação

Tipo R

31	26	25	21	20	16	15	11	10	6	5	0
opcode			rs		rt		rd		shamt		func
6 bits			5 bits		5 bits		5 bits		5 bits		6 bits

mult \$t0, \$t1

0H	8H	9H	0H	0H	18H
000000	01000	01001	00000	00000	011000

multu \$t0, \$t1

0H	8H	9H	0H	0H	19H
000000	01000	01001	00000	00000	011001

mfhi \$t2

0H	0H	0H	AH	0H	10H
000000	00000	00000	01010	00000	010000

mflo \$t3

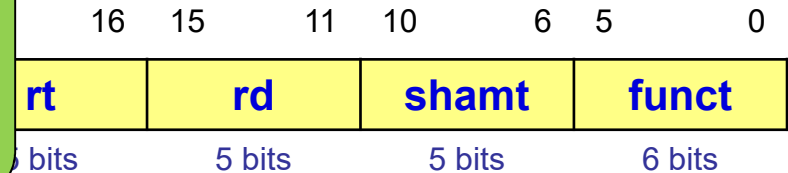
0H	0H	0H	BH	0H	12H
000000	00000	00000	01011	00000	010010

Operações Aritméticas (parte 2)

► Multiplicação

Tipo R

Multiplicação de
inteiros com sinal
hilo \leftarrow \$t0 * \$t1



mult \$t0, \$t1

0H	8H	9H	0H	0H	18H
000000	01000	01001	00000	00000	011000

multu \$t0, \$t1

0H	8H	9H	0H	0H	19H
000000	01000	01001	00000	00000	011001

mfhi \$t2

0H	0H	0H	AH	0H	10H
000000	00000	00000	01010	00000	010000

mflo \$t3

0H	0H	0H	BH	0H	12H
000000	00000	00000	01011	00000	010010

Operações Aritméticas (parte 2)

► Multiplicação

Tipo R

Multiplicação de
inteiros sem sinal
hilo $\leftarrow \$t0 * \$t1$

mult \$t0, \$t1

multu \$t0, \$t1

mfhi \$t2

mflo \$t3

31	26	25	21	20	16	15	11	10	6	5	0
rt					rd			shamt		funct	
16 bits					5 bits			5 bits		6 bits	
0H	0H	0H	0H	0H	0H	0H	0H	0H	0H	18H	
000000	010000	010001	000000	000000	000000	000000	000000	000000	000000	011000	
0H	8H	9H	0H	0H	0H	0H	0H	0H	0H	19H	
000000	010000	010001	000000	000000	000000	000000	000000	000000	000000	011001	
0H	0H	0H	AH	0H	0H	0H	0H	0H	0H	10H	
000000	000000	000000	010100	000000	000000	000000	000000	000000	000000	010000	
0H	0H	0H	BH	0H	0H	0H	0H	0H	0H	12H	
000000	000000	000000	010110	000000	000000	000000	000000	000000	000000	010010	

Operações Aritméticas (parte 2)

► Multiplicação

Tipo R

31	26	25	21	20	16	15	11	10	6	5	0				
opcode						rs		rt		rd		shamt		funct	
6 bits						5 bits		5 bits		5 bits		5 bits		6 bits	

mult \$t0, \$t1

multu \$t0, \$t1

mfhi \$t2

mflo \$t3

“move from hi”
 Move o resultado
 em **hi** para \$t2
\$t2 ← hi

9H	0H	0H	18H
----	----	----	-----

01001 00000 00000 011000

9H	0H	0H	19H
----	----	----	-----

000000 01000 01001 00000 00000 011001

0H	0H	0H	AH	0H	10H
----	----	----	----	----	-----

000000 00000 00000 01010 00000 010000

0H	0H	0H	BH	0H	12H
----	----	----	----	----	-----

000000 00000 00000 01011 00000 010010

Operações Aritméticas (parte 2)

► Multiplicação

Tipo R

31	26	25	21	20	16	15	11	10	6	5	0				
opcode						rs		rt		rd		shamt		funct	
6 bits						5 bits		5 bits		5 bits		5 bits		6 bits	

mult \$t0, \$t1

0H	8H	9H	0H	0H	18H
----	----	----	----	----	-----

multu \$t0, \$t1

0H	9H	0H	0H	19H
----	----	----	----	-----

mfhi \$t2

0H	AH	0H	10H
----	----	----	-----

mflo \$t3

0H	0H	0H	BH	0H	12H
----	----	----	----	----	-----

“move from lo”

Move o resultado
em **lo** para \$t3

$\$t3 \leftarrow lo$

01001 00000 00000 011000

01001 00000 00000 011001

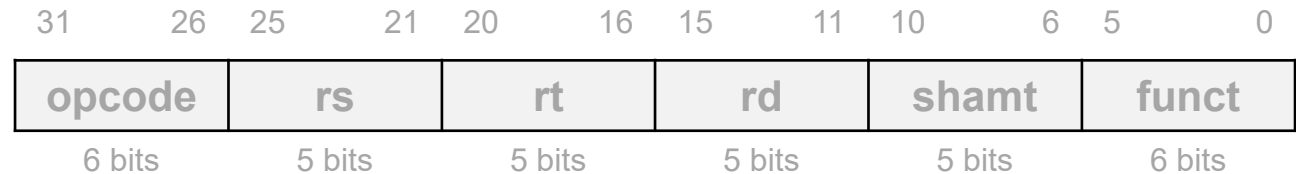
000000 00000 00000 01010 00000 010000

000000 00000 00000 01011 00000 010010

Operações Aritméticas (parte 2)

► Multiplicação

Tipo R



mult \$t0,

multu \$t0,

mfhi \$t2

mflo \$t3

Importante!!!

Instruções de multiplicação **NÃO PODEM** ser realizadas imediatamente após **mfhi** e **mflo**.

18H

011000

19H

011001

0H

0H

0H

AH

0H

10H

000000

000000

000000

01010

000000

010000

0H

0H

0H

BH

0H

12H

000000

000000

000000

01011

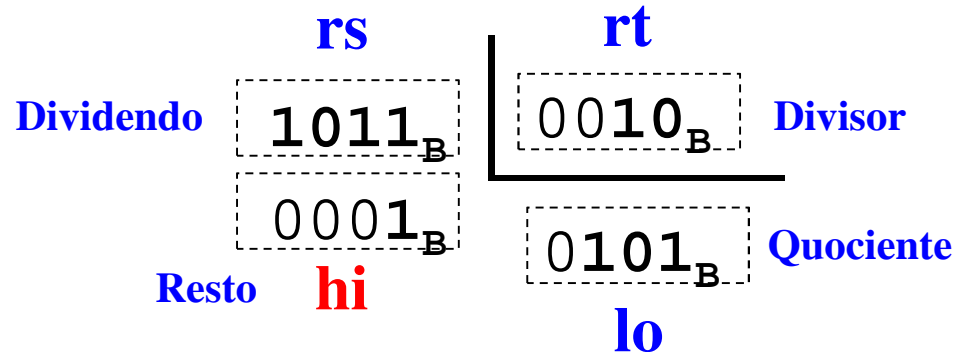
000000

010010

Operações Aritméticas (parte 2)

► Divisão

em **negrito**: bits significativos



Operações Aritméticas (parte 2)

► Divisão

Tipo R

31	26	25	21	20	16	15	11	10	6	5	0				
opcode						rs		rt		rd		shamt		funct	
6 bits						5 bits		5 bits		5 bits		5 bits		6 bits	

div \$t0, \$t1

0H	8H	9H	0H	0H	1AH
000000	01000	01001	00000	00000	011010

divu \$t0, \$t1

0H	8H	9H	0H	0H	1BH
000000	01000	01001	00000	00000	011011

mfhi \$t2

0H	0H	0H	AH	0H	10H
000000	00000	00000	01010	00000	010000

mflo \$t3

0H	0H	0H	BH	0H	12H
000000	00000	00000	01011	00000	010010

Operações Aritméticas (parte 2)

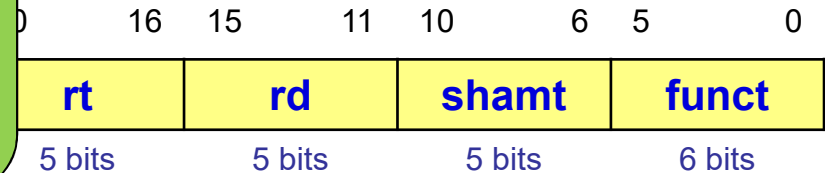
► Divisão

Tipo R

Divisão de inteiros
com sinal

hi $\leftarrow \$t0 \% \$t1$

lo $\leftarrow \$t0 / \$t1$



div \$t0, \$t1

0H	8H	9H	0H	0H	1AH
000000	01000	01001	00000	00000	011010

divu \$t0, \$t1

0H	8H	9H	0H	0H	1BH
000000	01000	01001	00000	00000	011011

mfhi \$t2

0H	0H	0H	AH	0H	10H
000000	00000	00000	01010	00000	010000

mflo \$t3

0H	0H	0H	BH	0H	12H
000000	00000	00000	01011	00000	010010

Operações Aritméticas (parte 2)

► Divisão

Tipo R

Divisão de inteiros
sem sinal

hi \leftarrow \$t0 % \$t1

lo \leftarrow \$t0 / \$t1

div \$t0, \$t1

divu \$t0, \$t1

mfhi \$t2

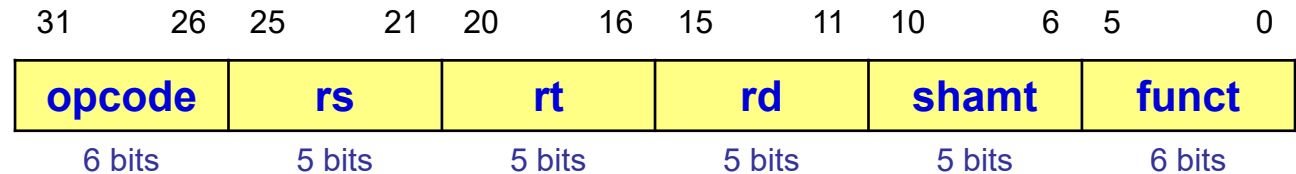
mflo \$t3

	16	15	11	10	6	5	0
	rt	rd	shamt	func			
	5 bits	5 bits	5 bits	6 bits			
	9H	0H	0H	1AH			
	01001	00000	00000	011010			
	0H	8H	9H	0H	0H	1BH	
	000000	01000	01001	00000	00000	011011	
	0H	0H	0H	AH	0H	10H	
	000000	00000	00000	01010	00000	010000	
	0H	0H	0H	BH	0H	12H	
	000000	00000	00000	01011	00000	010010	

Operações Aritméticas (parte 2)

► Divisão

Tipo R



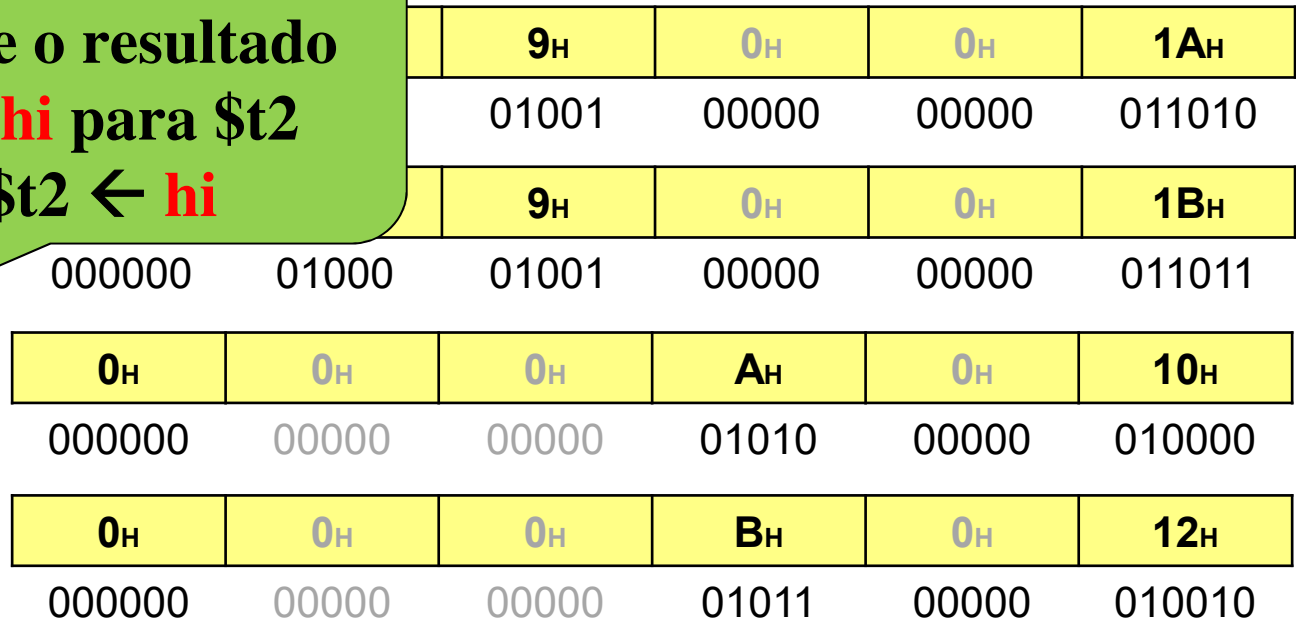
div \$t0, \$t1

divu \$t0, \$t1

mfhi \$t2

mflo \$t3

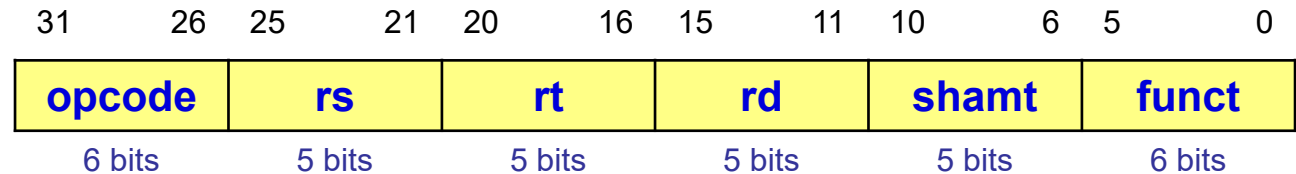
“move from hi”
 Move o resultado
 em **hi** para \$t2
\$t2 ← hi



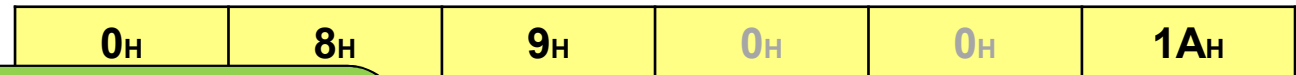
Operações Aritméticas (parte 2)

► Divisão

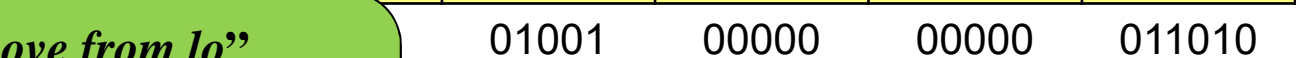
Tipo R



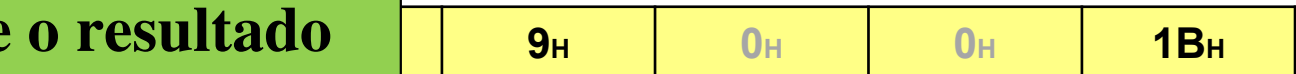
div \$t0, \$t1



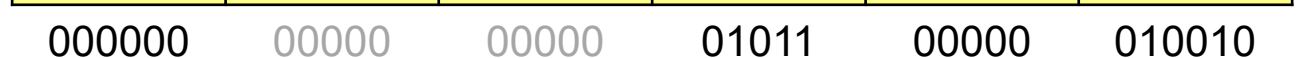
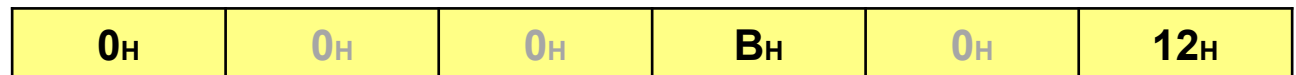
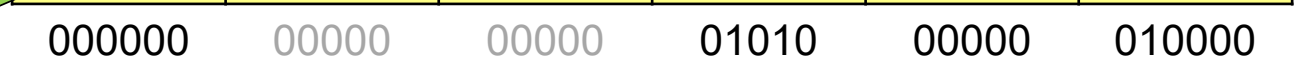
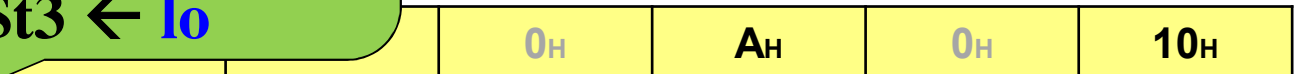
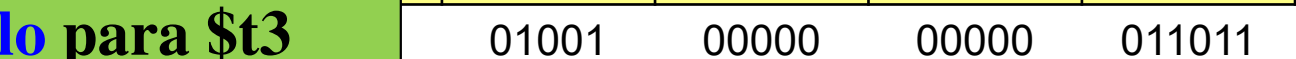
divu \$t0, \$t1



mfhi \$t2



mflo \$t3

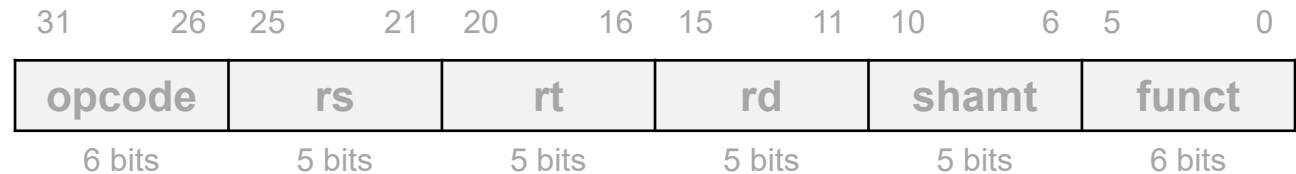


“move from lo”
 Move o resultado
 em **lo** para \$t3
 \$t3 ← **lo**

Operações Aritméticas (parte 2)

► Divisão

Tipo R



div \$t0, \$t1

divu \$t0, \$t1

mfhi \$t2

mflo \$t3

Importante!!!
Instruções de divisão **NÃO PODEM** ser realizadas imediatamente após **mfhi** e **mflo**.

1AH

011010

1BH

011011

0H

0H

0H

AH

0H

10H

000000

000000

000000

01010

000000

010000

0H

0H

0H

BH

0H

12H

000000

000000

000000

01011

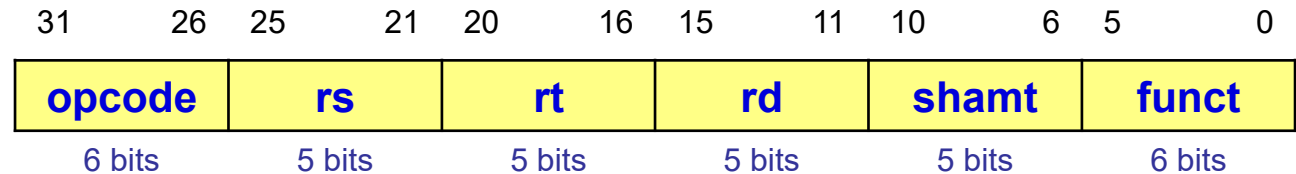
000000

010010

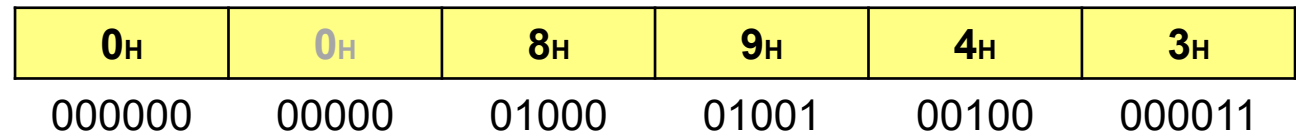
Operações Aritméticas (parte 2)

► Shift Right Arithmetic (sra)

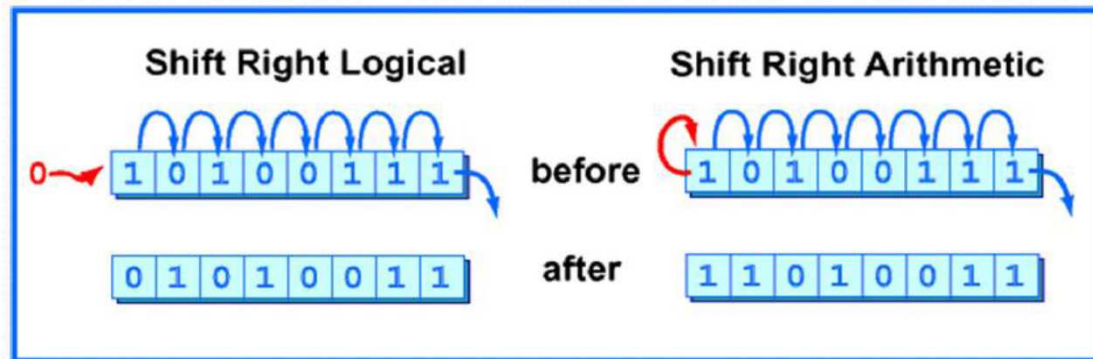
Tipo R



sra \$t1, \$t0, 4



Tem função similar ao `srl`, mas mantém o bit mais significativo (isto é, mantém o sinal em caso de Complemento de Dois)



Usado para divisões por 2^N
onde N é o número de deslocamentos