	breviação	Instrução	Tipo	Opcode	Funct	Descrição	Funcionamento
1	add	Soma	R	000000	000000	Soma o valor de dois registradores	[op r1 r2 rf xx funct]> rf = r1 + r
2	sub	Subtração	R	000000	000001	Subtrai o valor de dois registradores	[op r1 r2 rf xx funct]> rf = r1 - r
3	mult	Multiplicação	R	000000	000010	Multiplica o valor de dois registradores	[op r1 r2 rf xx funct]> rf = r1 * r
4	div	Divisão	R	000000	000011	Divide o valor de dois registradores	[op r1 r2 rf xx funct]> rf = r1 / r
5	and	And	R	000000	100000	Faz a operação and entre dois registradores	[op r1 r2 rf xx funct]> rf = r1 and
6	or	Nand	R	000000	100001	Faz a operação or entre dois registradores	[op r1 r2 rf xx funct]> rf = r1 or :
7	nand	NOr	R	000000	100010	Faz a operação nand entre dois registradores	[op r1 r2 rf xx funct]> rf = r1 mand
8	nor	Or	R	000000	100011	Faz a operação nor entre dois registradores	[op r1 r2 rf xx funct]> rf = r1 mor
9	sie	Set Less or Equal Than	R	000000	110000	Compara se o primeiro registrador é menor ou igual que o segundo	[op r1 r2 rf xx]> rf = 1 if r1 <= r
10	10	Set Less Than	R	000000	110001	Compara se o valor do primeiro registrador é menor que o segundo	[op r1 r2 rf xx]> rf = 1 if r1 < r
11	addi	Soma Imediato	1	0.00001		Soma o valor de um registrador e um imediato	[op rl rf inm]> rf = rl + inm
12	subi	Subtração Imediato	1	000010		Subtrai o valor de um reg. e um imediato	[op r1 rf inn]> rf = r1 - inn
13	divi	Divisão Imediato	1	000010		Divide o valor de um registrador e um imediato	[op r1 rf imm]> rf = r1 / imm
14	nori	NOr Imediato	1	001001		Faz a operação nor entre um registrador e um imediato	[op r1 rf imm]> rf = r1 / imm [op r1 rf imm]> rf = r1 nor imm
14	non	NUr Imediato Store Word	1	001001		Faz a operação nor entre um registrador e um imediato  Guarda um dado em um registrador	[op rl rf imm]> rf = rl nor imm [op xx rf addr]> ram[addr] = rf
16	lw	Load Word	- 1	000111		Carrega um dado de um registrador	[op xx rf addr]> rf = ram[addr]
17	multi	Multiplicação Imed.	1	000100		Multiplica o valor de um registrador e um imediato	[op r1 rf inm]> rf = r1 * inm
18	ori	Or Imediato	1	001010		Faz a operação or entre um registrador e um imediato	[op rl rf imm]> rf = rl or imm
19	andi	And Imediato	- 1	001011		Faz a operação and entre um registrador e um imediato	[op r1 rf imm]> rf = r1 and imm
20	50	Branch less than	- 1	010000		Muda a instução se o valor for menor que o esperado	[op r1 r2 addr]> if r1 < r2 then branch
21	sie/	Set Less or Equal Than Imediato	- 1	011100		Compara se o valor do registrador é menor ou igual que o imediato	[op r1 rf inm]> rf = 1 if r1 < imm
22	siti	Set Less Than Imediato	1	011110		Compara se o valor do registrador é menor que o imediato	[op r1 rf imm]> rf = 1 if r1 <= im
23	bgt	Branch greater than		100000		Muda a instrução se o valor for maior que o esperado	[op r1 r2 addr]> if r1 < r2 them branch
24	out	Salda	OUT	100001		Manda o dado para a saida	[op xx rf xx]> 007 = rf
25	in	Entrada	IN.	101000		Lê o dado da entrada	[op xx rf xx]> rf = IN
26	hen	Branch on Equal		110000		Muda a instrução se o valor de registrador for o esperado	[op r1 r2 addr]> if r1 = r2 then branch
27	hoe	Branch on not Equal		111000		Muda a instrução se o valor de registrador não for o esperado	[op r1 r2 addr]> if r1 != r2 then branc
28		Jump	- 1	111110		Nuda a instrução se o valor de registrador não tor o esperado Pula para o endereco calculado	[op r1 r2 addr]> 12 r1  = r2 then branc
	jmp						
29	he	Halt	- 1	111111		Paraliza o processador	[op xx]> para o processamento
30	jal	Jump and Link	J	111101		Pula para o endereço calculado e salva o endereço atual na pilha	[op addr]> PC = addr amd push stac
31	jst	Jump Stack	J	111100		Pula para o endereço no topo da pilha	[op xx]> PC = pop stack
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
	- 1					1	1
44							
45							
46							
47							
48							
49							
50							
51							
52							
53							
54							
55							
56							
57 58	- 1						1
59						1	
60							
61							
62							
63							
64			1				

Tipo			Operando	s (32 bits)											
R	ор	rl	12	rf	shamt	funct									
bits	6	5	5	5	5	6									
7	ор	rl	rf		imediato										
bits	6	5	5 5 16												
OUT	ор	don't care	don't care if don't care												
bits	6	5	5		16										
IN	ор	don't care	rf		don't care										
bits	6	5	5		16										
J	ор			endereço											
4.74				0.0											

	os (32 bits)			Número	Nome	Valor	Descrição
12	rf	shamt	funct	0	Sr0	00000	Propósito geral
5	5	5	6	1	Srt	00001	Propósito geral
ıf		imediato		2	Sr2	00010	Propósito geral
5		16		3	Sr3	00011	Propósito geral
ıf		don't care		4	Sr4	00100	Propósito geral
5		16		5	Sr5	00101	Propósito geral
ıf		don't care		6	Sr6	00110	Propósito geral
5		16		7	Sr7	00111	Propósito geral
	endereço			8	Sr8	01000	Propósito geral
	26			9	Sr9	01001	Propósito geral
				10	Sr10	01010	Propósito geral
				11	Sr11	01011	Propósito geral
				12	Sr12	01100	Propósito geral
				13	Sr13	01101	Propósito geral
				14	Sr14	01110	Propósito geral
				15	\$r15	01111	Propósito geral
				16	Sr16	10000	Propósito geral
				17	Sr17	10001	Propósito geral
				18	\$r18	10010	Propósito geral
				19	Sr19	10011	Propósito geral
				20	Sr20	10100	Propósito geral
				21	Sr21	10101	Propósito geral
				22	\$122	10110	Propósito geral
				23	\$123	10111	Propósito geral
				24	Sr24	11000	Propósito geral
				25	\$r25	11001	Propósito geral
				26	Sr26	11010	Propósito geral
				27	Sr27	11011	Propósito geral
				28	Sr28	11100	Propósito geral
				29	Sr29	11101	Propósito geral
				30	\$r30	11110	Retorno
				31	Srq	11111	Quociente

Instr.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	opcode						r1 r2						rf					shamt					funct									
- 1	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0			0	0	0	0	0	0		0	0	0		0	0
	г		ори	oode			rl				rf					imediato																
OUT	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	opcode						don't care					rf					Г	don't care														
IN	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	opcode							don't care				rf											don't care									
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	oncode							oncode endereco																								