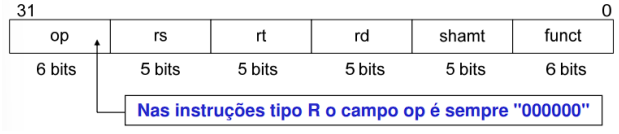
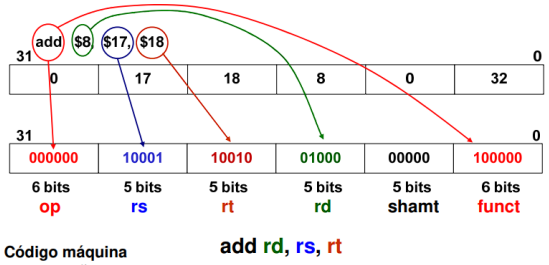
Codificação de instruções no MIPS – formato R

- O formato R é um dos três formatos de codificação de instruções no MIPS
- Campos da instrução:
  - op: opcode (é sempre zero nas instruções tipo R)
  - rs: Endereço do registo que contém o 1º operando fonte
  - rt: Endereço do registo que contém o 2º operando fonte
  - rd: Endereço do registo onde o resultado vai ser armazenado
  - shamt: shift amount (útil apenas em instruções de deslocamento)
  - funct: código da operação a realizar

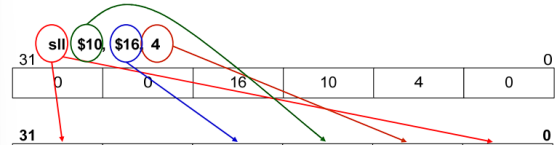




Código máquina da instrução: **add rd, rs, rt**

$000000\ 10001\ 10010\ 01000\ 00000\ 100000_2$

$0000\ 0010\ 0011\ 0010\ 0100\ 0000\ 0010\ 0000_2 = 0x02324020$

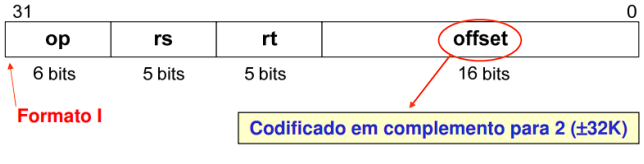


sll rd, rt, shamt

Código máquina da instrução: $00000000000100000101000100000000_2 = 0x00105100$

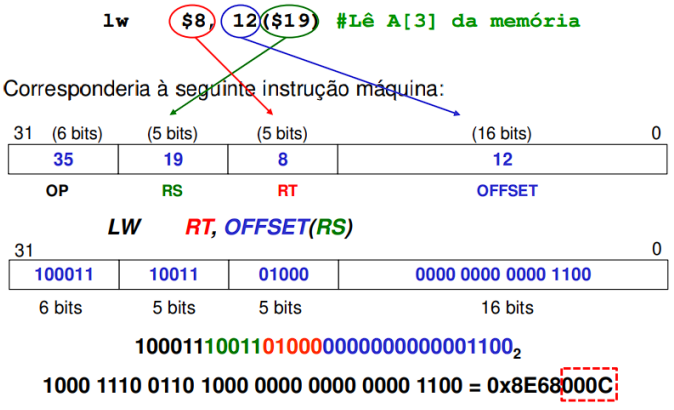
Codificação das instruções de acesso à memória no MIPS

A necessidade de codificação de uma constante de 16 bits, obriga à definição de um novo formato de codificação, o formato I

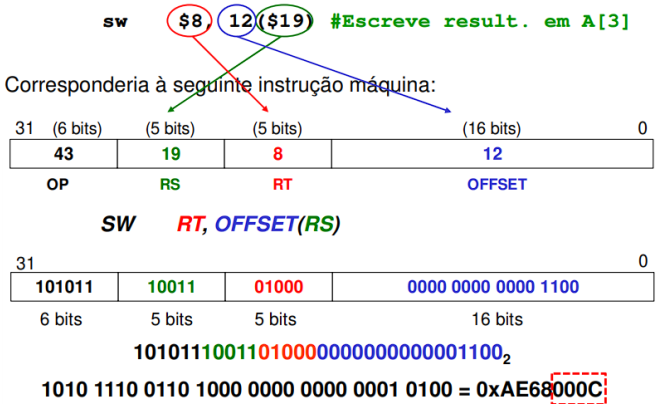


- Gama de representação da constante de 16 bits
  - [-32768, +32767]

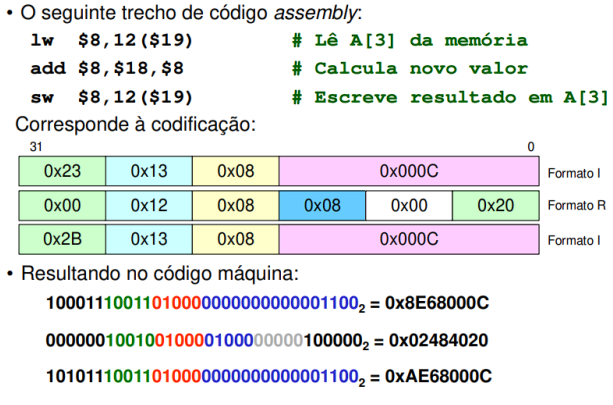
Codificação da instrução LW (Load Word)



Codificação da instrução SW (Store Word)



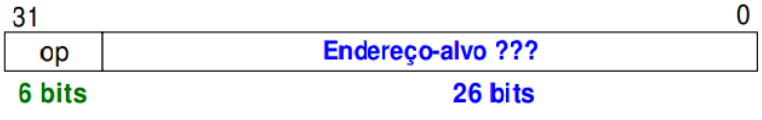
Exemplo de codificação



Codificação da instrução de salto incondicional

- No caso da instrução de salto incondicional (" j "), é usado endereçamento pseudo-direto, i.e. o código máquina da instrução codifica diretamente parte do endereço alvo

  • Formato J:



  • Endereço alvo da instrução "j" é sempre múltiplo de 4 (2 bits menos significativos são sempre 0)

• **Exemplo:** `j Label` **# com Label = 0x001D14C8**

0x001D14C8: 0000 0000 0001 1101 0001 0100 1100 1000

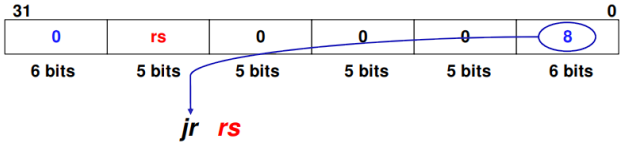
(26 bits) 00 0000 0111 0100 0101 0011 0010

• Código máquina (opcode do "j" é 0x02):

0000 1000 0000 0111 0100 0101 0011 0010 = 0x08074532

Instrução JR (jump on register)

O formato de codificação da instrução JR é o formato R:



Manipulação de constantes no MIPS

- As instruções aritméticas e lógicas que manipulam constantes (do tipo imediato) são identificadas pelo sufixo "i":

```
addi $3,$5,4          # $3 = $5 + 0x0004
andi $17,$18,0x3AF5   # $17 = $18 & 0x3AF5
ori  $12,$10,0x0FA2   # $12 = $10 | 0x0FA2
slti $2,$12,16        # $2 = 1 se $12 < 16
                      # ($2 = 0 se $12 ≥ 16)
```

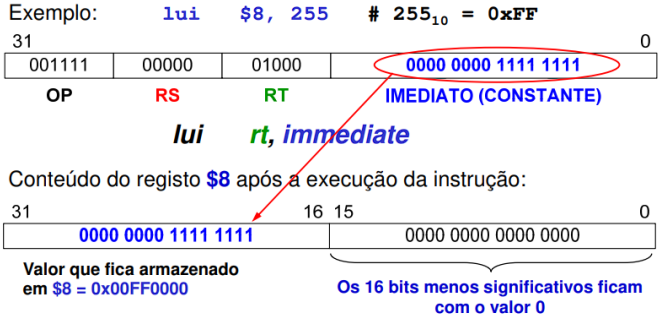
Estas instruções são codificados usando o formato I. Logo apenas 16 bits podem ser usados para codificar a constante.

Manipulação de constantes de 32 bits – LUI

- Para permitir a manipulação de constantes com mais de 16 bits, o ISA do MIPS inclui a seguinte instrução, também codificada com o formato I:

    lui $reg, immediate

- A instrução lui ("Load Upper Immediate"), coloca a constante "immediate" nos 16 bits mais significativos do registo destino ($reg)
- Os 16 bits menos significativos ficam com 0x000



Instruções JAL e JALR

- A instrução "jal" é codificada do mesmo modo que a instrução " j": formato j em que os 26 bits menos significativos são obtidos dos 28 bits menos significativos do endereço-alvo, deslocados à direita dois bits