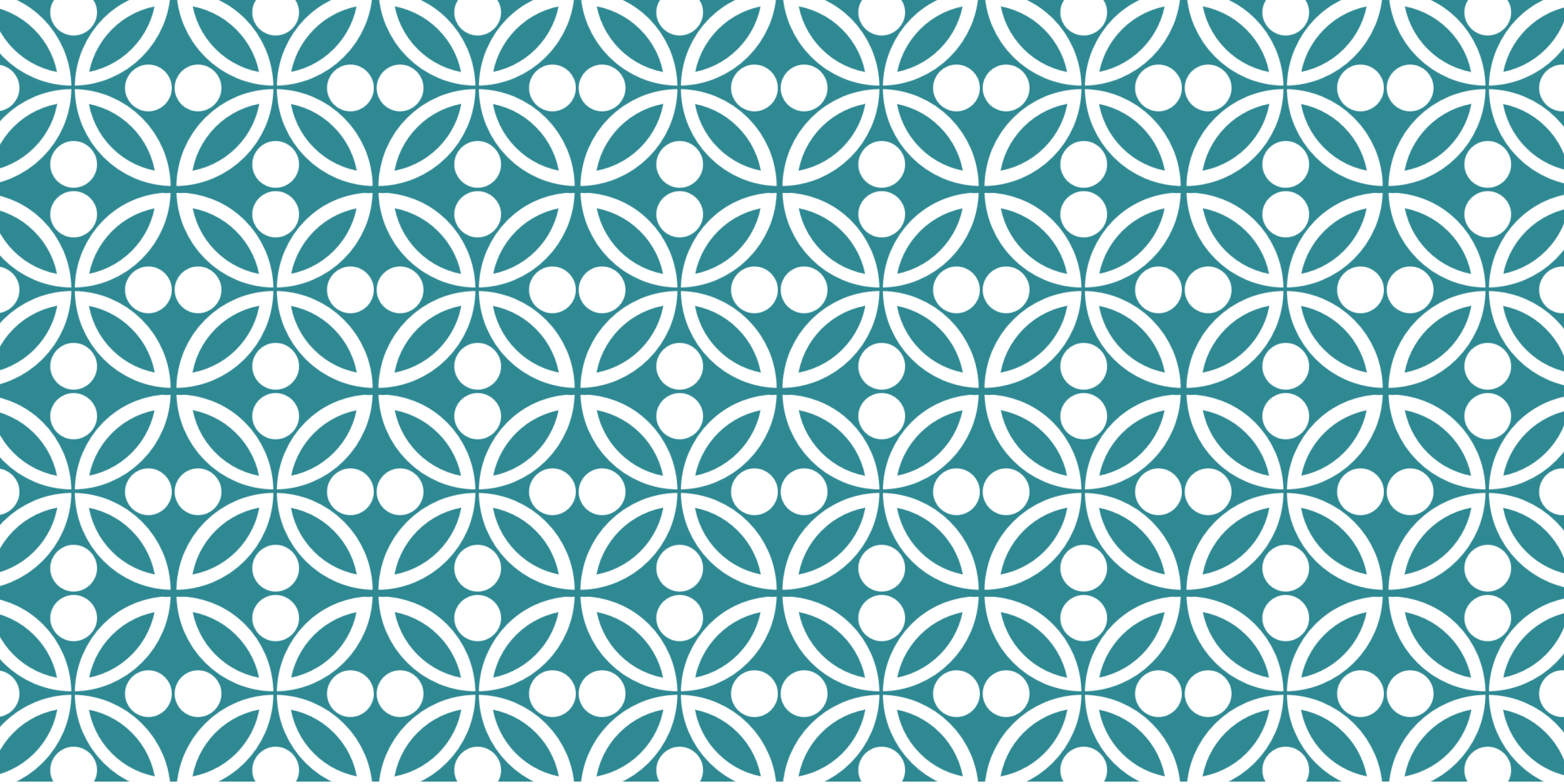


PROJETOS DIGITAIS E MICROPROCESSADORES
MIPS: FORMATO DAS INSTRUÇÕES

Marco A. Zanata Alves

FORMATOS DAS INSTRUÇÕES

Instruções têm 32 bits

Todas têm opcode de 6 bits

- Indicando o que a instrução faz
- Formato da instrução é codificado no opcode.
- Modo de endereçamento é codificado juntamente no opcode

Apenas três formatos diferentes

- R – Register
- I – Immediate
- J – Jump

FORMATOS DAS INSTRUÇÕES – TIPO R

op-code: operação básica da instrução (usará ULA)

rs: 1o registrador de origem

rt: 2o registrador de origem

rd: registrador de destino

shamt: shift amount

funct: function code, indica o que a ULA irá fazer

FORMATOS DAS INSTRUÇÕES – TIPO R

> Aritméticas e lógicas

> Movimentação entre registradores

Exemplo:

- ADD R[rd] = R[rs] + R[rt] $0/20_{hex}$
- $8 = $9 + $10
- 000000 | 01010 | 01001 | 01000 | 00000 | 100000

FORMATOS DAS INSTRUÇÕES – TIPO I

> Operações aritméticas e lógicas com operando imediato

> Loads e Stores

> Desvios condicionais (“branches”)

> Desvios incondicionais para endereço em registrador

- O campo rt muda de significado (pode ser o destino)

FORMATOS DAS INSTRUÇÕES – TIPO I

Operações aritméticas e lógicas com operando imediato

Exemplo:

- ADDI R[rt] = R[rs] + SignExtImm 8_{hex}
- $8 = $9 + 32
- 001000 | 01001 | 01000 | 0000 0000 0001 0010

FORMATOS DAS INSTRUÇÕES – TIPO I

Load

- End. Memória = rs + Immediate

Exemplo:

- LW R[rt] = M[rs + SignExtImm] 23_{hex}
- $8 = [$9 + 32]
- 100011 | 01001 | 01000 | 0000 0000 0001 0010

FORMATOS DAS INSTRUÇÕES – TIPO I

Load/Store

- End. Memória = rs + Immediate

Exemplo:

- SW M[rs + SignExtImm] = R[rt] $2b_{hex}$
- [$9 + 32] = $8
- 101011 | 01001 | 01000 | 0000 0000 0001 0010

FORMATOS DAS INSTRUÇÕES – TIPO I

Jumps

- Destino = (PC + 4) + (4 * deslocamento)

BranchAddr = Imm. * 4

Exemplo:

- BEQ if(R[rs] == R[rt]) PC = PC + 4 + BranchAddr 4_{hex}
- if($8 == $9) PC = PC + 4 + (4 * 32)
- 000100 | 01000 | 01001 | 0000 0000 0001 0010

FORMATOS DAS INSTRUÇÕES – TIPO J

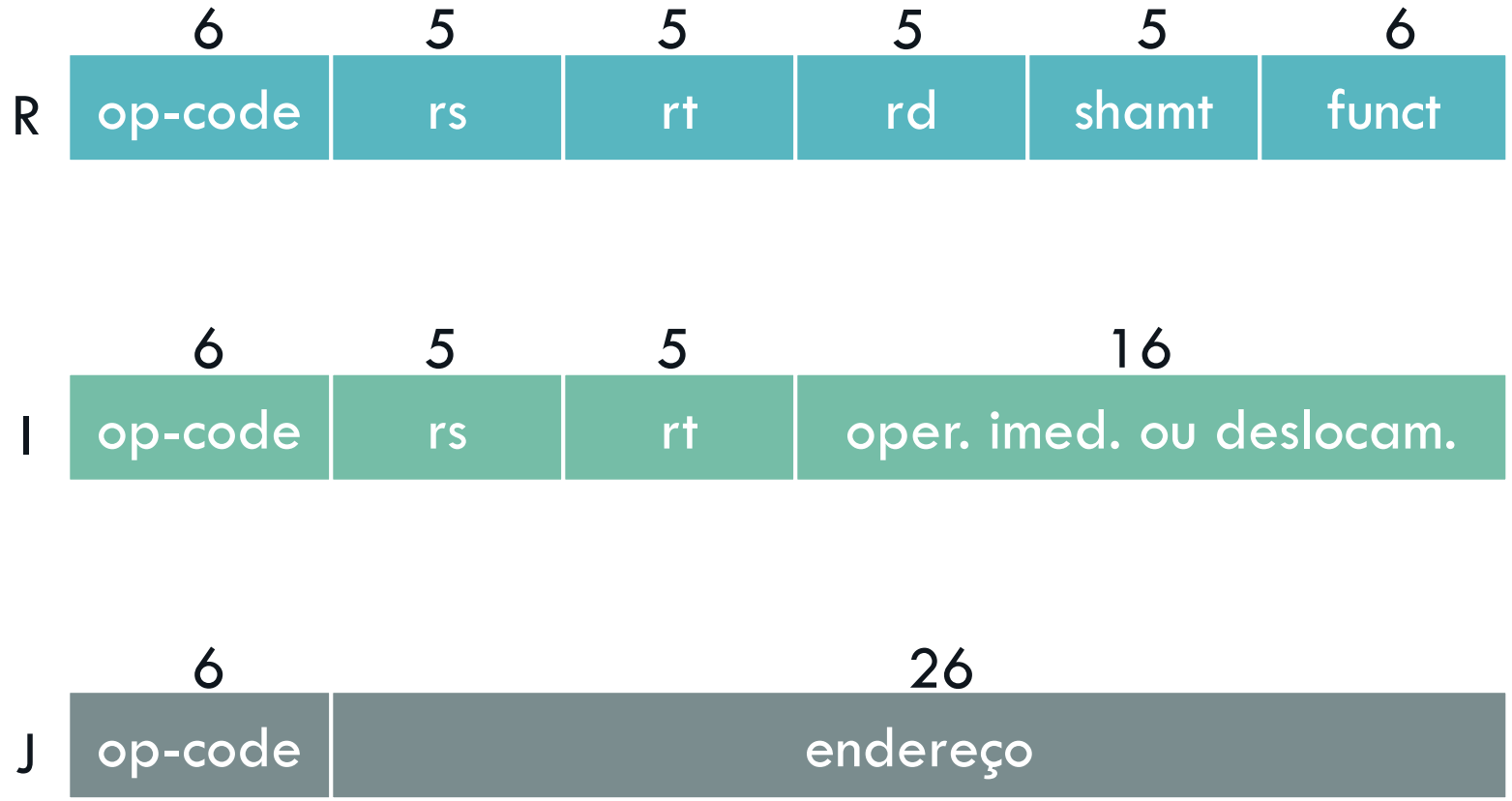
Desvios com endereçamento absoluto (ainda usa 4 bits do PC)

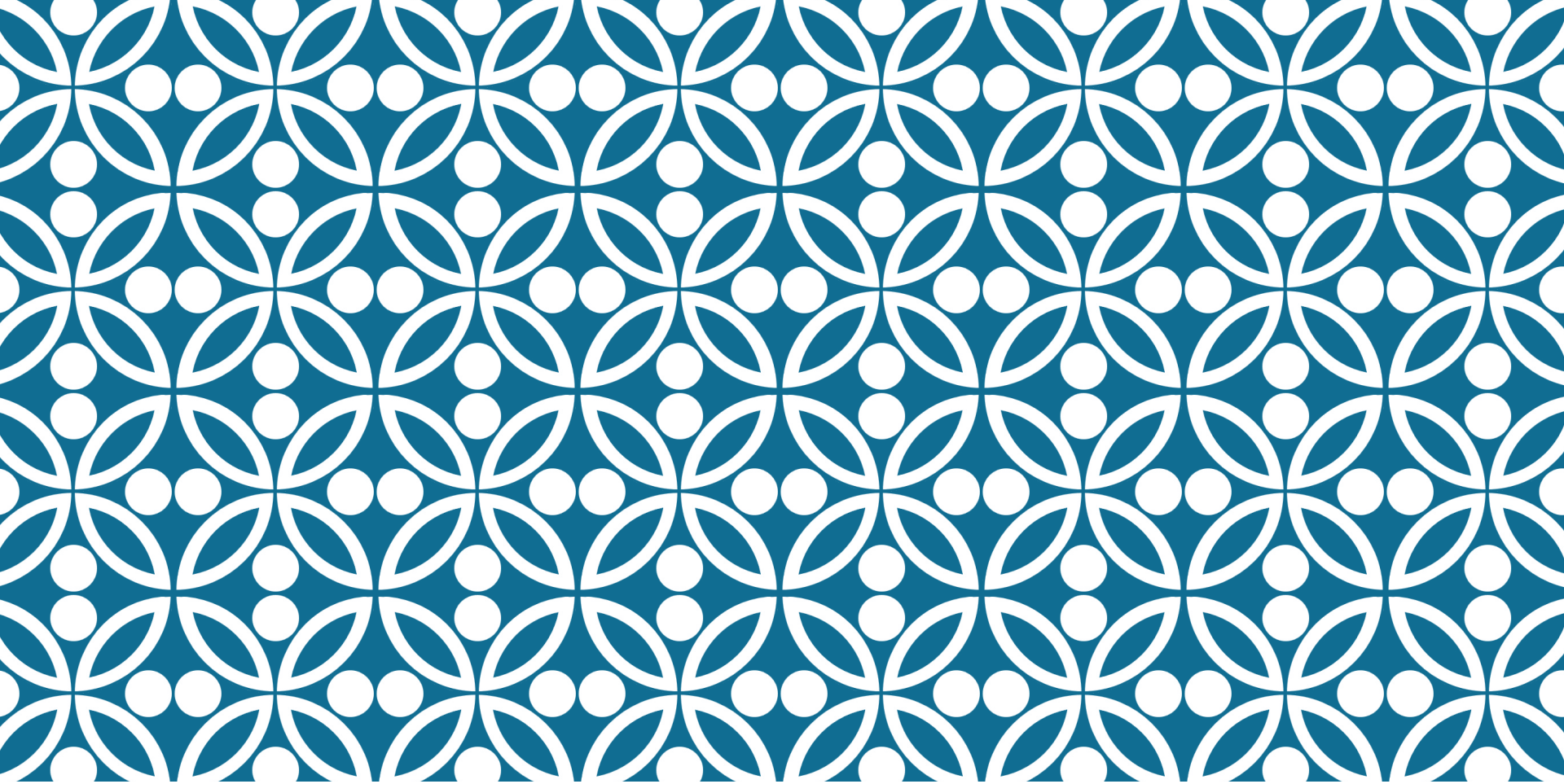
Chamada de sub-rotinas

Exemplo:

- J goto address CONST 2_{hex}
- PC = 1060
- 000010 | 00 0000 0000 0000 0100 0010 0100

TRÊS FORMATOS





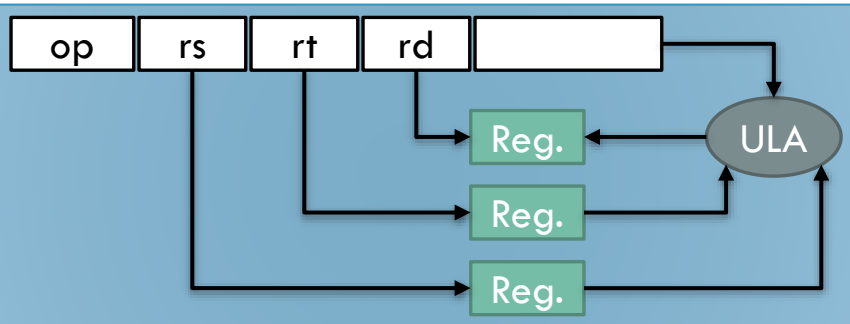
MIPS: MODOS DE ENDEREÇAMENTO

MODOS DE ENDEREÇAMENTO

Modo registrador

- Para instruções aritméticas e lógicas: dado está em registrador
- Ex: add $2 $3 $4 ;; $4 = $2 + $3

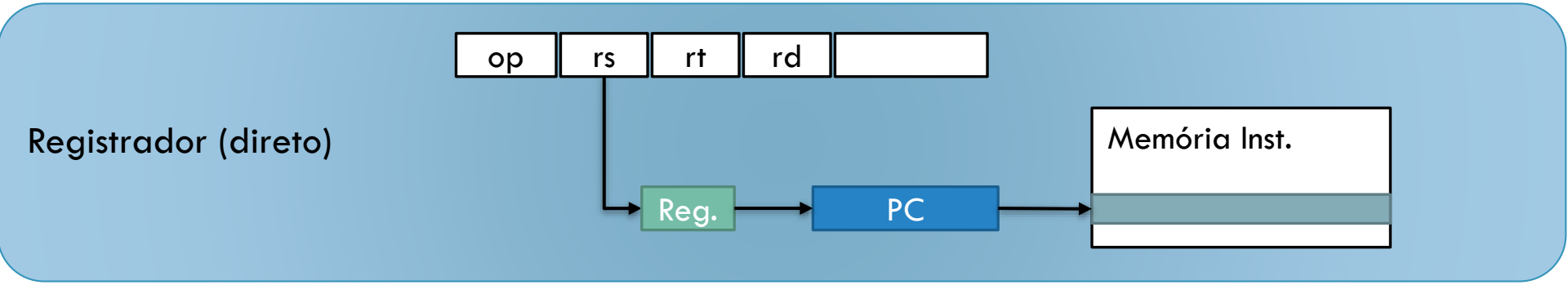
Registrador (direto)



MODOS DE ENDEREÇAMENTO

Modo registrador

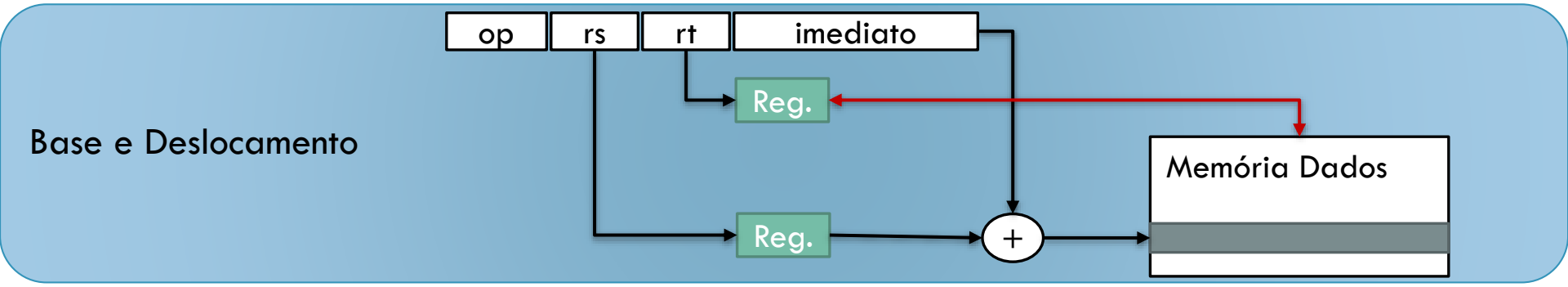
- Para instruções de salto incondicional: endereço está em registrador
- Ex: jr $31 ;; PC = $31



MODOS DE ENDEREÇAMENTO

Modo base e deslocamento

- Para instruções Leitura ou Escrita
- Base é registrador inteiro de 32 bits
- Deslocamento de 16 bits contido na própria instrução
- Ex: lw $2 $3 16 ;; $3 = M($2 + 16)

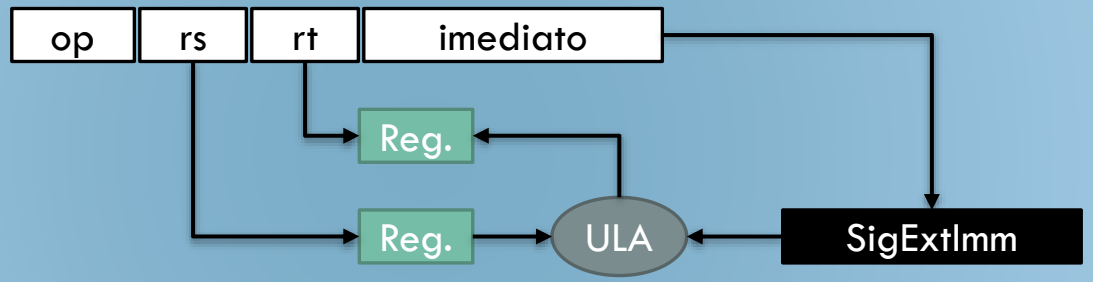


MODOS DE ENDEREÇAMENTO

Modo imediato

- Para instruções aritméticas e lógicas
- Dado imediato de 16 bits contido na própria instrução
- Dado é estendido para 32 bits
  - Extensão com sinal nas instruções aritméticas com sinal
  - Ex: addi $2 $3 -128 ;; $3 = $2 + (-128)
  - Extensão sem sinal nas instruções aritméticas sem sinal e lógicas
  - Ex: ori $2 $3 4 ;; $3 = $2 or (0...000100)

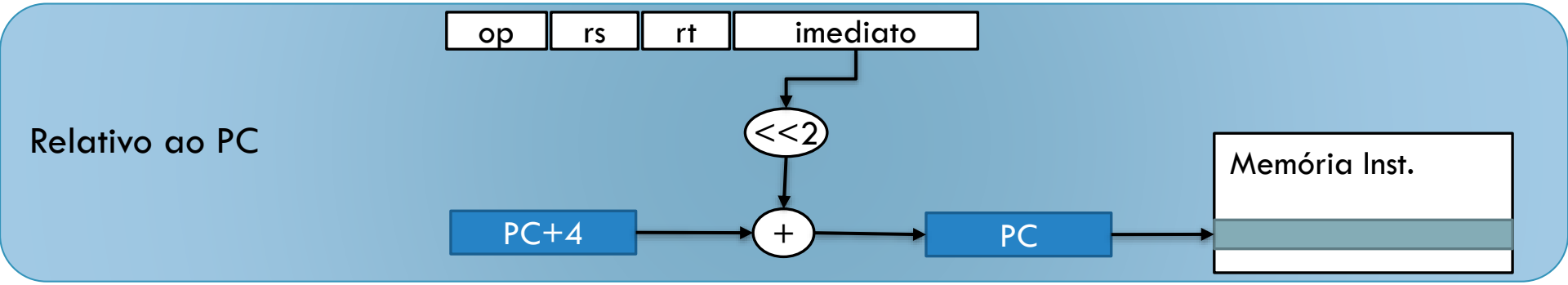
Imediato



MODOS DE ENDEREÇAMENTO

Modo relativo ao PC

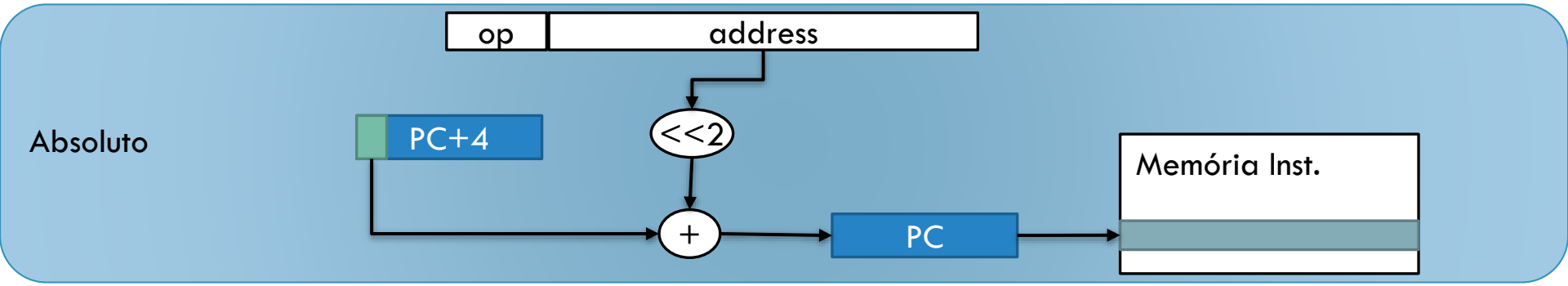
- Para instruções de desvio condicional
- Endereço é a soma do PC com deslocamento contido na instrução
- Deslocamento é dado em palavras e precisa ser multiplicado por 4
- Ex: bne $2 $3 16 ;; if ($2 != $3)
  then PC = (PC +4) + (16*4)
  else PC=PC+4



MODOS DE ENDEREÇAMENTO

Modo absoluto

- Para instruções de salto incondicional
- Instrução tem campo com endereço de palavra com 26 bits
- Endereço obtido com shift de dois bits para esquerda (total 28 bits)
- 4 bits mais significativos obtidos do PC
- Só permite desvios dentro de uma área de 256 Mbytes?!
- Ex: J 1024 ;; PC = PC[31:28] , 1024 , 00



EXERCÍCIOS (I)

Para uma ISA de apenas 32 instruções, quantos bits de opcode são necessários?

Caso metade dessas instruções forem do tipo R, qual o impacto no projeto?

EXERCÍCIOS (II)

Considerando um projeto agressivo, com 128 registradores internos no processador, qual o impacto no tamanho das instruções?

EXERCÍCIOS (III)

Sabendo que o código binário para a instrução addi é igual a 8(hexa), como ficará a tradução da seguinte instrução…

```
addi $t0 $v0 11
```

EXERCÍCIOS (III)

Sabendo que o código binário para a instrução addi é igual a 8(hexa), como ficará a tradução da seguinte instrução...

```
addi $t0 $a0 11
```

```
$t0 = $8 // Temporário
$a0 = $4 // Argumento
```

Tipo R | Opcode | RS | RT | RD | ShiftAmount | Funct |

Tipo I | Opcode | RS | RT | Imediato |

Tipo J | Opcode | Endereço |