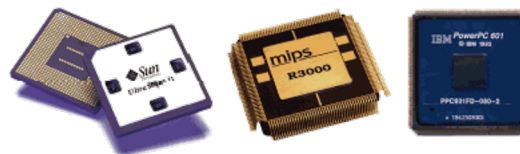


# Tema 3: Microprocesador MIPS (Seminario)

Fundamentos de Ordenadores  
y Sistemas Operativos

Mario Martínez Zarzuela  
marmar@tel.uva.es



# Piensa en el valor necesario para las señales de la Unidad de Control

N

- Pensar en **valor necesario en líneas de control (0/1)**, para las siguientes instrucciones
  - Instrucciones tipo-R
    - Ej. Instrucción **add rd, rs, rt**
  - Instrucción de carga/almacenamiento tipo-I
    - Ej. Instrucción **lw rt, imm(rs)**
  - Instrucción de salto condicional tipo-I
    - Ej. Instrucción **beq rt, rs, imm**
  - Instrucción de salto no condicional tipo-J
    - Ej. Instrucción **j addr**

## V



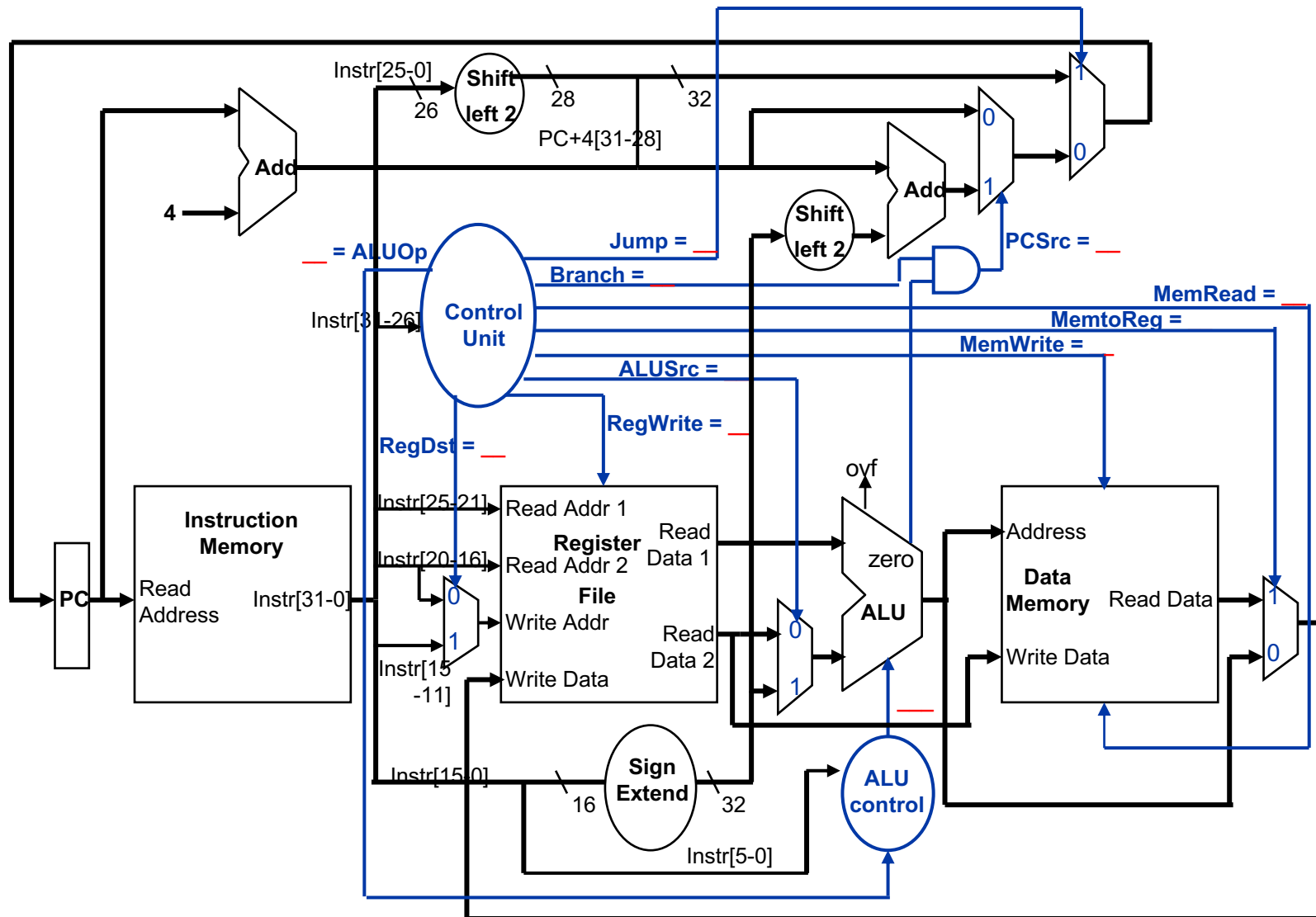
# Valores ALU Op y ALU Control

- Señales ALU Op y ALU Control según instrucción

Instrucc.	ALU Op	ALU control	Función ALU
lw sw	00	010	suma
beq bne	01	110	resta
add	10	010	suma
sub	10	110	resta
and	10	000	multiplicación lógica
or	10	001	suma lógica
slt	10	111	menor que

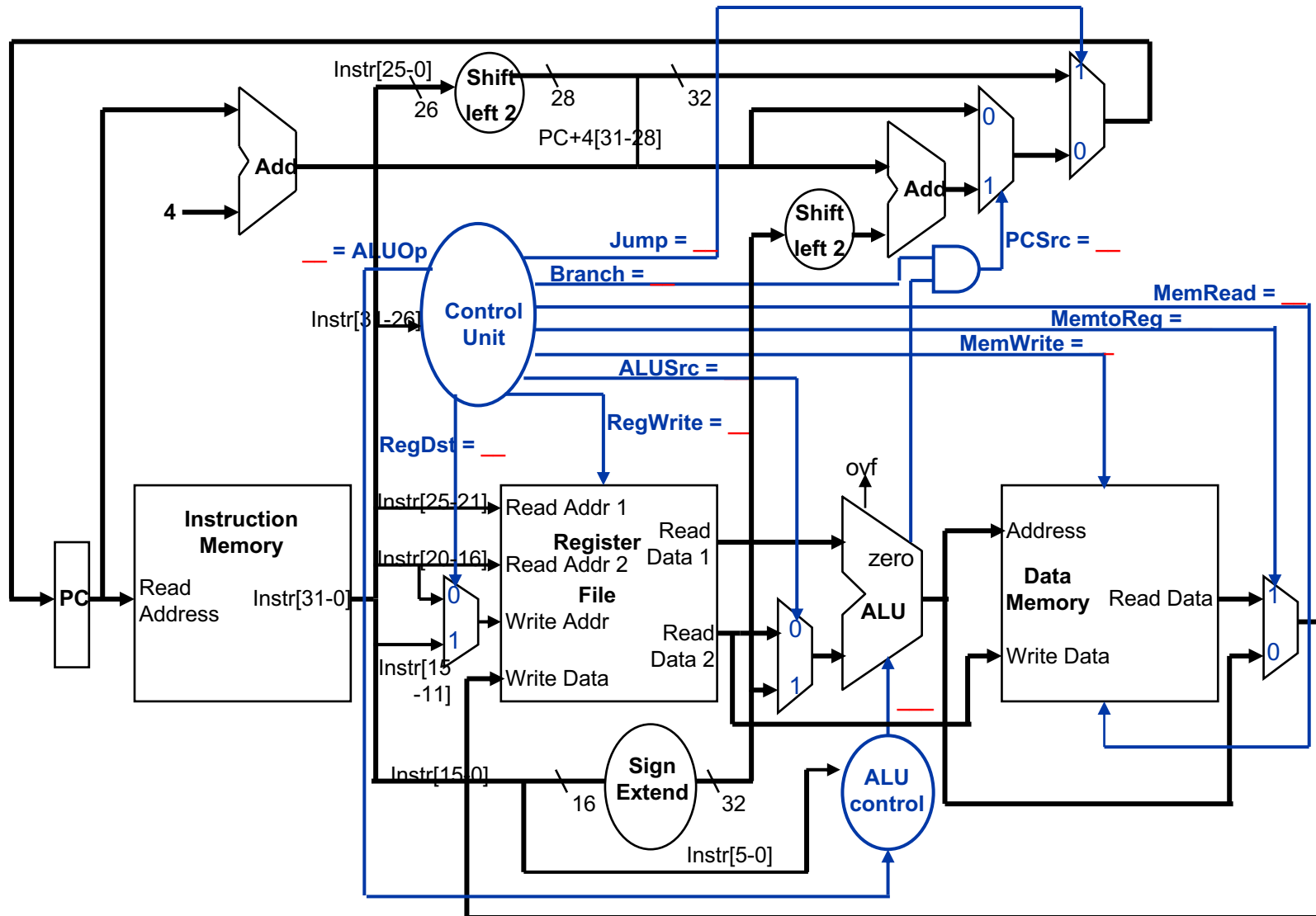
# Unidad de control: lw rt, imm(rs)

N



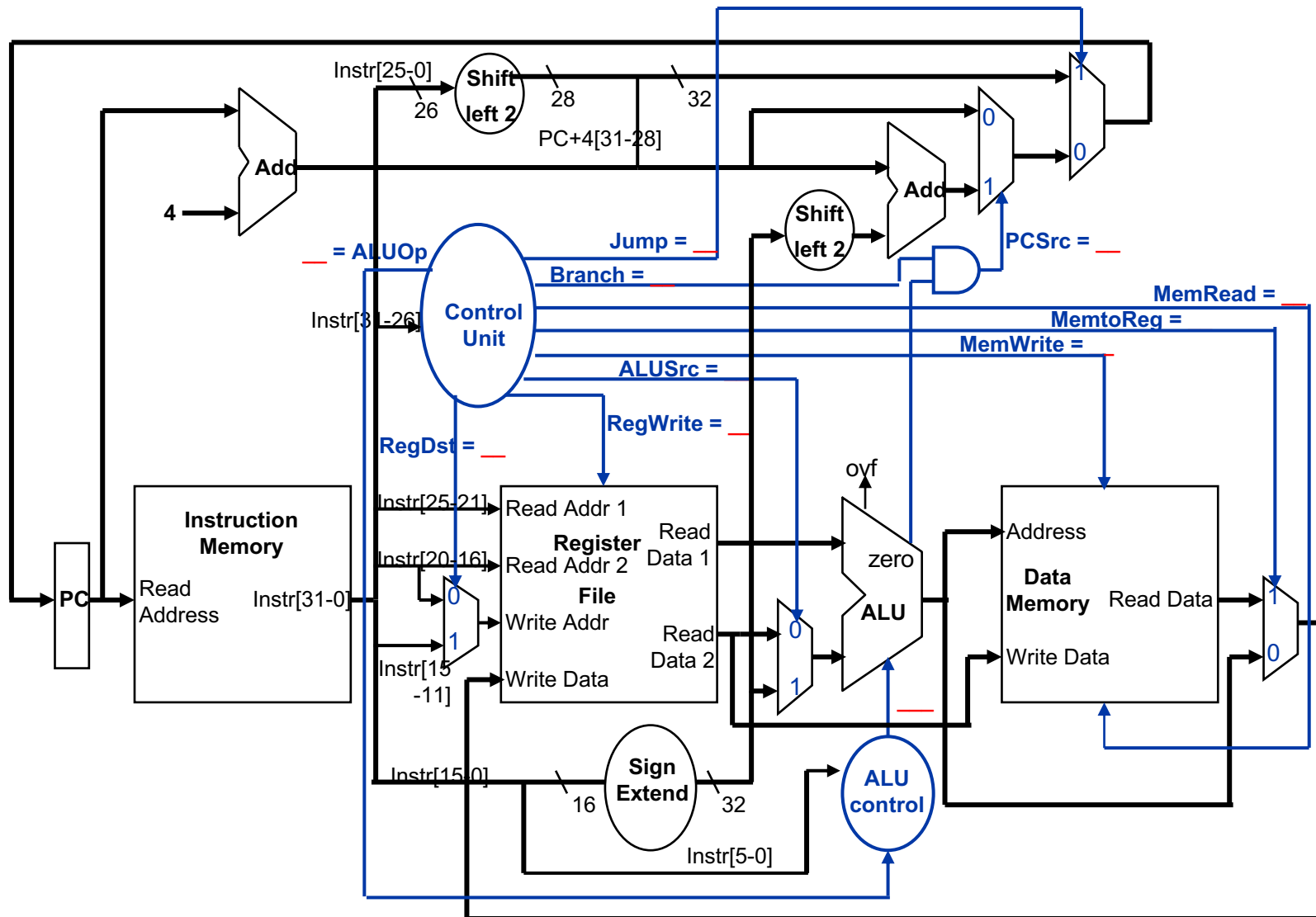
# Unidad de control: beq rt, rs, imm

N



# Unidad de control: j addr

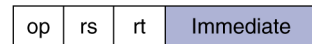
N



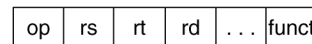
# Modos de direccionamiento: ¿cómo se accede al dato/instrucción?

R

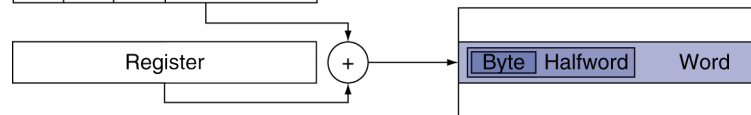
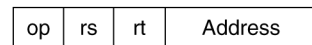
## 1. Immediate addressing



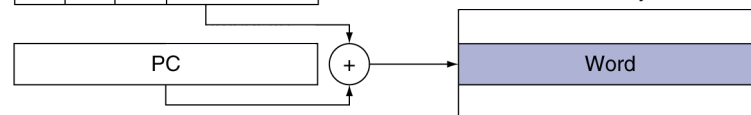
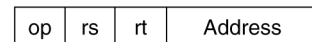
## 2. Register addressing



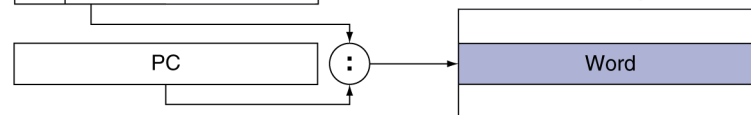
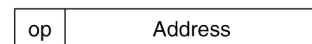
## 3. Base addressing



## 4. PC-relative addressing



## 5. Pseudodirect addressing



- Aritméticas tipo-I
  - addi
- Aritméticas tipo-R
  - add
- Salto relativo a dirección base en registro tipo-I
  - lw
- Salto relativo a PC tipo-I
  - beq
- Salto pseudodirecto tipo-J
  - jal



# Camino de Datos y Unidad de Control

R

- Ejercicio tipo (Examen)
  - Para el programa en MIPS de la siguiente transparencia y teniendo en cuenta que:
    - La memoria de datos (.data) comienza en la posición de memoria  $20000_{10} = 0x4E20$
    - La memoria de instrucciones (.text) comienza en la posición de memoria  $10000_{10} = 0x2710$
  - **A)** Indicar al lado de cada instrucción los códigos correspondientes usando base decimal
  - **B)** Traduce los códigos anteriores a base binaria y hexadecimal
  - **C)** Escribe en un esquema del Camino de Datos y la Unidad de Control todos los valores necesarios en el para ejecutar cada instrucción

# Camino de Datos y Unidad de Control

R

```

                .data
_____ 0x_____ num1: .word    13
                num2: .word    12
                res:  .word    0

                .text
_____ 0x_____  lw      $s1,num1($zero)
                lw      $s2,num2($zero)
                beq     $s1,$s2,salto
                sub     $s0,$s1,$s2
                j       fin
                salto:add $s0,$s1,$s2
                fin:  sw   $s0,res($zero)
    
```

*op rs rt rd sh f*

*op rs rt imm/addr*

*op addr*


# Camino de Datos y Unidad de Control

N

```

                                .data
20000  0x4E20  num1: .word    13
                                num2: .word    12
                                res:  .word    0
                                .text
10000  0x2710  lw          $s1,num1($zero)
                                lw          $s2,num2($zero)
                                beq         $s1,$s2,salto
                                sub         $s0,$s1,$s2
                                j           fin
                                salto:add   $s0,$s1,$s2
                                fin:  sw    $s0,res($zero)
    
```

*op rs rt rd sh f*

*op rs rt imm/addr*

*op addr*


# Convierte a código máquina

R

sub \$s0, \$s1, \$s2

tipo-R	op(6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)
Decim						
Bin						
Hex						

lw \$s1, num1(\$zero)

tipo-I	op (6)	rs (5)	rt (5)	imm/offset (16)
Decim				
Bin				
Hex				

j fin

tipo-J	op (6)	address (26)
Decim		
Bin		
Hex		

# Convierte a código máquina

N

sub \$s0, \$s1, \$s2

tipo-R	op(6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)
Decim	0		18			
Bin		10 001		1000 0	000 00	
Hex	0x02328022					

lw \$s1, num1(\$zero)

tipo-I	op (6)	rs (5)	rt (5)	imm/offset (16)
Decim				20000
Bin				0100 1110 0010 0000
Hex				

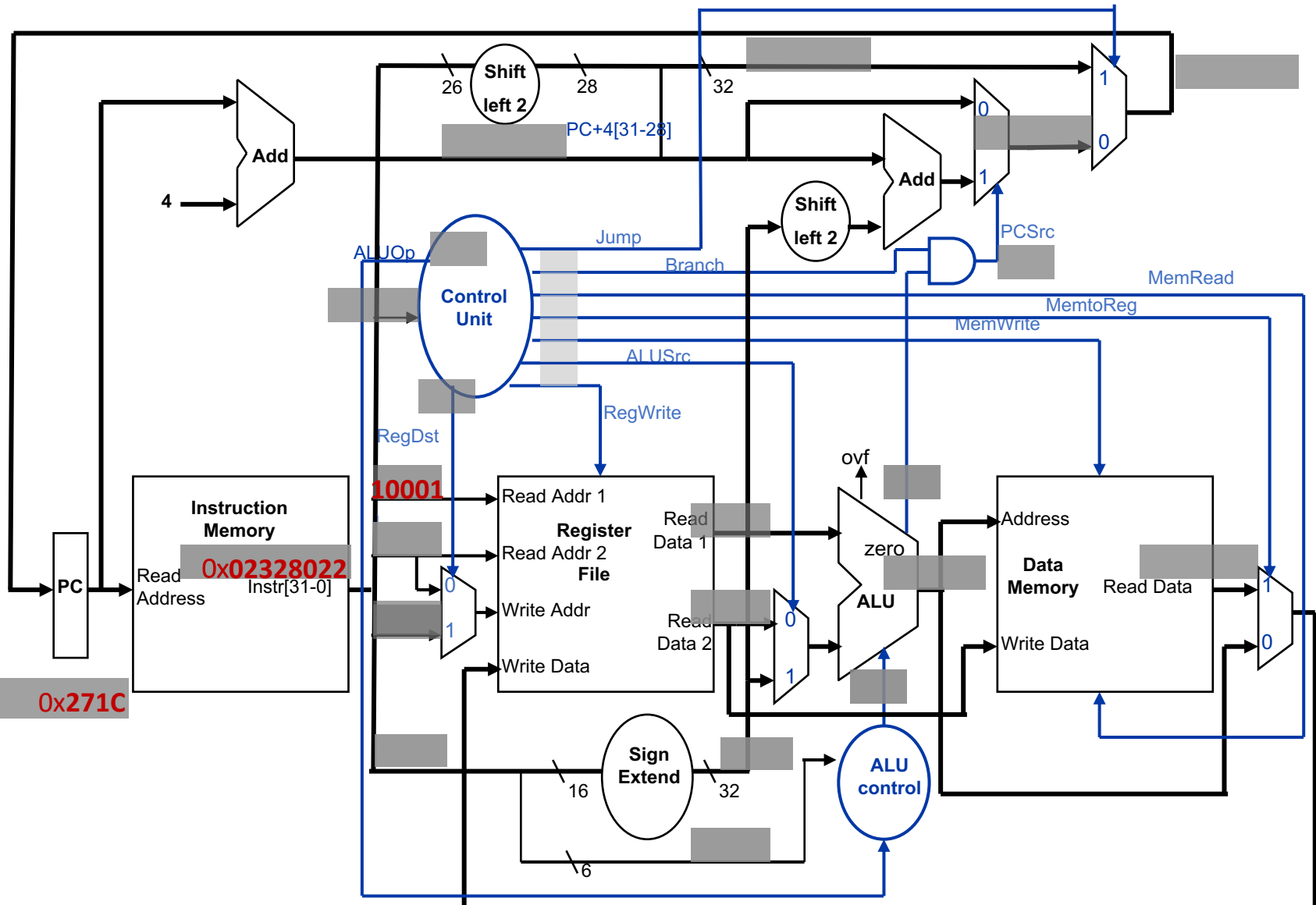
j fin

tipo-J	op (6)	address (26)
Decim		
Bin		00 0000 0000 0000 1001 1100 1010
Hex		

# Camino de Datos y Unidad de Control

## sub \$s0, \$s1, \$s2

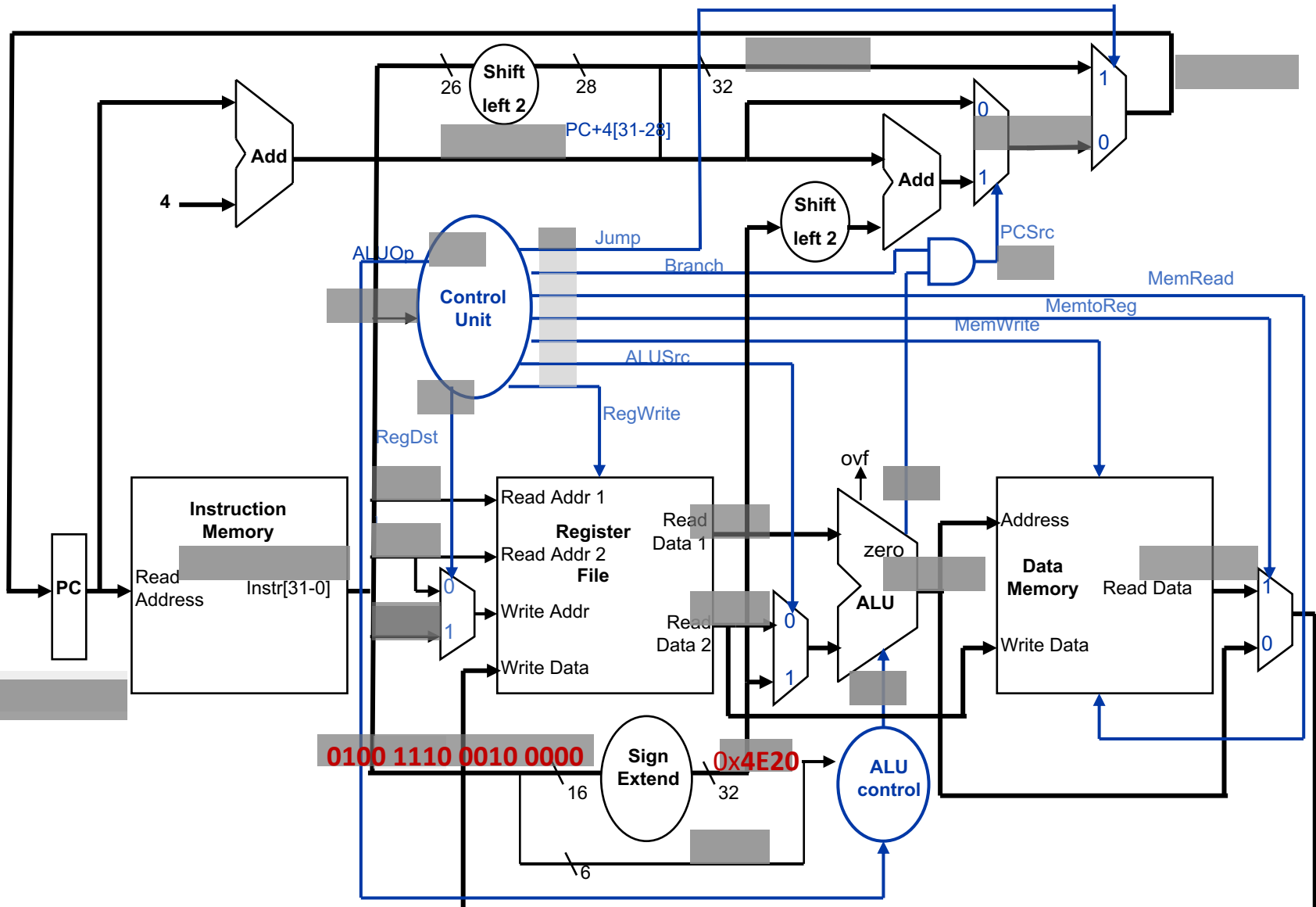
N



# Camino de Datos y Unidad de Control

## lw \$s1, num1(\$zero)

N



# N

