Temario

(siempre en MIPS)

- Formato de instrucciones máquina
- Instrucciones más comunes
- Camino de datos sencillo (microarquitectura u organización de un procesador)

» Organización de una computadora

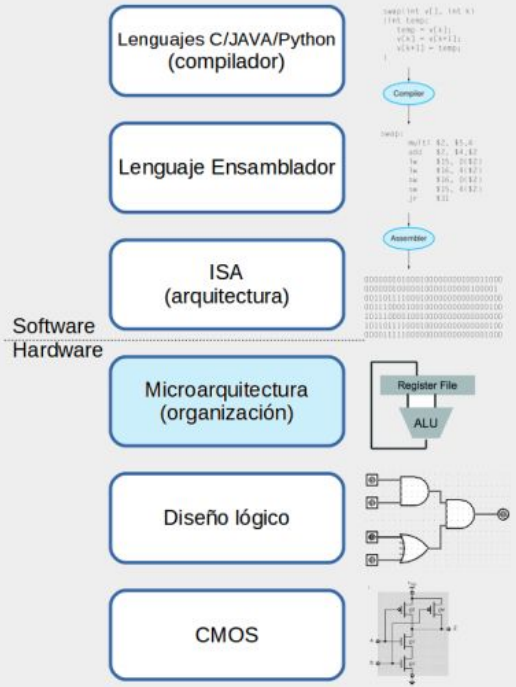
Es el diseño e implementación de la arquitectura (ISA) del microprocesador

Micro Hoy en día, a la organización de una computadora se la conoce como su **microarquitectura**

Familia Una arquitectura puede tener muchas organizaciones diferentes.

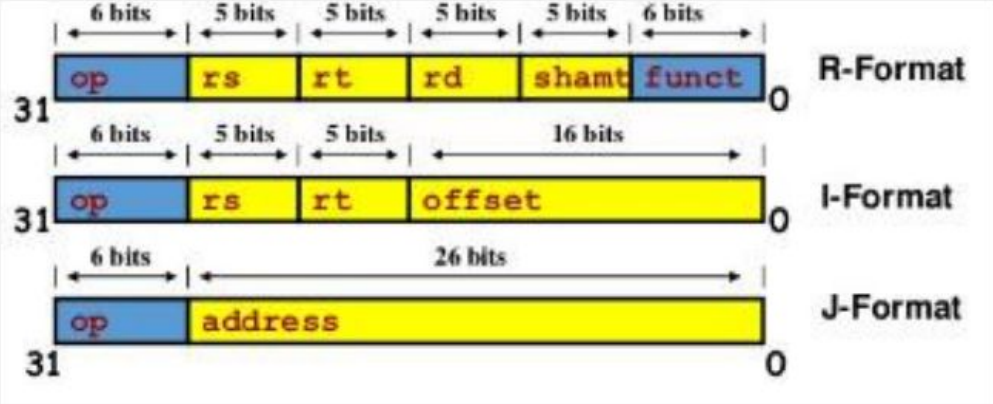
Ortogonales La arquitectura y la organización son ortogonales; es decir, son totalmente independientes

La ***arquitectura*** especifica lo ***que*** puede hacer una computadora y la ***organización*** especifica ***cómo*** lo hace.



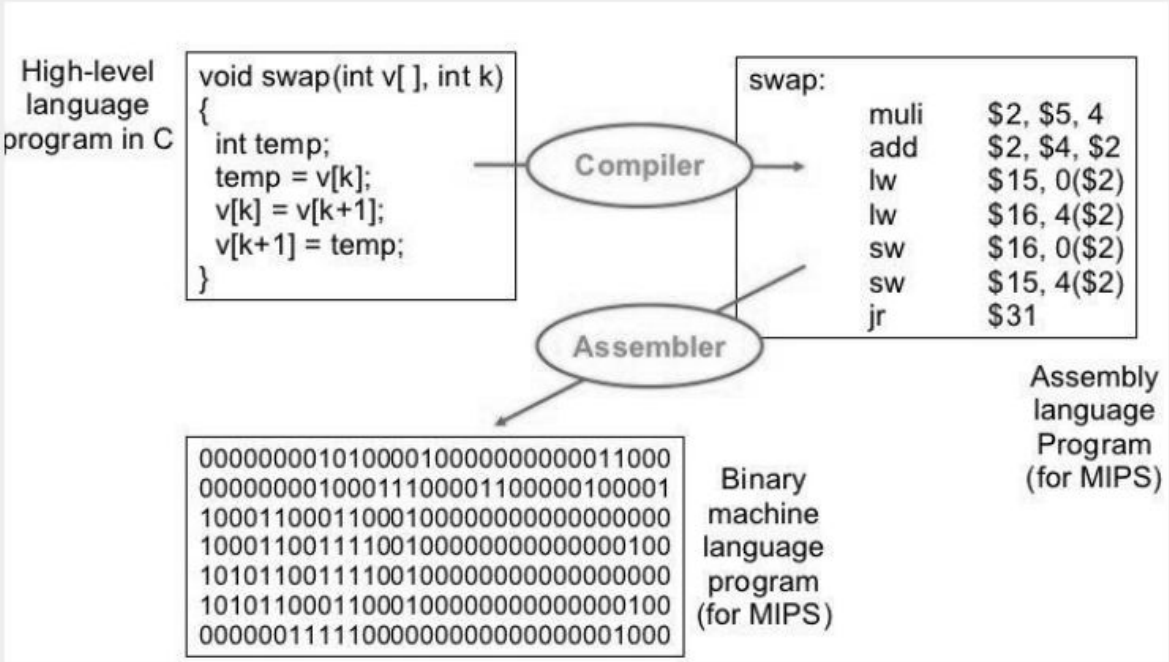
» Formatos de instrucciones en MIPS

- **3 Formatos (fácil decodificación en hardware)**
- Las instrucciones que usan 3 registros son de tipo R (ejemplo: add, sub, and, or)
- Las instrucciones que usan 2 registros y una constante son de tipo I (ejemplo: lw, sw, addi, ori, beq)
- La instruccion j es de tipo J



» Formatos de instrucciones en MIPS

Un ejemplo completo de traducción al lenguaje de la máquina



Instrucciones mas comunes

Cargar constantes: `lui, addi`

ALU: `addi, addu, add, sub, subu, and, or`

Transferir datos entre procesador y RAM: `lw, sw, lb, sb, lbu`

Bifurcacion (y saltos condicionales): `beq, bnq, slt`

Saltos: `j, jr, b`

Por combinar bifurcación y saltos

se puede sintetizar en código máquina

cualquier combinacion de

if/then/else, bucles FOR o WHILE, comparaciones <, >, ==, <=, >=

» Instrucciones de salto condicional

Instrucciones de transferencia de Control

- Salto condicional: bne, beq
- ¿Qué sucede con saltar si es menor qué?
- Nueva instrucción:

```
slt $t0, $s1, $s2

Significado:      if $s1 < $s2 then
                      $t0 = 1
                  else
                      $t0 = 0
```

- Puede ser utilizada para construir

  `"blt $s1, $s2, Etiqueta"`

  - Se pueden construir estructuras de control de ejecución generales
- El ensamblador necesita utilizar un registro *temporal*
  - Convención de uso de registros

Instrucciones mas comunes

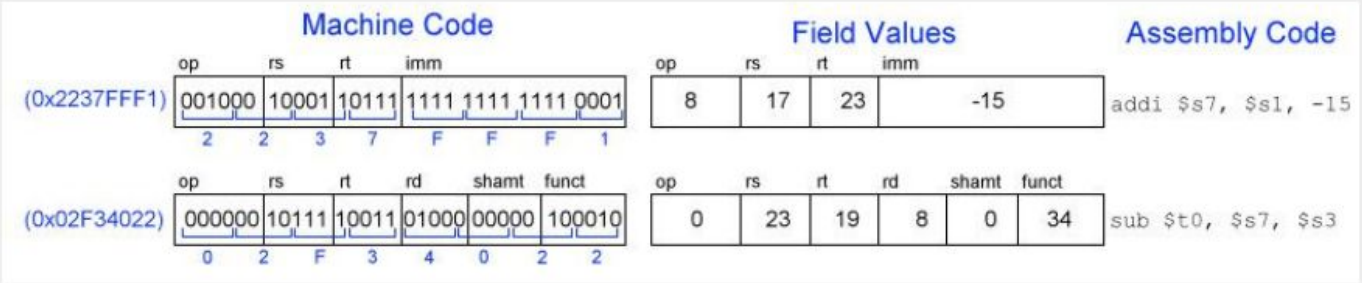
```
/*
 * obtener el mayor valor de un arreglo
 */

#define N 20

char p[20];

main() {
     char mayor = 0;
     int i;

     for (i=0; i<N; i++)
          if (p[i] > mayor)
               mayor = p[i];


}
```

```
.data      # suponemos segemento de datos en 0x00080000
     p: .space 20

.text
     main:
          addi $8, $0, 0        # mayor
          addi $9, $0, 0        # i
          addi $12, $0, 20      # 20

          lui $10, 0x0008       # direccion base de p[]

       for:
          addu $10, $10, $9     # direccion de p[i]
          lbu $11, 0($10)       # carga p[i]
          slt $13, $8, $11      # si p[i] es mayor
          beq $13, $0, cont
          addu $8, $11, $0

          cont:
            addi $9, $9, 1
            beq $9, $12, salir
            j for

       salir:
```

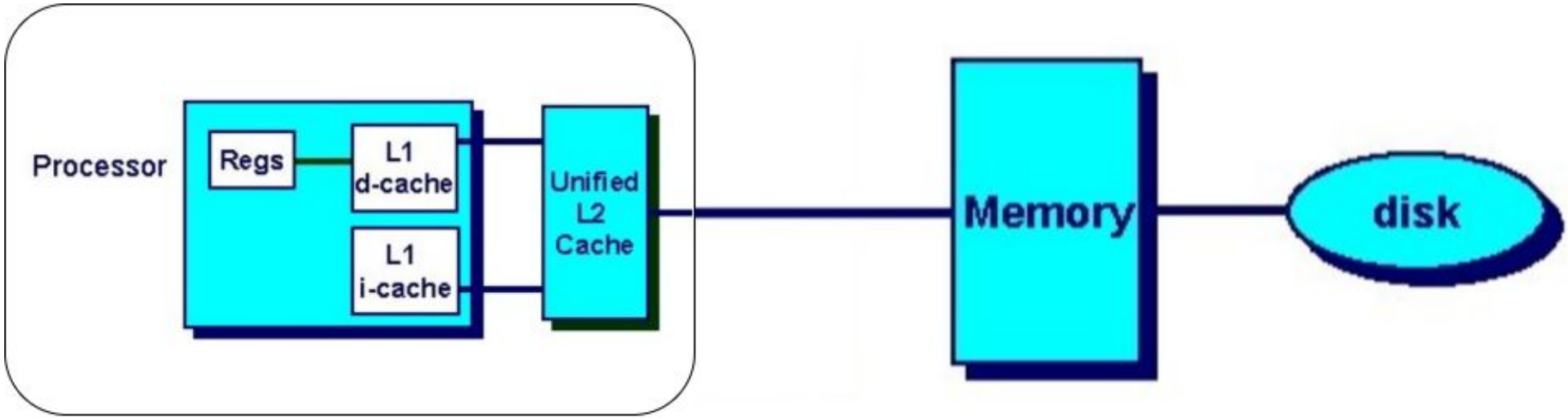
» Implementación de la Microarquitectura

Diseño de la Microarquitectura (diseño del procesador)

1. Analizar el conjunto de instrucciones (ISA)
   * **Obtener los requerimientos del camino de datos**



¿Qué tipos de instrucciones (formato) MIPS son las dos anteriores?


Processor
Regs
L1
d-cache
L1
i-cache
Unified
L2
Cache
Memory
disk

Text

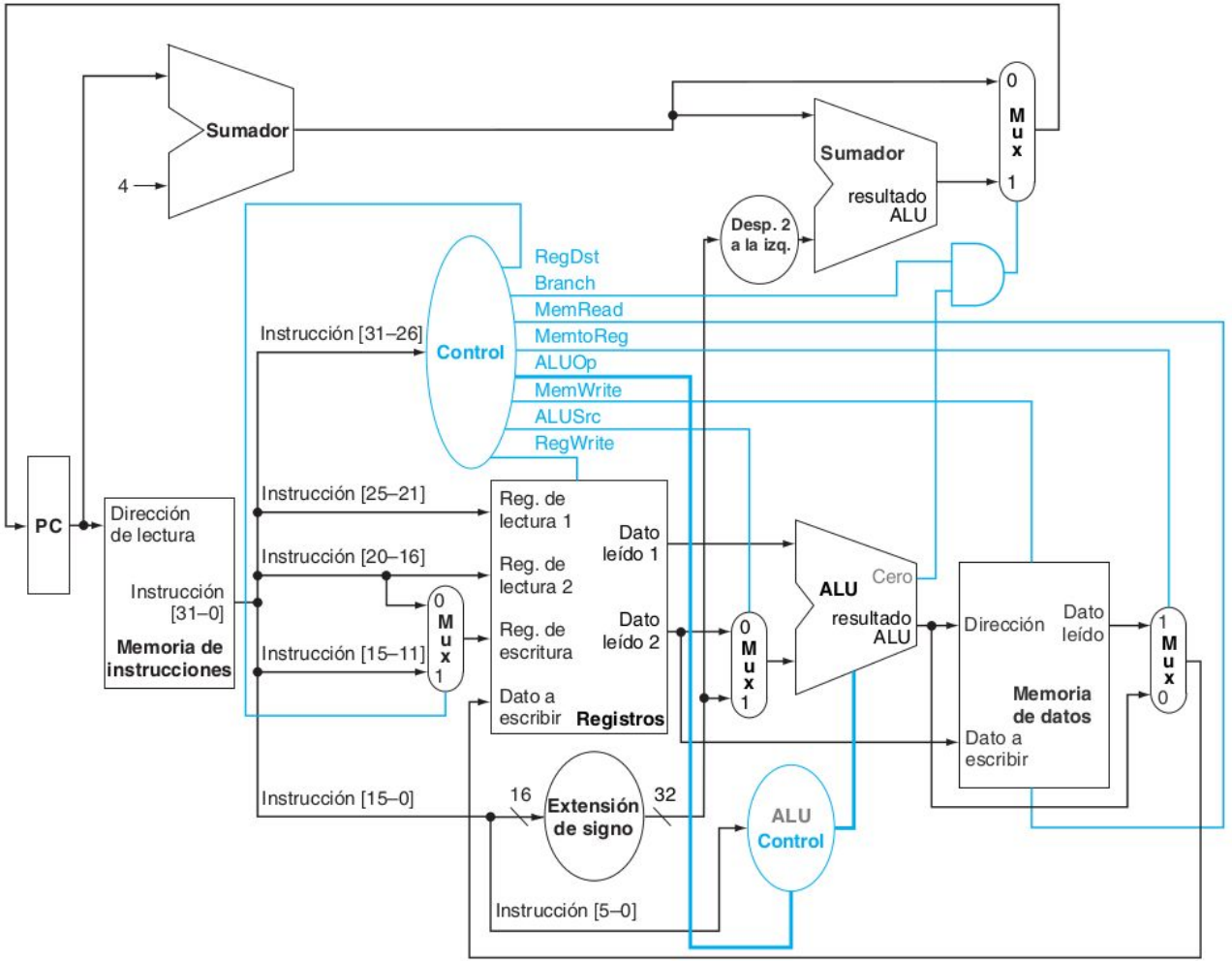


FIGURA 4.17 Un sencillo camino de datos con la unidad de control. La entrada de la unidad de control está compuesta por los