

# Circuitos Digitales II

introducción al codiseño SW/HW

Ferney Alberto Beltrán Molina



Febrero 2015

Nombre: Ferney Alberto Beltrán Molina, Ing, MSc, ...  
Email: fabeltranm@unal.edu.co  
: ferney.beltran@urjc.es

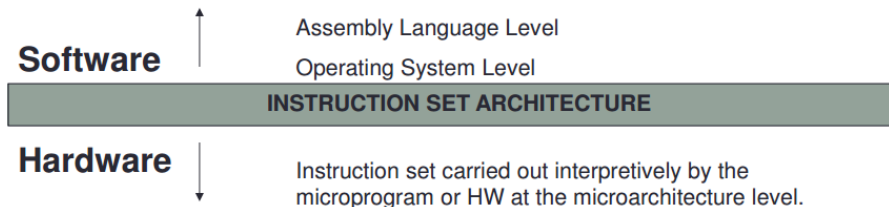
Sigue las instrucciones del programa al pie de la letra. Suma y compara números, ordena activarse a los dispositivos de I/O, etc.

El procesador consta de dos componentes:

- El datapath. Ejecuta operaciones aritméticas y lógicas.
- El control. Ordena al datapath, memoria y dispositivos de I/O lo que hay que hacer de acuerdo al program

# ISA, conjunto de instrucciones

La frontera entre el hardware y el software



Cada instrucción es directamente ejecutada por el hardware

# Comó se diseña el set de instrucciones (ISA)

- Simplicidad favorece la regularidad
- Cuanto más pequeños, más rápido
- Hacer lo común, lo más rápido
- Un buen diseño exige buenos compromisos

Set de instrucciones para arquitecturas basadas en pilas.

Set de instrucciones para arquitecturas basadas en registros.

# Abstracción

High-level  
language  
program  
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

C compiler

Assembly  
language  
program  
(for MIPS)

```
swap:
  muli $2, $5, 4
  add $2, $4, $2
  lw $15, 0($2)
  lw $16, 4($2)
  sw $16, 0($2)
  sw $15, 4($2)
  jr $31
```

Assembler

Binary machine  
language  
program  
(for MIPS)

```
000000001010000100000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
100011001111001000000000000000100
10101100111100100000000000000000
101011000110001000000000000000100
0000001111100000000000000000100
```

- Con un formato binario. El hardware solo entiende bits
- Los objetos físicos son bits, bytes, palabras (words).
- Tamaño típico de palabra: 4 u 8 bytes (32 o 64 bits).
- Se identifica por un opcode (código de operación)  
**add \$s0,\$s1, \$s2**
- Requiere de 0 a 3 operandos.  
identificador de la zona donde estan almacenados  
memoria, registros, stack

## Implicito

- El opcode implica la dirección de los operandos.
- Ejemplo en una máquina de pila (stack) **add**
- La instrucción saca (pop) dos valores de la pila, hace la suma y deja (push) el resultado en la pila



## Explícito

- Las direcciones vienen en los operandos.
- Ejemplo de MIPS:  
**add \$s0,\$s1, \$s2**
- Dos operandos fuentes: s1 y s2
- Un operando destino: s0
- $s0 = s1 + s2$

# Clasificación de las Instrucciones

- Instrucciones de Transferencia de Datos

Movimiento (Move), copia (copy), carga (load), guarda (store),  
Alteración Datos (Clear, Inc, Dec) Rotación Bits (Shift, Rotate)

- Instrucciones Aritméticas

(Add, Sub, Mult, Div)

- Instrucciones Lógicas

(And, Or, Xor)

- Instrucciones Booleanas

(Set bit, Clear bit, Jump if bit set, Jump if bit clear)

- Instrucciones de Salto

Control (Jump, Conditional jumps) Relacionadas con Subrutinas (Push, Pull)  
Relacionadas con Interrupción (Retorno de Int.)

## Unidad aritmético-lógica

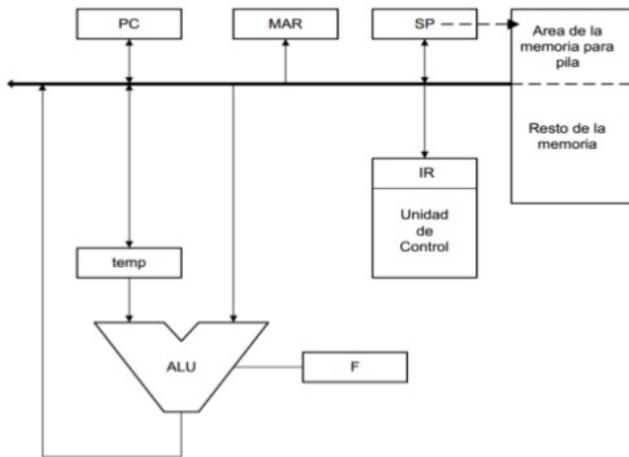
La cual se encarga de realizar las operaciones que requiera el algoritmo. Lógicamente las arquitecturas no tienen implementadas todas las posibles funciones matemáticas o funciones aritméticas (instrucciones)

## Banco de registros

Mantiene almacenada la información o los datos

Registros de acceso a memoria, registros programados, registro especiales etc.

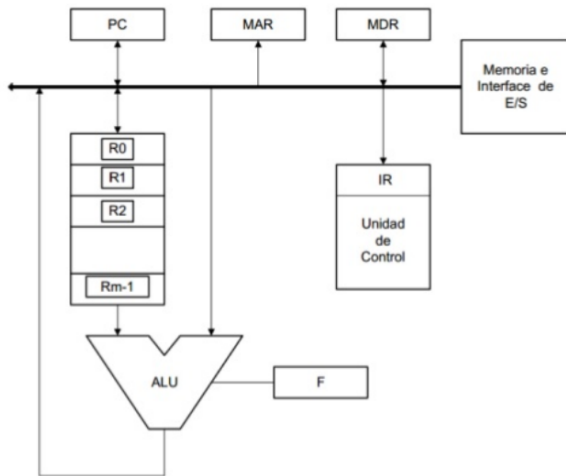
# Datapath (stack)



**SP: Puntero de la Pila:** registro con la dirección de la ultima palabra insertada en la pila **TOS, Top of Stack**

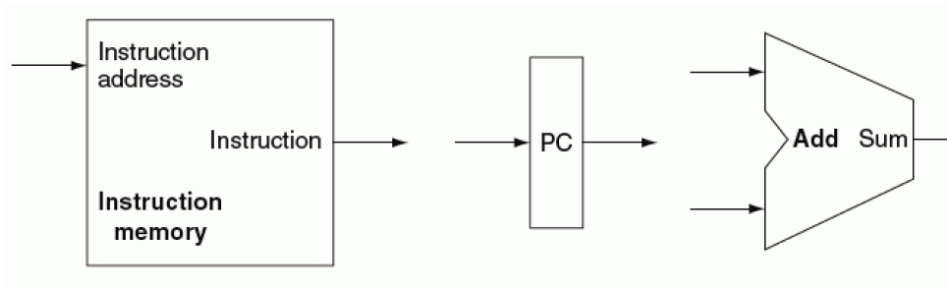
en funcion de Sp se obtiene el proximo registro **NOS Next of Stack**

# Datapath (Acumulador)



Banco de registros de propósito general ( $R_0 \dots R_{m-1}$ )  
Maquinas de dos (Lectura destructiva) o tres operandos

# Datapath (Componentes básicos)

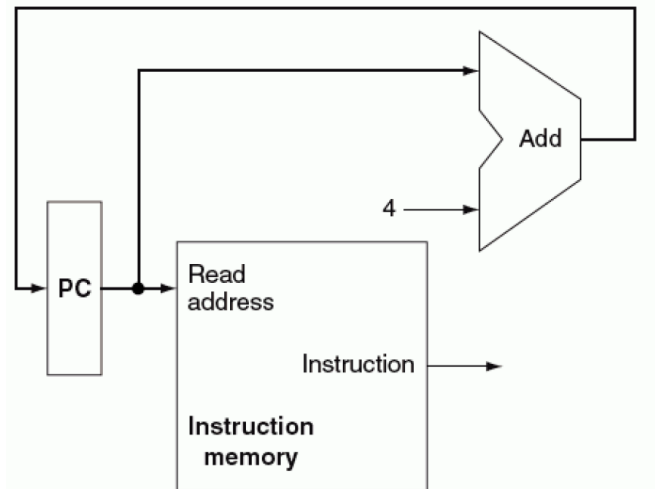


Una memoria para guardar y leer instrucciones

Un registro, llamado PC (contador de programa), para guardar la dirección de la instrucción actual.

Un sumador para incrementar el PC.

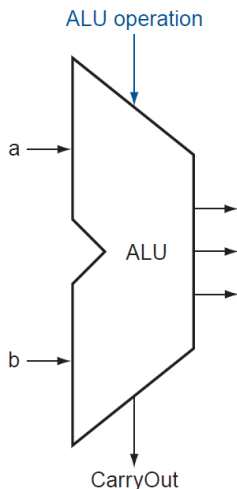
# Datapath (Componentes básicos)



Obtener la instrucción de la memoria.

Incrementar el PC para preparar la ejecución de la instrucción siguiente.

# Datapath (Componentes básicos)



code	operation
0	$T$
1	$N$
2	$T + N$
3	$T \text{ and } N$
4	$T \text{ or } N$
5	$T \text{ xor } N$
6	$\sim T$
7	$N = T$
8	$N < T$
9	$N \text{ rshift } T$
10	$T - 1$
11	$R$
12	$[T]$
13	$N \text{ lshift } T$
14	depth
15	$N \text{ u} < T$



field	width	action
$T'$	4	ALU op, replaces $T$ , see table II
$T \rightarrow N$	1	copy $T$ to $N$
$R \rightarrow PC$	1	copy $R$ to the $PC$
$T \rightarrow R$	1	copy $T$ to $R$
dstack $\pm$	2	signed increment data stack
rstack $\pm$	2	signed increment return stack
$N \rightarrow [T]$	1	RAM write

