

Arquitectura de Computadoras

(Cód. 5561)
1° Cuatrimestre 2018

Dra. Dana K. Urribarri
DCIC - UNS

Introducción a la materia

Cátedra

- Profesora: Dra. Dana K. Urribarri
- Asistente: Lic. Gabriela Diaz
- Ayudantes:
 - Santiago Barboza
 - Agustín Pacheco Di Santi
 - Martín Burón Brarda
 - ...



Cronograma tentativo

- Primer Parcial:
 - Jueves 3 de mayo
- Segundo Parcial:
 - Jueves 14 de junio
- Recuperatorio:
 - Jueves 28 de junio

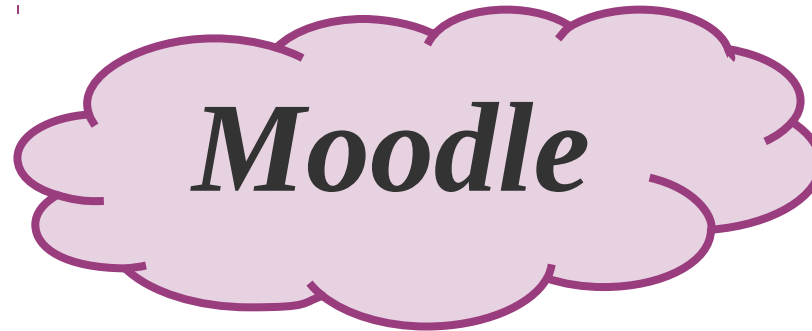


Horarios

- Teoría
 - Martes y Jueves de **8:30** a 10 hs
- Práctica
 - Martes y Jueves de 10 a 12 hs
- Lugar
 - Aula 4 del complejo Palihue.

Prácticos y Transparencias

- Las transparencias y prácticos así como todo otro material del que se disponga una versión electrónica será dejado en la página web de la asignatura.



- Todo otro material adicional que se requiera será oportunamente dejado en la fotocopiadora del CeCom.



Material de estudio

- Las transparencias son sólo una guía.
- Pueden contener errores u omisiones.
- El material de estudio son los (muchos) libros de texto sobre arquitectura de computadoras.



Material de estudio

- Morris Mano, Kime & Martin. Logic and computer design fundamentals. Prentice Hall (2015, 5ta Ed) y capítulos suplementarios.
- M. Morris Mano & Michael D. Celetti. Digital Design: With an Introduction to the Verilog HDL. Pearson. (2015, 5ta Ed.)
- Israel Koren. Computer Arithmetic Algorithms. 2da Edición, A K Peters, Natick, MA, 2002.
Adapted from Koren, UMass. Copyright 2008 Koren, UMass and A.K. Peters.
- Behrooz Parhami. Computer Arithmetic: Algorithms and Hardware Designs. Oxford University Press, New York, 2002.

Material de estudio

- Jean-Loup Baer. Multiprocessor Architecture. From simple pipelines to chip multiprocessor. Cambridge University Press. 2010.
- David M. Harris & Sarah L. Harris. Digital Design and Computer Architecture. Elsevier. 2013, 2da Ed.
- John L. Hennessy & David A. Patterson. Computer Architecture. A Quantitative Approach. Elsevier Inc. 2012, 5ta Ed.
- David A. Patterson & John L. Hennessy. Computer Organization and Design. The Hardware/Software Interface. Elsevier Inc. 2014, 5ta Ed.

Material de estudio

- Abraham Silberschatz & Peter Baer Galvin. Operating System Concepts. Addison Wesley 1998. 5ta Ed. (en adelante).
- Bruce Jacob, Spencer W. Ng & David T. Wang. Memory Systems. Cache, DRAM, Disk. Elsevier 2008.
- William Stallings. Computer Organization and Architecture. Designing for Performance. Pearson. 2013, 9na Ed.

Material de estudio

Suplementario

- Editor Wai-Kai Chen. *The VLSI Handbook*. CRC Press. (2da Ed. 2007)
- AMD64 Technology. *AMD64 Architecture Programmer's Manual Volume 2: System Programming*. Revision 3.23, Mayo 2013.
developer.amd.com/wordpress/media/2012/10/24593_APM_v21.pdf
- J. Hayes, *Computer Architecture and Organization*. McGraw-Hill 1978

Sistema de cursado

- Para cursar la materia:
 - Aprobar todos los exámenes parciales.
 - Aprobar el recuperatorio final.
- Aprobar el examen final (regular o libre).
 - El curso no se promociona.

Arquitectura vs. Microarquitectura

Arquitectura

- La *arquitectura* de una computadora la define el set de instrucciones (ISA)
 - Descripción funcional de las operaciones soportadas por el hw.
 - Descripción de cómo invocar las funciones
 - Descripción de la ubicación de los operandos (registros y memoria)

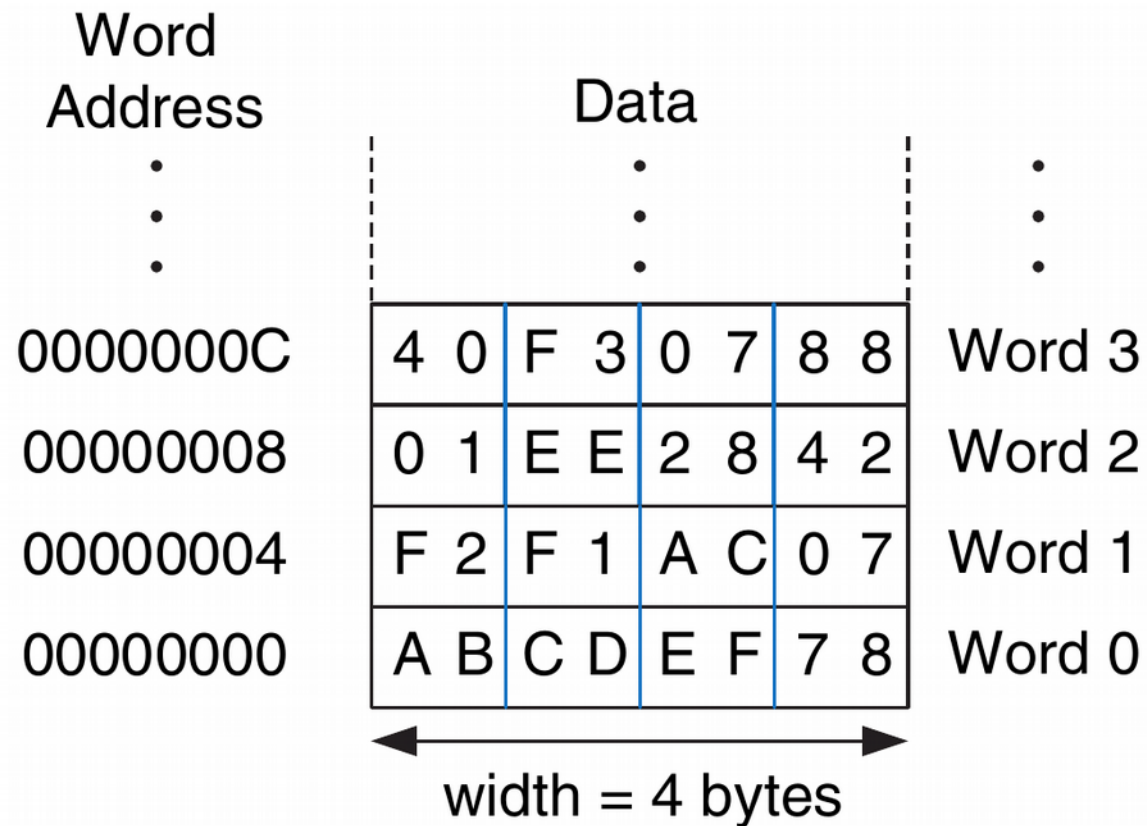
Ej: x86, MIPS, SPARC, PowerPC.

Arquitecturas n -bits

- El procesador opera con palabras (words) de n -bits.
 - Usualmente: registros de n -bits
- Microprocesadores: 64-bits, 32-bits
- Sistemas embebidos: 16-bits, 8-bits

Direccionamiento de la Memoria


- Direccionamiento al byte



Direccionamiento de la Memoria

- Direccionamiento al word

Word Address	Data	
⋮	⋮	⋮
00000003	4 0 F 3 0 7 8 8	Word 3
00000002	0 1 E E 2 8 4 2	Word 2
00000001	F 2 F 1 A C 0 7	Word 1
00000000	A B C D E F 7 8	Word 0



width = 4 bytes

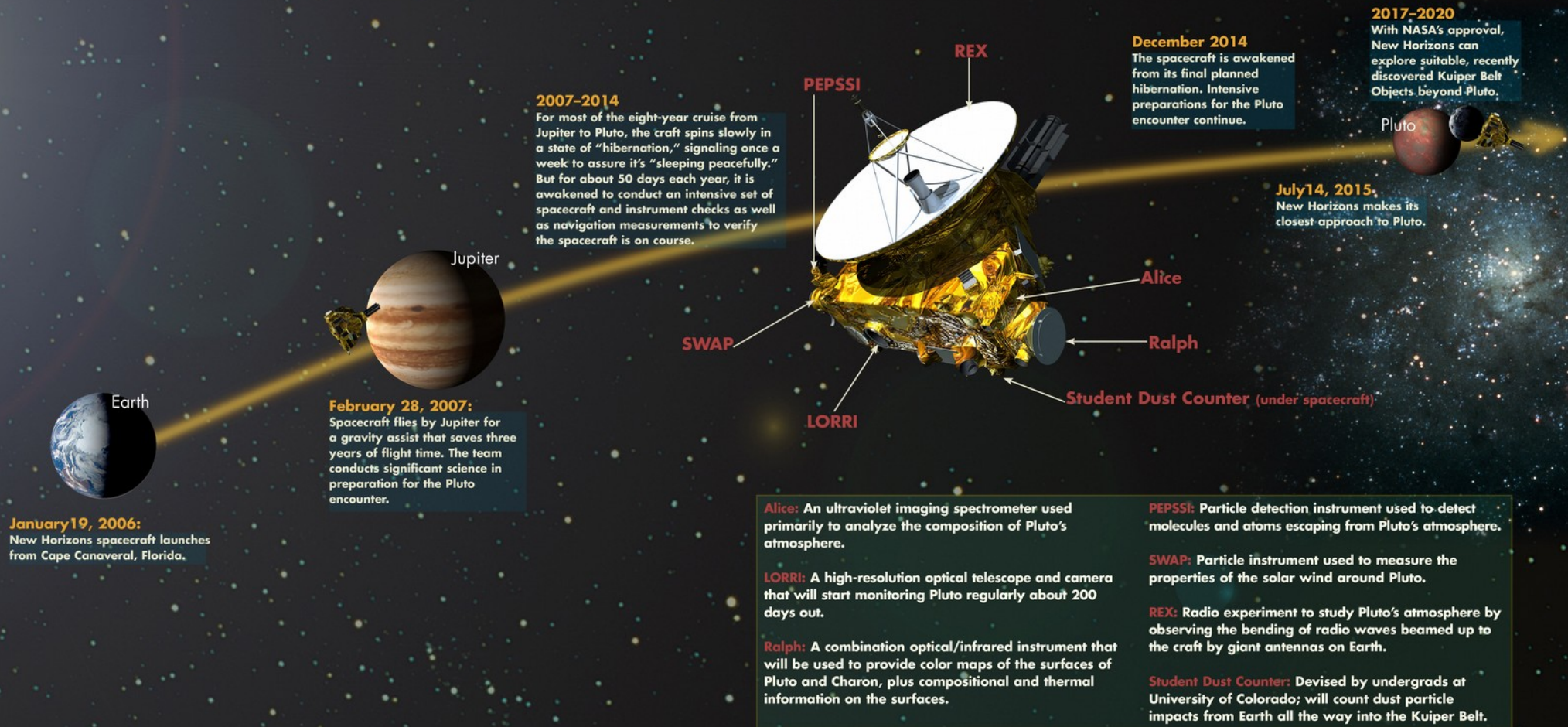
Arquitectura MIPS

- Vamos a considerar MIPS:
 - RISC (*reduced instruction set computer*)
 - 32 y 64 bits
- Implementaciones de MIPS se usan en Sistemas Embebidos:
 - Dispositivos con Windows CE
 - Routers
 - Consolas como Nintendo 64, Sony PlayStation, PlayStation 2 y PlayStation Portable.
 - Industria aeroespacial.

New Horizons

Ten Years and Three Billion Miles...

Mongoose-V:
versión resistente a
la radiación del
MIPS R3000
(1997)



Arquitectura MIPS 32 bits

- Palabras (words) de 32 bits
 - 32 registros de 32 bits
 - Instrucciones 32 bits
 - 32 bits de direccionamiento
 - 32 bits de dato

Arquitectura MIPS: Registros

Name	Number	Use
\$0	0	the constant value 0
\$at	1	assembler temporary
\$v0-\$v1	2-3	function return value
\$a0-\$a3	4-7	function arguments
\$t0-\$t7	8-15	temporary variables
\$s0-\$s7	16-23	saved variables
\$t8-\$t9	24-25	temporary variables
\$k0-\$k1	26-27	operating system (OS) temporaries
\$gp	28	global pointer
\$sp	29	stack pointer
\$fp	30	frame pointer
\$ra	31	function return address

Arquitectura MIPS: Registros

Name	Number	Use
\$0	0	the constant value 0
\$at	1	assembler temporary
\$v0-\$v1	2-3	function return value
\$a0-\$a3	4-7	function arguments
\$t0-\$t7	8-15	temporary variables
\$s0-\$s7	16-23	saved variables Variables
\$t8-\$t9	24-25	temporary variables
\$k0-\$k1	26-27	operating system (OS) temporaries
\$gp	28	global pointer
\$sp	29	stack pointer
\$fp	30	frame pointer
\$ra	31	function return address

a = b + c



```
// a → $s0  
$s1 = b  
$s2 = c  
add $s0, $s1, $s2
```

Arquitectura MIPS: Registros

Name	Number	Use
\$0	0	the constant value 0
\$at	1	assembler temporary
\$v0-\$v1	2-3	function return value
\$a0-\$a3	4-7	function arguments
\$t0-\$t7	8-15	temporary variables Temporales
\$s0-\$s7	16-23	saved variables
\$t8-\$t9	24-25	temporary variables Temporales
\$k0-\$k1	26-27	operating system (OS) temporaries
\$gp	28	global pointer
\$sp	29	stack pointer
\$fp	30	frame pointer
\$ra	31	function return address

$a = b + c - d$



```
// a → $s0
$s1 = b
$s2 = c
$s3 = d
sub $t0, $s2, $s3
add $s0, $s1, $t0
```


Arquitectura MIPS: Instrucciones

MIPS incluye:

- **Instrucciones R-type**

Instrucciones aritmético-lógico

- **Instrucciones de escritura y lectura en memoria**

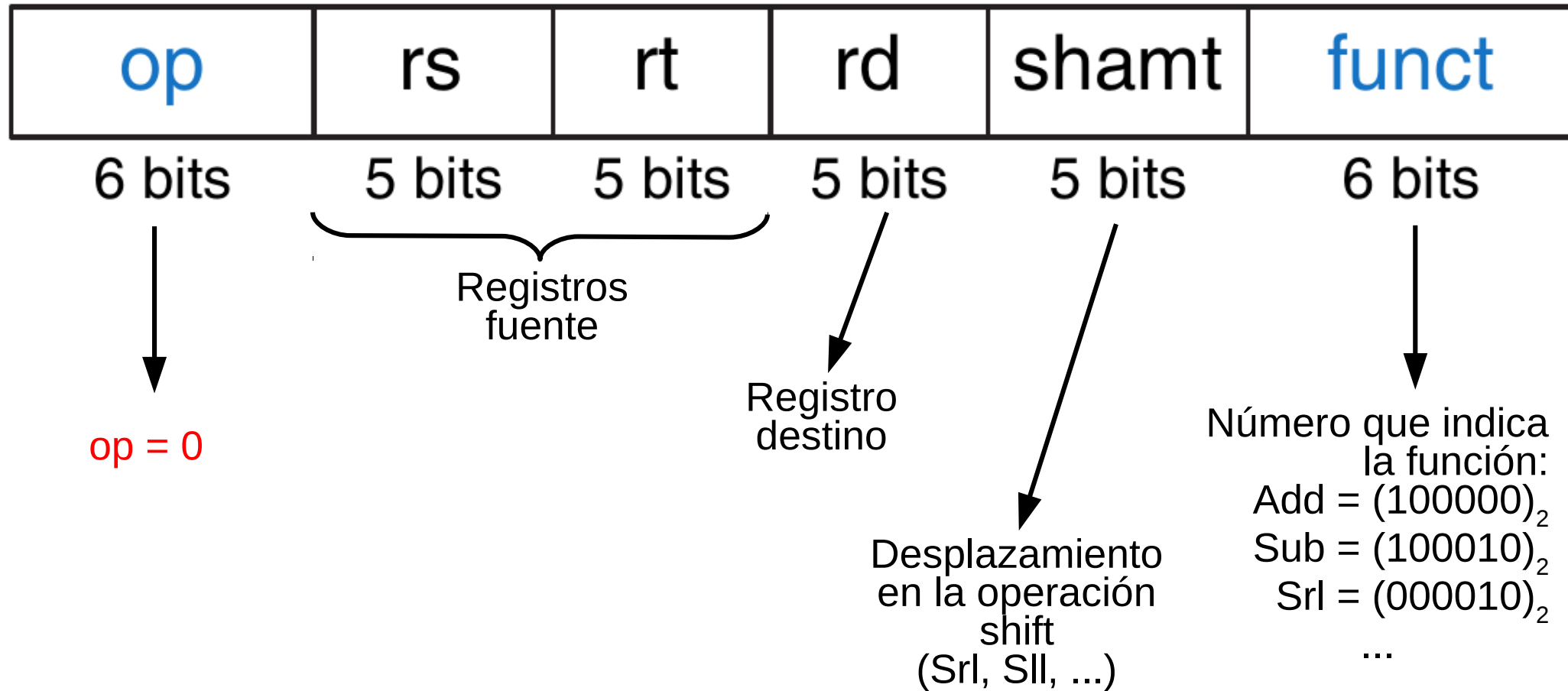
Load y store

- **Instrucciones de bifurcación**

Saltos condicionales (branch) e incondicionales (jump)

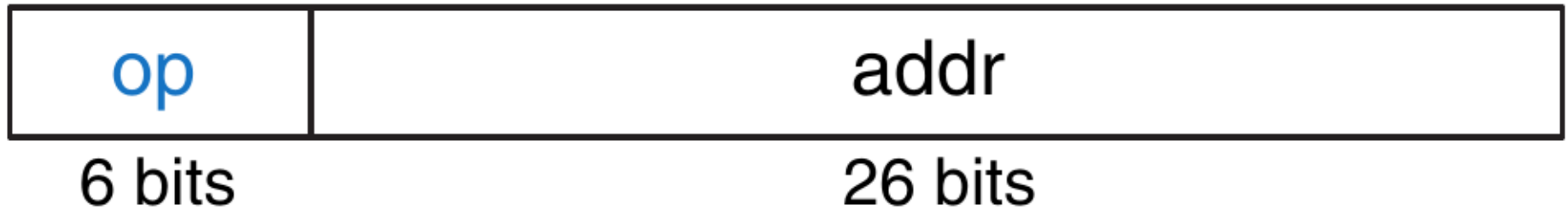
Arquitectura MIPS: Formato de instrucción

R-type



Arquitectura MIPS: Formato de instrucción

J-type



000010 (2)	j label	jump	PC = JTA
000011 (3)	jal label	jump and link	\$ra = PC + 4, PC = JTA

Arquitectura MIPS: Formato de instrucción

I-type

op	rs	rt	imm
6 bits	5 bits	5 bits	16 bits
000100 (4)	beq rs, rt, label	branch if equal	if ([rs] == [rt]) PC = BTA
000101 (5)	bne rs, rt, label	branch if not equal	if ([rs] != [rt]) PC = BTA
000110 (6)	blez rs, label	branch if less than or equal to zero	if ([rs] ≤ 0) PC = BTA
000111 (7)	bgtz rs, label	branch if greater than zero	if ([rs] > 0) PC = BTA
001000 (8)	addi rt, rs, imm	add immediate	[rt] = [rs] + SignImm
001001 (9)	addiu rt, rs, imm	add immediate unsigned	[rt] = [rs] + SignImm
100011 (35)	lw rt, imm(rs)	load word	[rt] = [Address]
101011 (43)	sw rt, imm(rs)	store word	[Address] = [rt]

Microarquitectura

- La *arquitectura* de una computadora la define el set de instrucciones (ISA)
 - Descripción funcional de las operaciones soportadas por el hw.
 - Descripción de cómo invocar las funciones
 - Descripción de la ubicación de los operandos (registros y memoria)

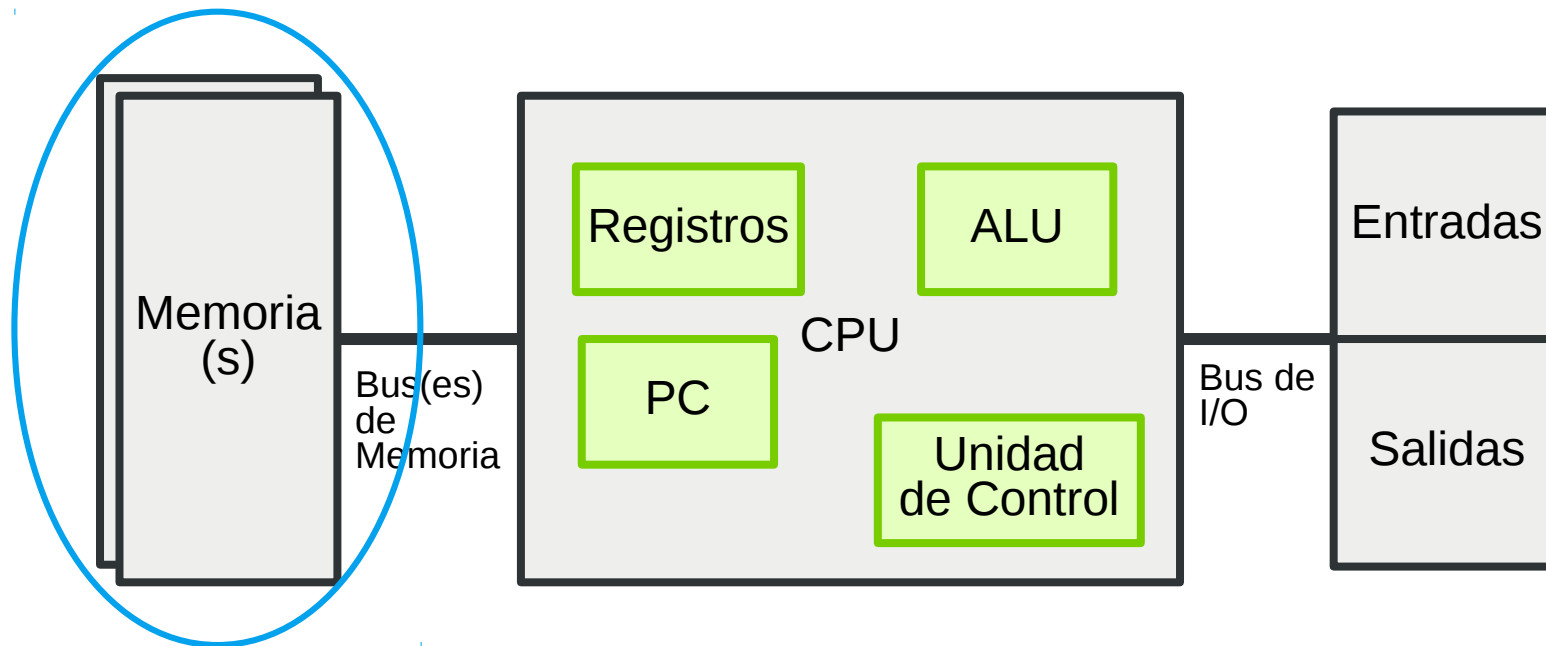
Ej: x86, MIPS, SPARC, PowerPC.

Microarquitectura

- La arquitectura **no** define la implementación del hardware.
 - Intel y AMD implementan la arquitectura x86.
 - Diferencias en rendimiento, precio y consumo.
- La *microarquitectura* es la combinación específica del hw (registros, memoria, ALUs, etc.) que implementa las operaciones.

Microarquitectura de Microprocesador (Modelo von Neumann)

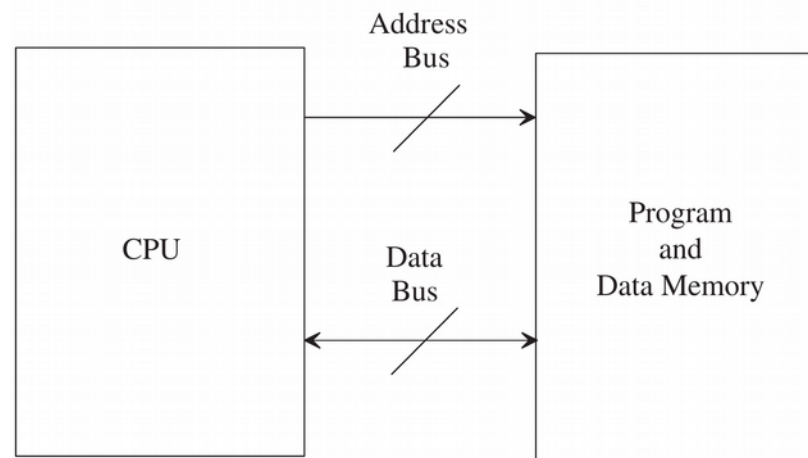
Bloques de una Microarquitectura



- Modelo von Neumann
- Modelo Harvard

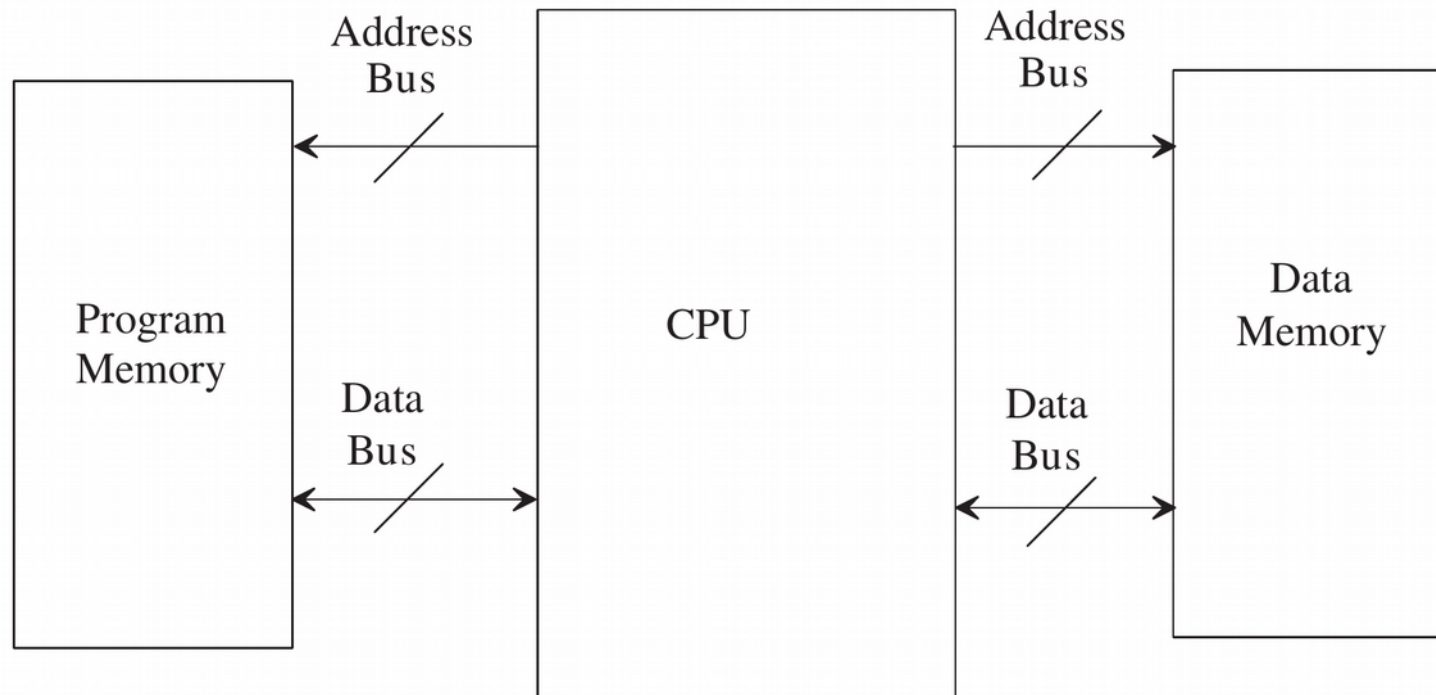
Modelo von Neumann

- La memoria almacena instrucciones, datos y resultados intermedios y finales.
Se implementa como una jerarquía.
- Tiene un bus de datos y otro de dirección.



Modelo Harvard

- Usa memoria y buses separados para instrucciones y datos.
- Pueden ejecutar instrucciones y acceder a datos simultáneamente.
- Requiere 4 buses, dos de direcciones y dos de contenido.



Microprocesador vs Microcontrolador

- Microprocesador:

- Chip que contiene la CPU.
- Modelo von Neumann. La memoria es externa al microprocesador.

Modelo Harvard modificado

[Profundizar en la materia](#)

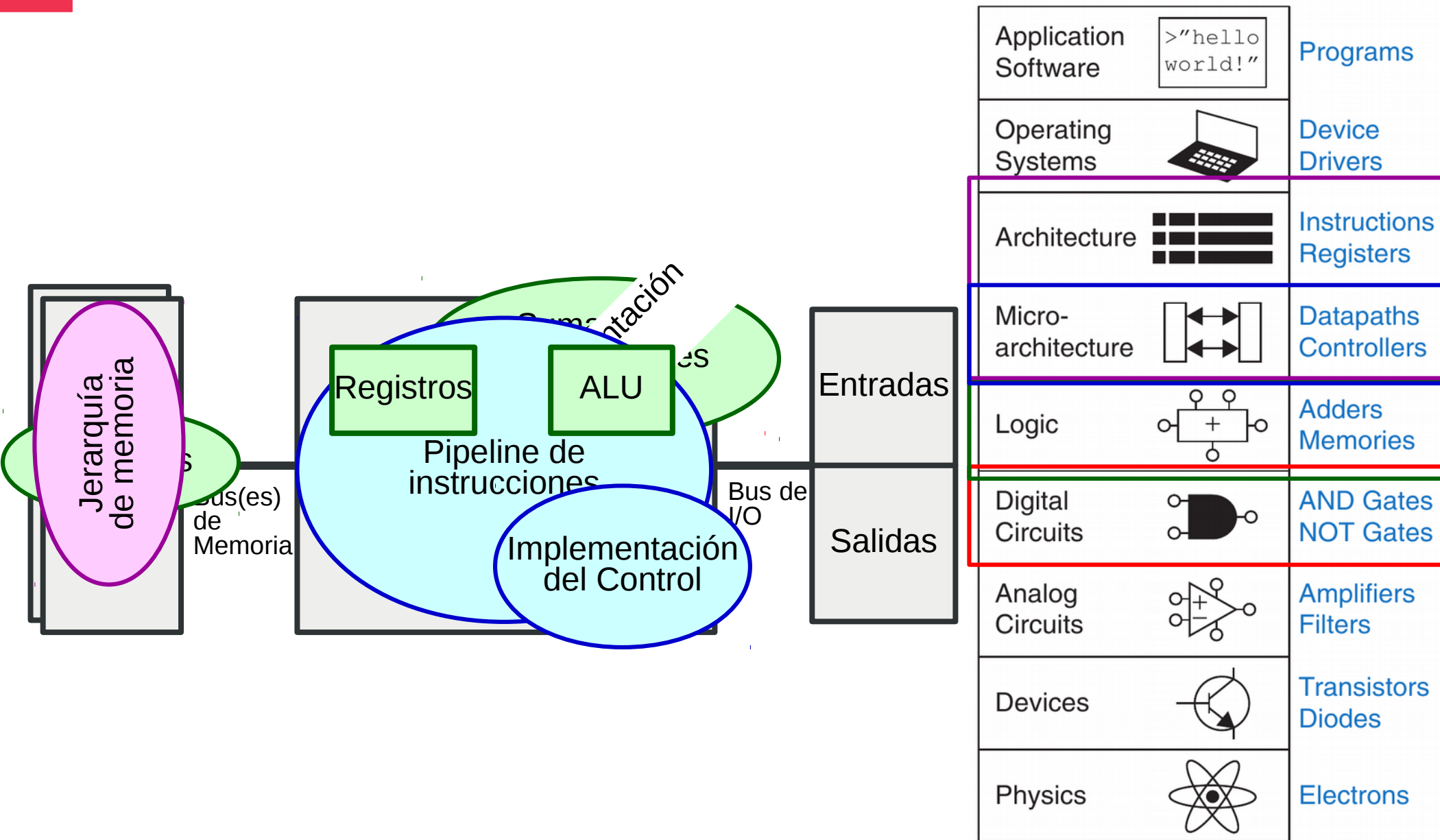
- Microcontrolador:

- Un único chip que contiene CPU, memoria, entrada/salida y periféricos.
- Modelo Harvard.

Microarquitectura

- La microarquitectura se divide en dos partes que interactúan entre sí:
 - Datapath
 - Unidad de Control
- Datapath
 - Opera sobre los datos.
 - Involucra memorias, registros, ALUs y multiplexores.
- Unidad de Control
 - A partir de la instrucción actual le indica al datapath cómo ejecutar la instrucción.
 - Produce señales de multiplexado, habilitaciones de registros y señales a la memoria para controlar las operaciones del datapath.

Niveles de abstracción



Bibliografía



- Capítulo 1,6,B. David M. Harris & Sarah L. Harris. *Digital Design and Computer Architecture*. Elsevier. 2013, 2da Ed.
- Capítulo 1. Jean-Loup Baer. *Multiprocessor Architecture. From simple pipelines to chip multiprocessor*. Cambridge University Press. 2010.
- Capítulo 4. David A. Patterson & John L. Hennessy. *Computer Organization and Design. The Hardware/Software Interface*. Elsevier Inc. 2014, 5ta Ed.