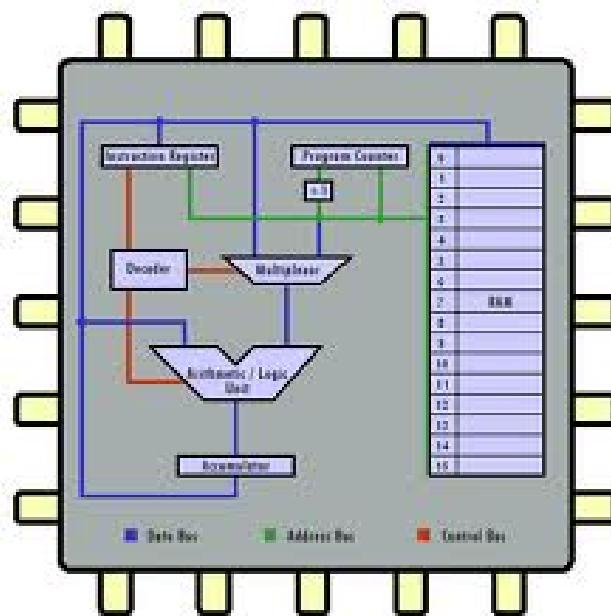


# 1º CLASE



***Laboratorio de***  
***Microcomputadoras***      **(66-09)**

---

***Laboratorio de***  
***Microprocesadores***      **(86-07)**

---

**2º cuatrimestre año 2017**

# Docentes y horarios

---

- **Curso 1:** Martes, 19:00 a 22:00 hs  
Ing. Gerardo Stola
- **Curso 2:** Miércoles, 19:00 a 22:00 hs
  - Ing. Ricardo Arias
- **Curso 3:** Jueves, 19:00 a 22:00 hs  
Inga. Graciela Ratto
- **Teórica:** Viernes, 16:00 a 19:00 hs  
Ing. Guillermo Campiglio

# Correlatividades de la materia Plan (Electrónicos)

---

Para cursar esta materia se debe tener  
**aprobadas** las materias:

- [66-01] Técnica Digital (Electrónicos)
  - [66-02] Laboratorio.
- 
- [66-02] Técnica Digital (Sistemas)
  - [66-70] Estructura del Computador

# Correlatividades de la materia

## Plan 2009 (Electrónicos)

Para cursar esta materia  
se debe tener  
**aprobadas** las materias:

[86-01] Técnica Digital

[86-03] Dispositivos  
Semiconductores

[86-04] Análisis de  
Circuitos

[95-04] Análisis  
Numérico1

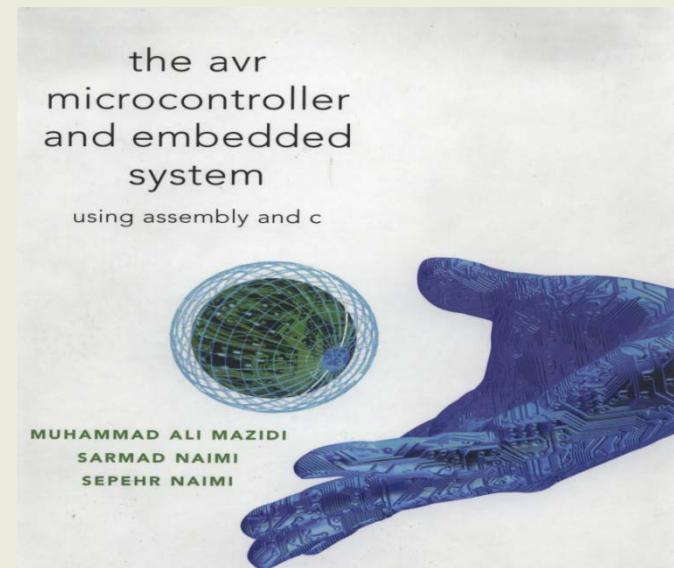
# Bibliografía..... AVR

Hojas de datos del micro controlador elegido

<http://www.atmel.com/products/microcontrollers/avr/megaAVR.aspx>

The avr microcontroller and embeddeed system. Using assembly and C

Muhammad Ali Mazidi  
Sarmad Naimi  
Sepehr Naimi

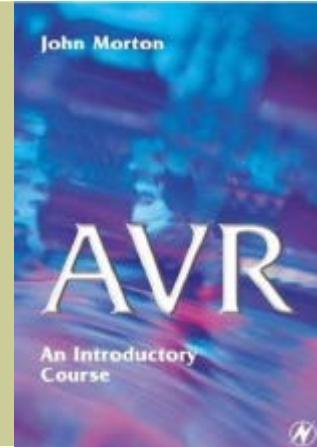


PEARSON

[www.pearsonhighered.com](http://www.pearsonhighered.com)

# Bibliografía..... AVR

**AVR An Introductory Course**  
**John Norton**



Micro controladores AVR  
Configuración Total de  
Periféricos  
ASDRÚBAL LÓPEZ CHAU.

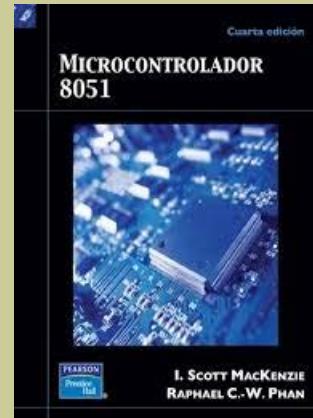


Apuntes / Teoricas de cátedra

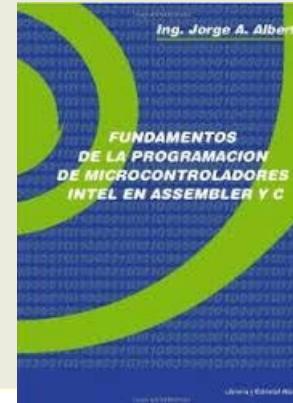
# Bibliografía.....8051

---

**Microcontrolador 8051**  
**4<sup>a</sup>. Edición,**  
**Scott McKenzie,**  
**Editorial Prentice-Hall .....****2007**



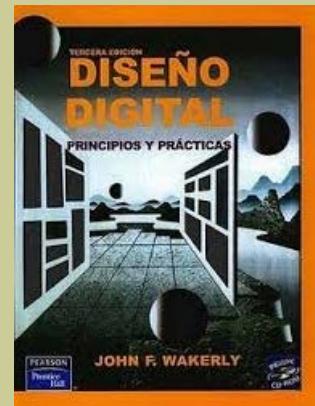
**Fundamentos de la programación  
de microcontroladores Intel en  
Assembler y C,**  
**Ing Jorge Alberto,**  
**Editorial Alsina.....2005**



# Bibliografía

---

Técnicas Digitales  
Diseño Digital , principios y  
prácticas  
Wakerly J.,  
Ed. Prentice Hall. 2001



# Metodología

---

1. Clases Teóricas
2. Resolución de Ejercicios
3. Proyecto

# Pasos para aprobar

---

1. Realizar las practicas de Laboratorio
2. Aprobar parcial (8<sup>a</sup> semana - 2 recup. )
3. Aprobar proyecto
4. Coloquio integrador

# Proyecto

---

- 6<sup>a</sup>/7<sup>a</sup> semana entrega del anteproyecto
- 8<sup>a</sup> semana comienza el proyecto
- ¿ Que se espera del Proyecto ?



[Te First Microcontroller Project - LED Blinking.flv](#)

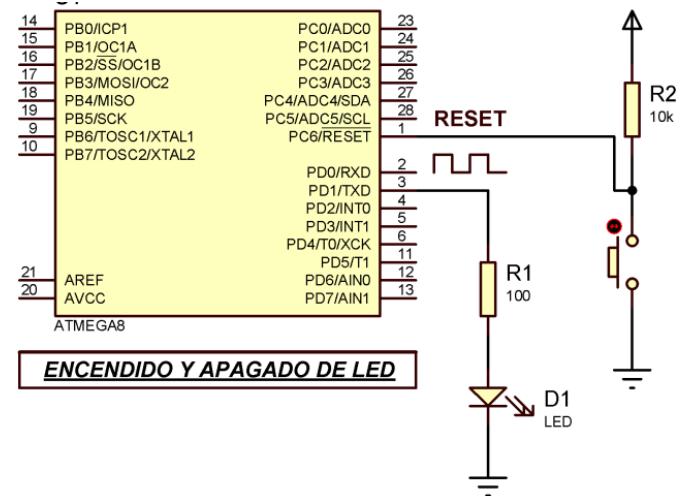


[Humanoid Robot.flv](#)

# Practica a desarrollarse con el microcontrolador



Figura 3: Programador del Club de Robótica[1], derivado del proyecto USBtinyISP[2]



- Armado de un programador para micro controlador de la flia. AVR
- Armado de circuito básico con el micro controlador AVR
- Practica con los Puertos
- Practica con interrupciones
- Practica con temporizadores
- Practica con el Puerto Serial

# Proyecto

---

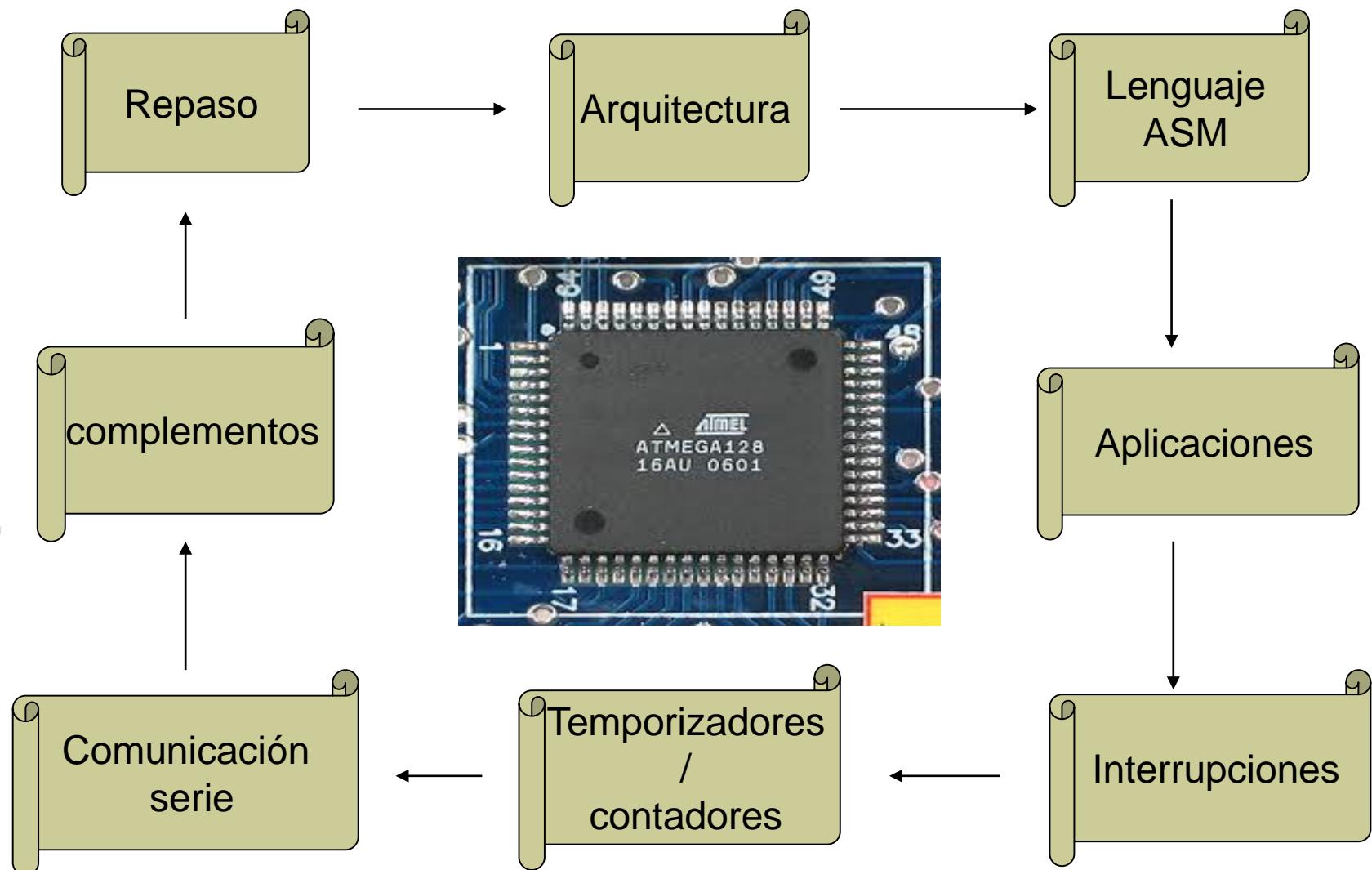
1. 100% funcionando al finalizar el curso
2. Uso de interrupciones / temporizadores /comunicación entre periféricos
3. Y la más *importante que se diviertan en la realización del proyecto*

- **Las transparencias son una guía de estudio**
- **Los Libros dados en la bibliografía junto con lo explicado en las clases teóricas y prácticas representa el material de estudio para el temario de la asignatura**

**Información adicional en :**

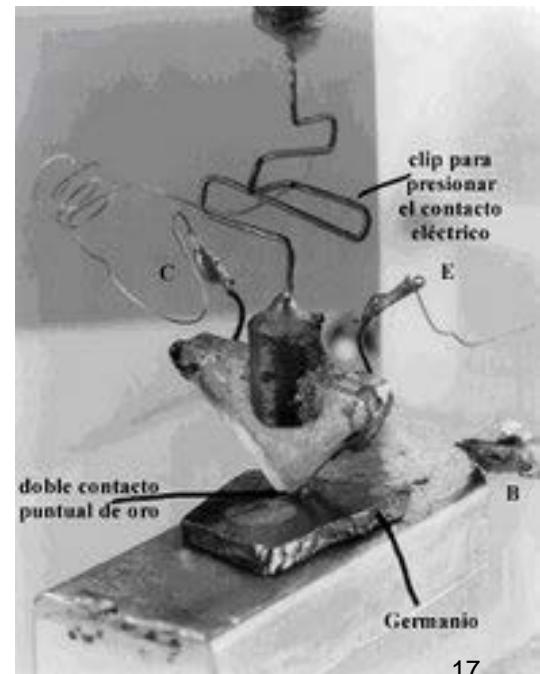
**[materia.6609@gmail.com](mailto:materia.6609@gmail.com)**

# Laboratorio de microcomputadoras



# *primer transistor.*

- *La fecha exacta fue 16 de diciembre de 1947, cuando William Shockley, John Bardeen y Walter Brattain armaron el primer transistor.*

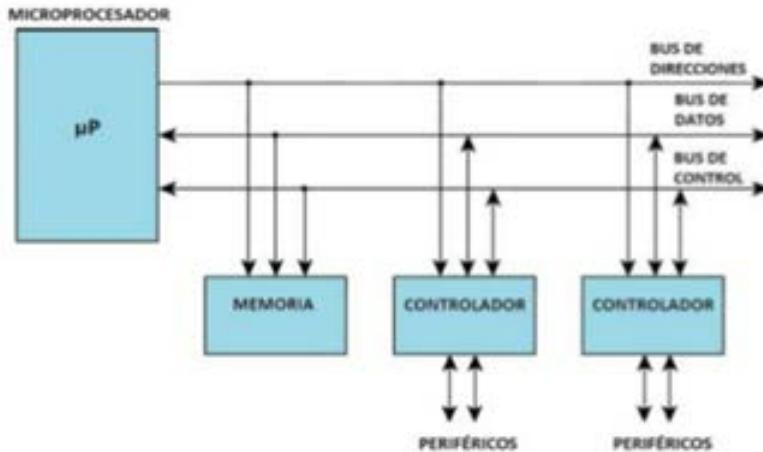


# Qué es un microcontrolador

---

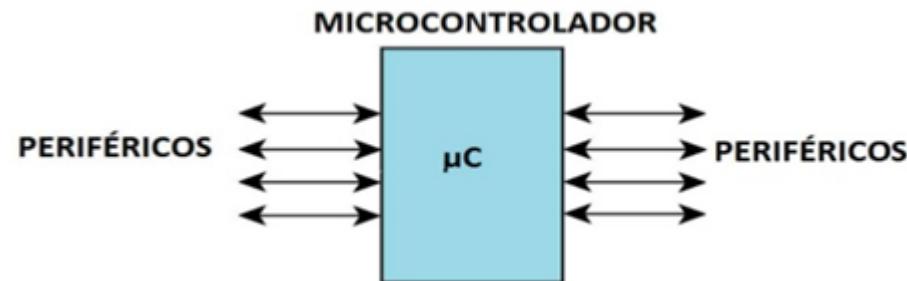
- Fue Intel quien en **1971** diseña el primer microprocesador .
- Verdadera revolución :se pasa de la idea de lógica cableada, donde se hace un diseño digital no modificable, a la de lógica programada, donde el diseño electrónico se combina con software.
- El microcontrolador es la evolución natural del microprocesador. (**1976**)

# DIFERENCIAS ENTRE MICROCONTROLADOR Y MICROPROCESADOR



El microprocesador es un **sistema abierto**, algunas partes se encuentran en el exterior, lo cual le permite configurar a medida de la aplicación

El microcontrolador es un **sistema cerrado**. Todas las partes del procesador están contenidas en su interior y solo salen al exterior la líneas que gobiernan los periféricos.

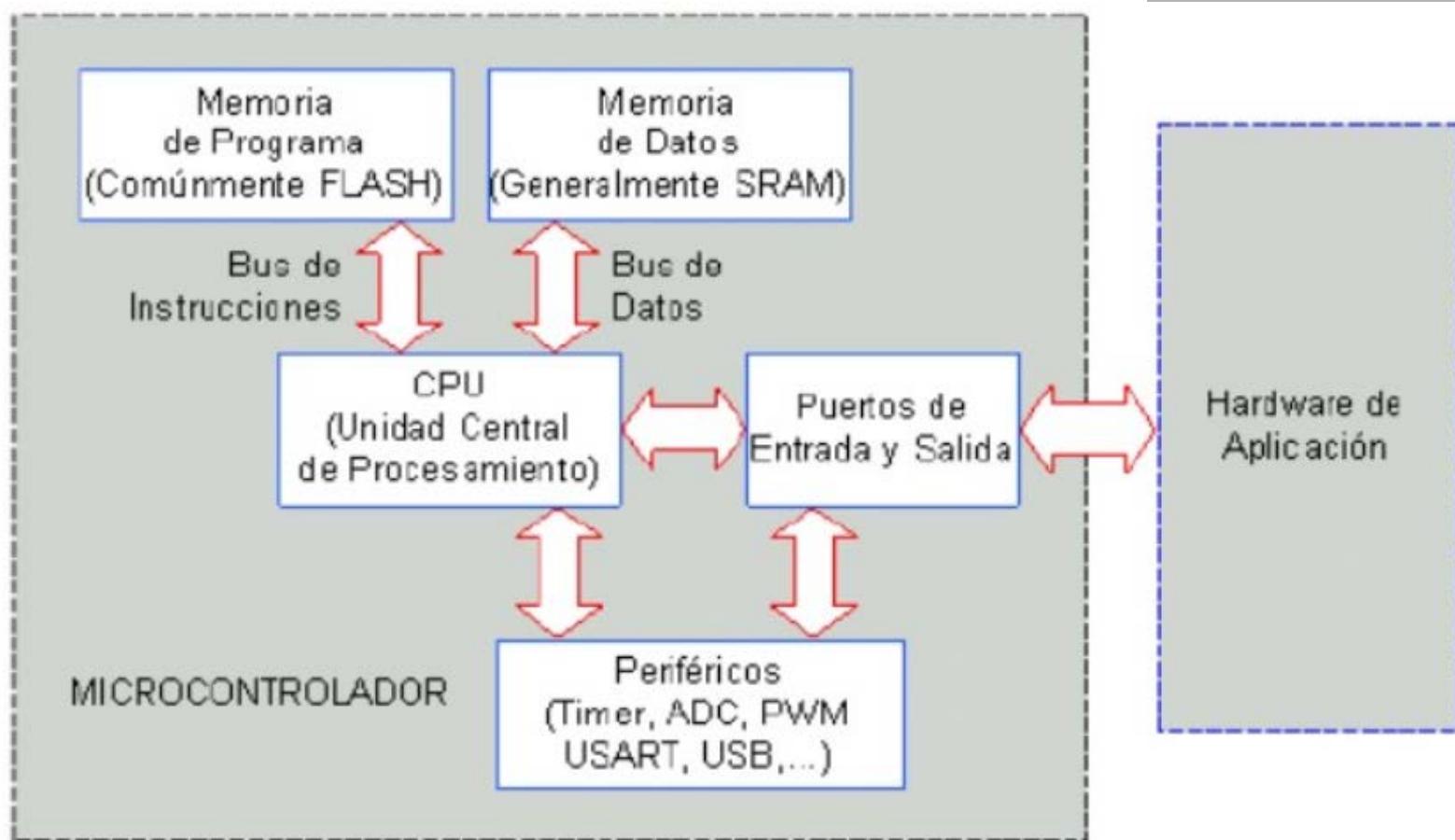


Surgen las primeras microcomputadoras de un sólo chip, que más tarde se denominarán **microcontroladores**. Dos de los primeros microcontroladores, son el 8048 de Intel y el 6805R2 de Motorola.



# Estructura genérica de un microcontrolador

Podemos definir un microcontrolador como un computador completo en un chip.



	<b>Microporcesadores</b>	<b>Microcontroladores</b>
<b>CPU o Procesador</b>	Un microporcesador es una Unidad de Procesamiento Central en un solo chip.	Depende de una Unidad de Procesamiento Central para realizar sus funciones con los datos que obtiene.
<b>Memorias RAM y ROM</b>	Son dispositivos externos que lo complementan para su buen funcionamiento.	Las incluye en un solo circuito integrado.
<b>Velocidad de Operación</b>	Rápida.	Más lenta que un microporcesador.
<b>Tamaño</b>	La configuración mínima básica de un microporcesador está constituida por un microporcesador de 40 pines, una memoria RAM de 28 pines, una memoria ROM de 28 pines y un decodificador de direcciones de 18 pines, lo cual lo convierte en un circuito bastante engorroso y costoso.	El microcontrolador incluye todos estos elementos en un solo circuito integrado por lo que implica una gran ventaja en varios factores, como por ejemplo, la disminución del tamaño del circuito impreso por la reducción de los circuitos externos.
<b>Costos</b>	Para el microporcesador, el costo es muy alto en la actualidad.	El costo para un sistema basado en microcontroladores es mucho menor.
<b>Interferencias</b>	Son más susceptibles a la interferencia electromagnética debido a su tamaño y a su cableado externo que lo hace más propenso al ruido.	El alto nivel de integración reduce los niveles de interferencia electromagnética.

## *Microprocesador*

Es un chip (circuito integrado) que procesa o manipula la información almacenada en la memoria (datos y/o instrucciones), requiere de periféricos adicionales para su operación, tiene muy altas prestaciones y capacidades, de precio elevado y tamaño considerable.

- Es un sistema digital abierto
- Requiere de RAM externa de gran capacidad (Mbytes)
- Memoria de programa muy extensa
- Requiere un decodificador por dispositivo y por puerto externo.
- Velocidades de operación de hasta Giga/Tera MIPS
- Son de uso y propósito general
- Costo de desarrollo e implementación elevado
- Gran dificultad para el desarrollo / implementación
- Tendencia a la centralización de las tareas
- Capacidad de realizar multitareas no limitadas
- Interrupciones Vectorizadas
- Muchos modos de direccionamiento
- Casi todos son **CISC**
- Tiempo de aprendizaje/desarrollo elevado
- Para el Año 2002 se habían vendido **1.000 Millones pzas.**



## *Microcontrolador*



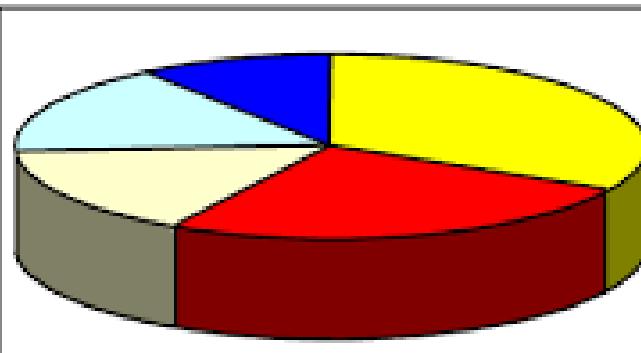
Es un chip (Circuito integrado) que contiene un computador completo junto a diversos recursos auxiliares con prestaciones y capacidades limitadas, de muy bajo costo y tamaño reducido.

- Es un sistema digital cerrado
- RAM de poca capacidad interna (bytes o Kb)
- Memoria de programa pequeña (algunos Kb)
- No se requiere de decodificadores de E/S o de memoria
- Velocidad de operación de algunos MIPS
- Orientados a operaciones de E/S
- Costo de desarrollo e implementación reducido
- Muy fáciles y sencillos de utilizar
- Circuitos simples y tendencia a la modularidad
- Por lo general dan cabida a un sólo programa y el soporte de tareas múltiples (de existir), es limitado
- Dos o tres modos de direccionamiento
- Casi todos son **RISC**
- Tiempo de aprendizaje/desarrollo reducido  
Para el Año 2002 se habían vendido **49.000 Millones pzas.**

- Aplicaciones de los microcontroladores

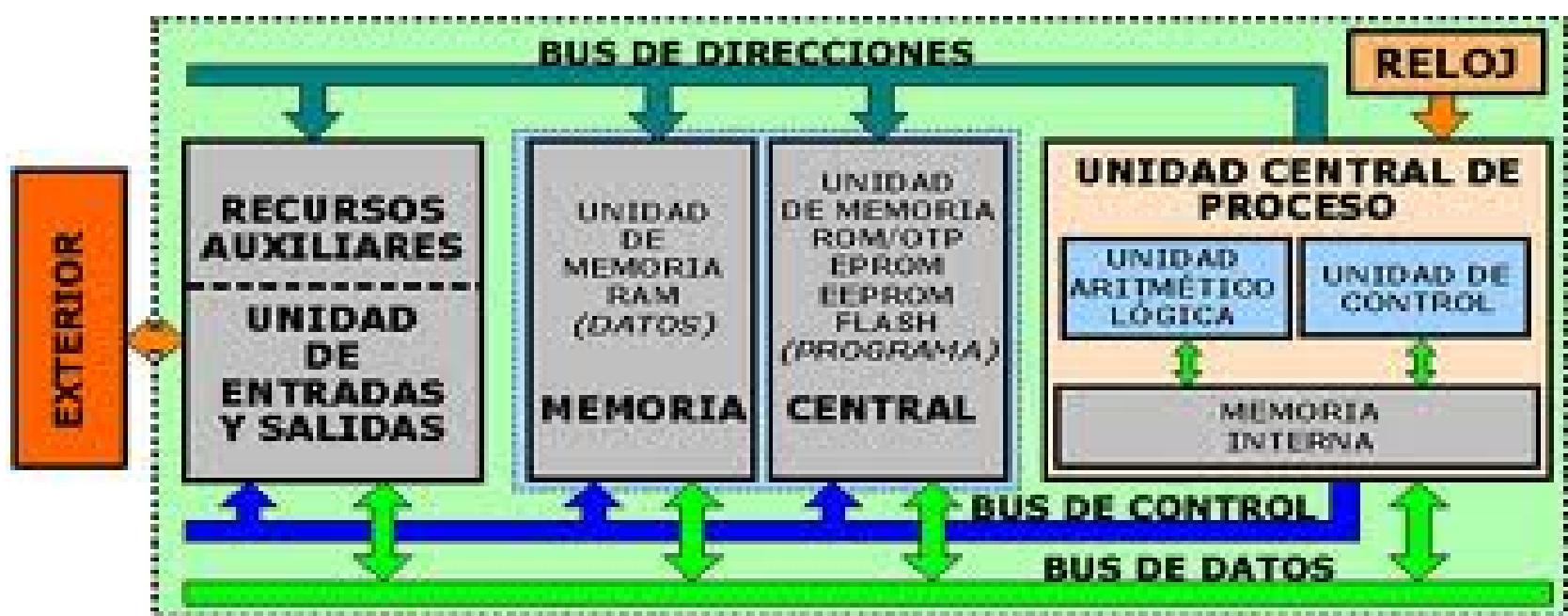
- Robótica: Muy usados en subsistemas específicos de control (extremidades, facciones del rostro, soportes prensiles, etc.)
- Equipamiento informático: impresoras, scanners, copiadoras...
- Sistemas portátiles y autónomos
- Sector automotriz: control centralizado de puertas y ventanas, climatizadores, inyección, alarmas, etc.
- Sector doméstico: integrado en los sistemas de televisores, lavarropas, microondas, heladeras, videos, etc.

Reparto del uso de microcontroladores en las áreas de mayor difusión



- 33% Industria informática.
- 25% Electrodomésticos, Juegos, TV, Video, etc.
- 16% Sistemas de comunicación.
- 16% Industria.
- 10% Automoción.

# Microcontrolador



- Genera señales de control
- La **unidad central de procesamiento** interpreta las instrucciones contenidas en los programas y procesa los datos

# Clasificaciones típicas de los microcontroladores

---

Las tres clasificaciones principales son:

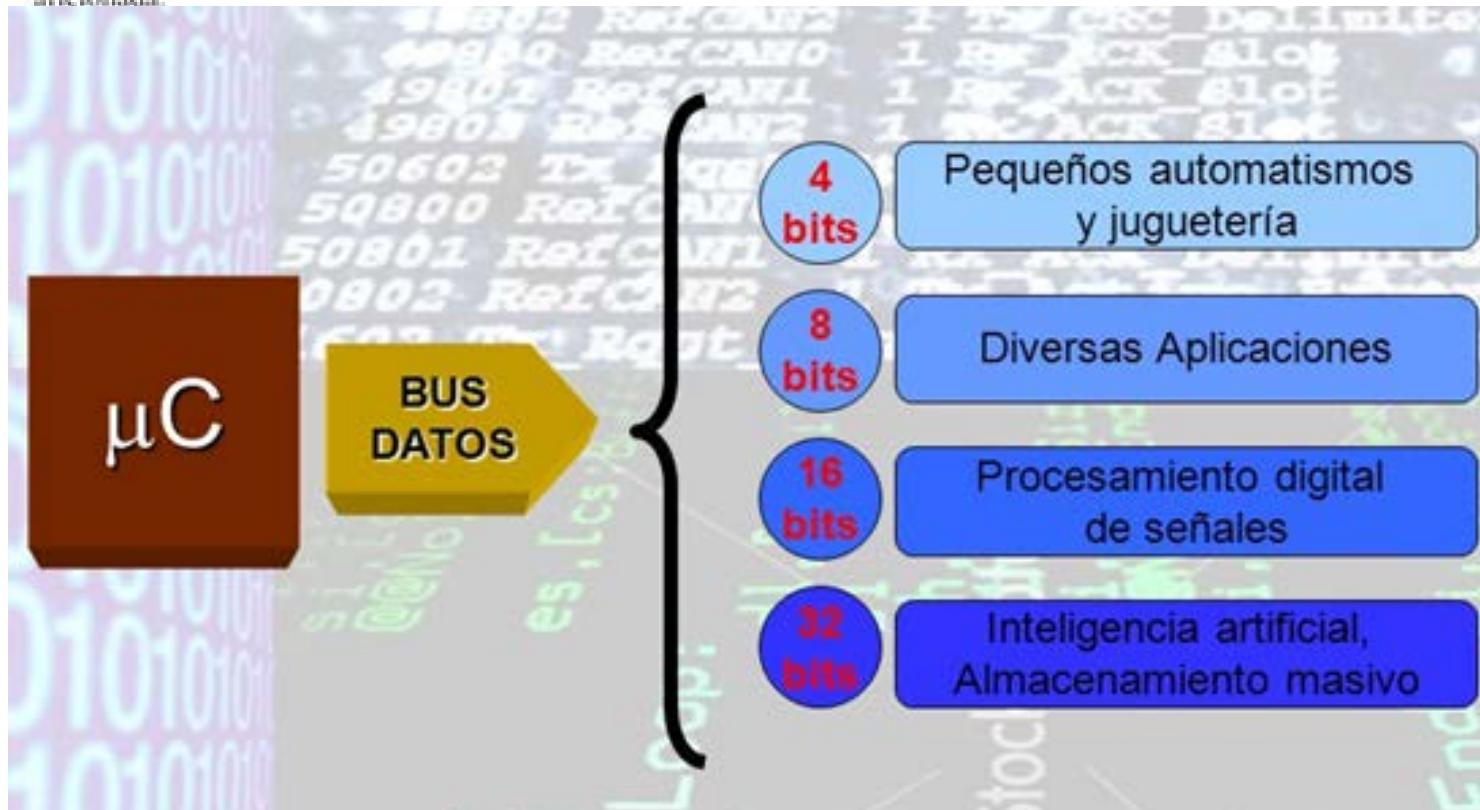
- ✓ Por el tamaño de palabra
- ✓ Por el tipo de juego de instrucciones máquina
- ✓ Por el modo de acceso a los programas y dato

En cualquier caso, los microcontroladores actuales suelen mezclar internamente varios de estos conceptos, por lo que es muy difícil clasificarlo absolutamente.

# Por el tamaño de palabra

- Microcontroladores de 8 bits, de 16 bits o de 32 bits

A mayor longitud del bus de datos, mayor será la eficiencia del microcontrolador en operaciones con datos grandes. Pero al mismo tiempo la complejidad del chip y por ende su costo también aumentarán. Por ejemplo, para los dispositivos multimedia, que procesan datos de video y audio, un bus de datos de 8 bits sería insuficiente. Hoy en día los microcontroladores de 8 bits encuentran mayor aplicación y están, de lejos, mucho más difundidos en aplicaciones de la gente aficionada.



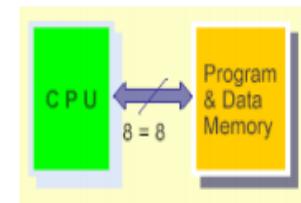
# Acceso a la Memoria de programa y de datos

## Arquitecturas



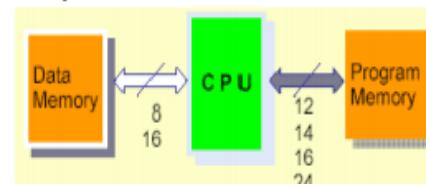
### Von Neumann

- Un único bus de datos para instrucciones y datos.
- Las instrucciones del programa y los datos se guardan conjuntamente en una memoria común.
- Cuando la CPU se dirige a la memoria principal, primero accede a la instrucción y después a los datos necesarios para ejecutarla, esto retarda el funcionamiento.



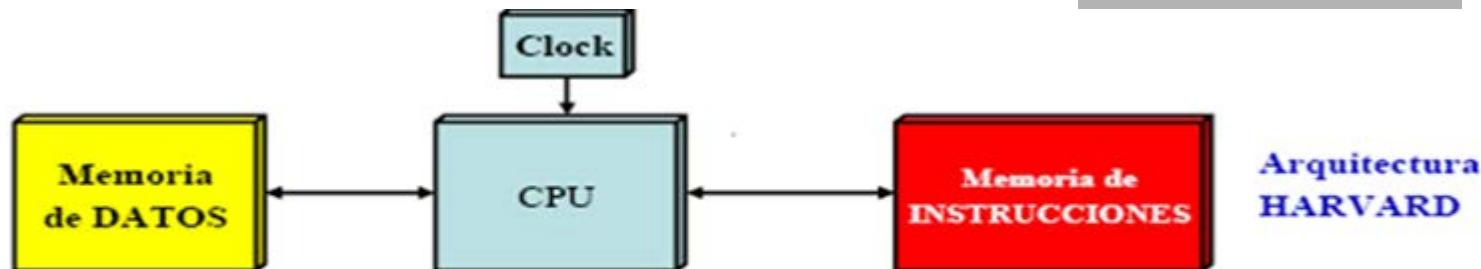
### Hardvard

- El bus de datos y el bus de instrucciones están separados
- Acceso en paralelo:
  - Cuando se está leyendo una instrucción, la instrucción actual está utilizando el bus de datos. Una vez finalizada la instrucción actual, la siguiente ya está disponible en la CPU.
  - Permite una ejecución más rápida.



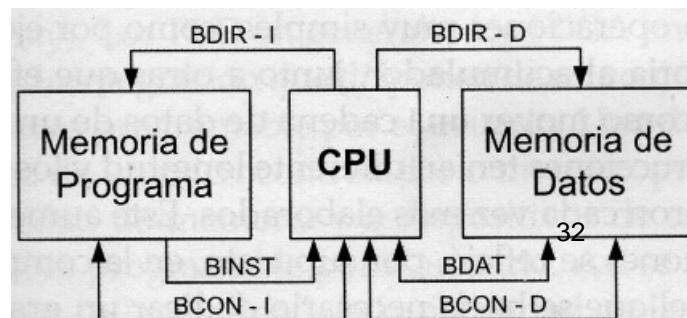
# Tipos de arquitecturas

## HARVARD



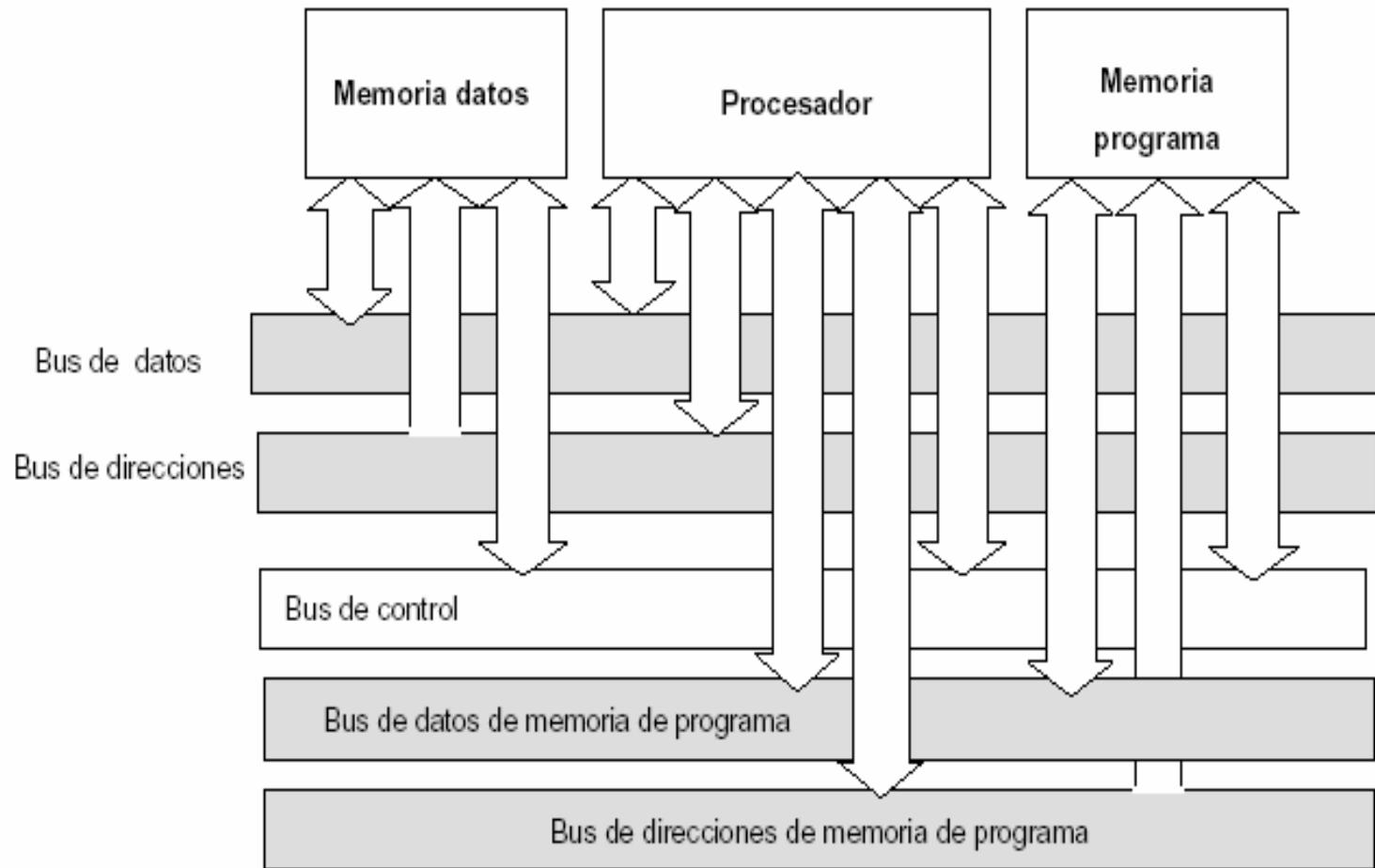
**Arquitectura Harvard** Dispone de dos memorias independientes, una que contiene sólo instrucciones, y otra que contiene sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

El 8051<sup>AVR</sup>, y los PIC's tienen este tipo de Arquitectura.



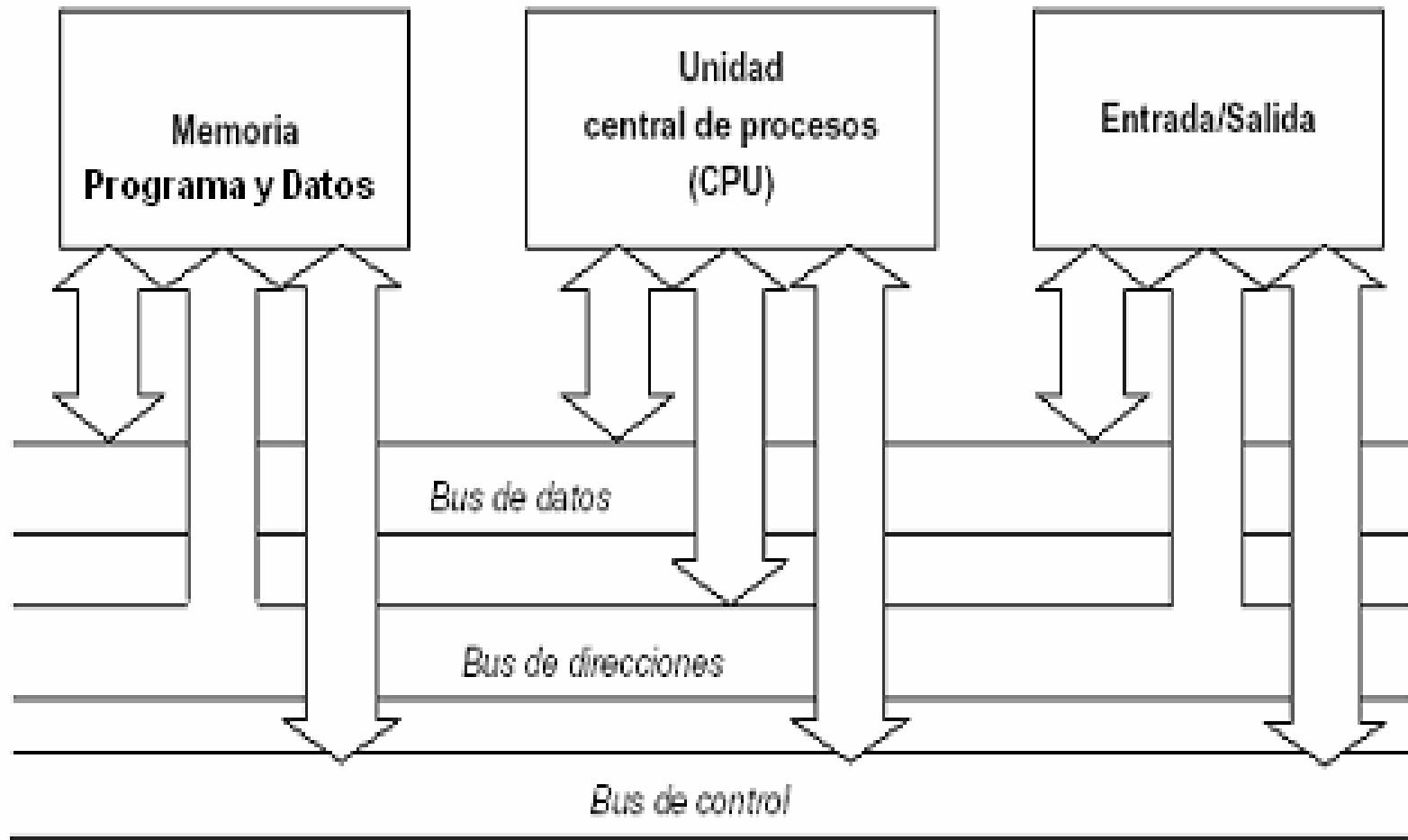
# Tipos de arquitecturas

## HARVARD

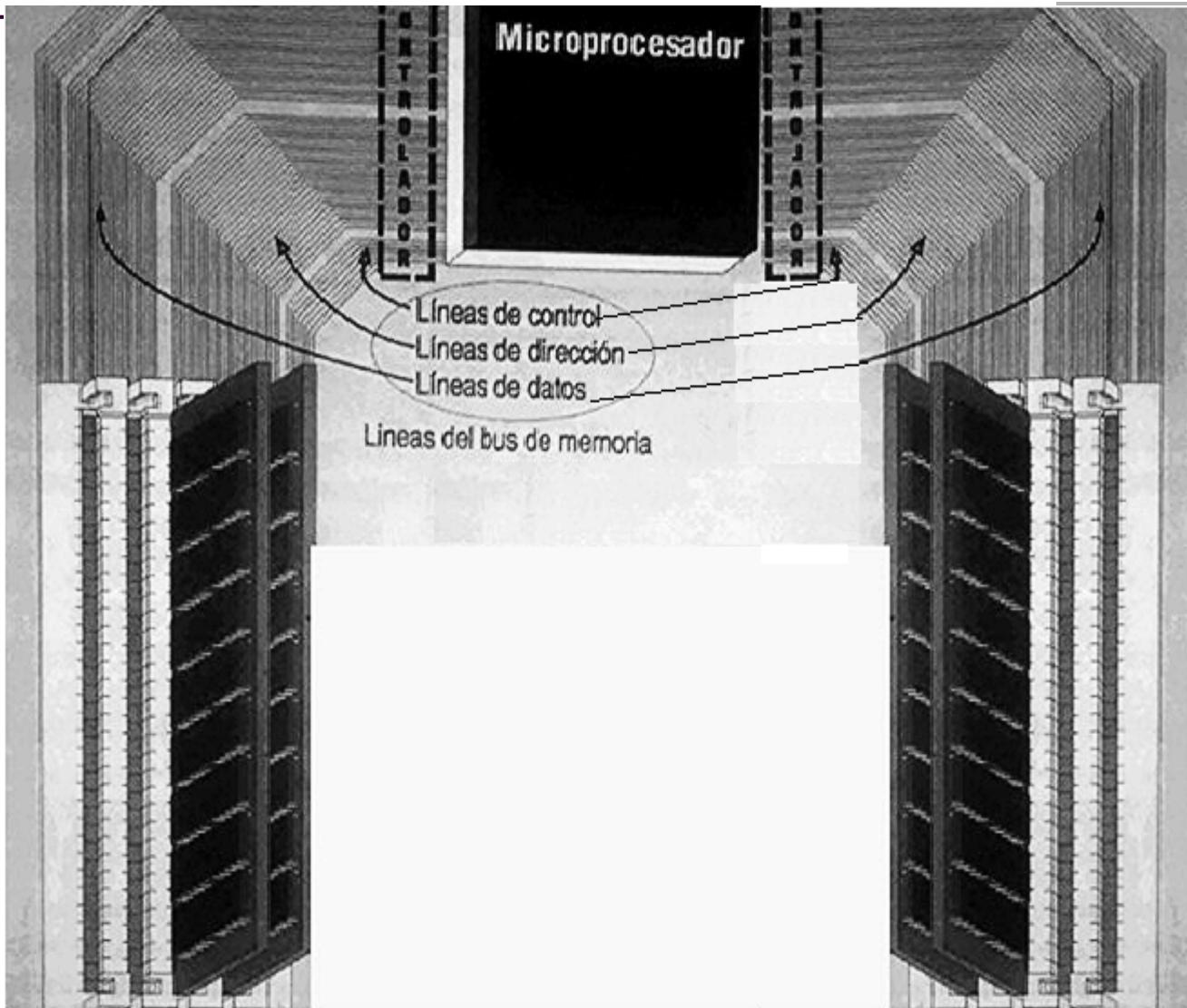


# Tipos de arquitecturas

## Von Neuman



# Tipos de arquitecturas



# Arquitecturas CISC y RISC

juego de instrucciones máquina

- La meta principal es incrementar el rendimiento del procesador.
- Para esto se deben considerar tres áreas principales.

## 1. La tecnología de proceso.

## 2. El encapsulado.

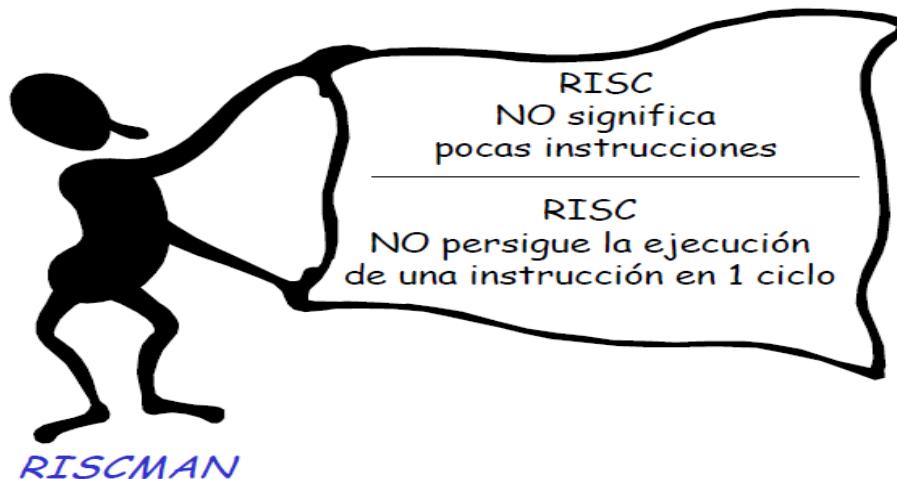
## 3. La arquitectura.

- Memoria
- Dispositivos de E/S
- El bus
- La CPU

- **La tecnología de proceso**, se refiere a los materiales y técnicas utilizadas en la fabricación del circuito integrado.
- **El encapsulado** se refiere a cómo se integra un procesador con lo que lo rodea en un sistema funcional.
- Es la **arquitectura** del procesador lo que hace la diferencia en el rendimiento de una CPU

# Arquitecturas CISC y RISC

- Significado de los términos CISC y RISC:
- **CISC** (complex instruction set computer)
- **RISC** (reduced instruction set computer) .



# Evolución de los CISC

Al principio, la memoria era costosa



Objetivo:

Minimizar los programas (en memoria)

La microprogramación facilita la construcción de funciones más complejas

Lenguaje máquina más potente (y complejo)

Los programas constan de menos instrucciones

→ Ocupan menos memoria

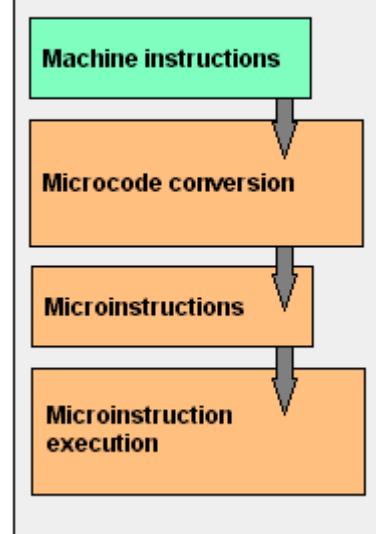
→ Menos tiempo de acceso a memoria

ADEMÁS, se acorta la distancia con los lenguajes de alto nivel



Es más fácil construir compiladores

CISC



## PERO:

- ✓ El microprograma crece ( y se complica)
- ✓ Crecimiento por compatibilidad de familias
- ✓ Muchas instrucciones
  - Muchos formatos
  - Instrucciones de longitud variable

# CISC

---

- Computador con repertorio de instrucciones complejo.
- Gran número de instrucciones complejas.
- Gran variedad de tipos de datos y de modos de direccionamiento.
- Permite implementar instrucciones de alto nivel directamente o con un número pequeño de instrucciones ensamblador..

- A mediados de la década de los setentas, con respecto a la frecuencia de utilización de una instrucción en un CISC y al tiempo para su ejecución, se observó lo siguiente
- 20% de las instrucciones ocupa el 80% del tiempo total de ejecución
- Existen secuencias de instrucciones simples que obtienen el mismo resultado que secuencias complejas predeterminadas

Tiempo de Ejecución  
de un Programa

$$I \times C \times T_c$$

Dos opciones  
para mejorar el rendimiento

CISC

Minimizar I  
(incrementando C y Tc)

RISC

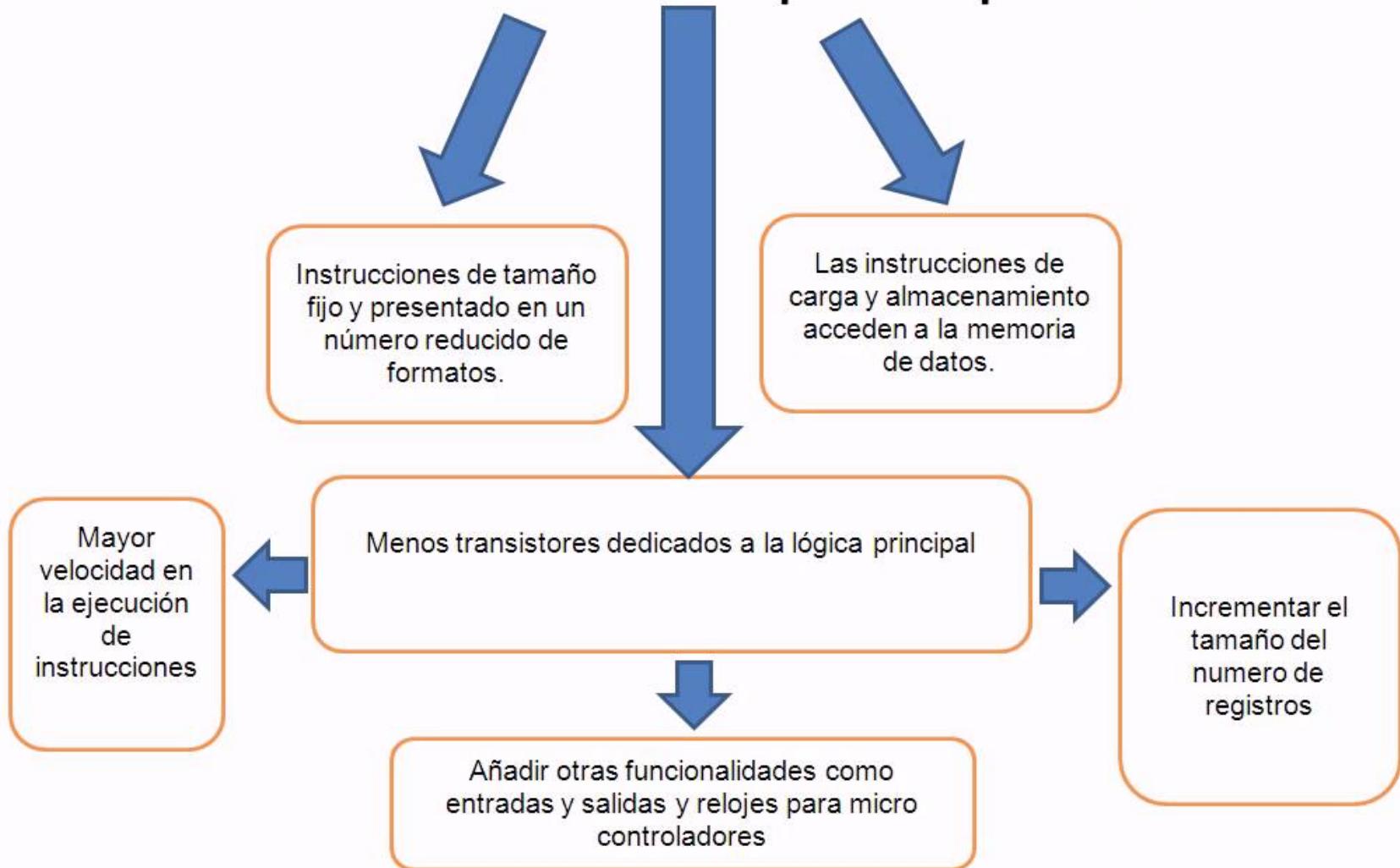
Minimizar C y Tc  
(Incrementando I)

# Arquitectura RISC

---

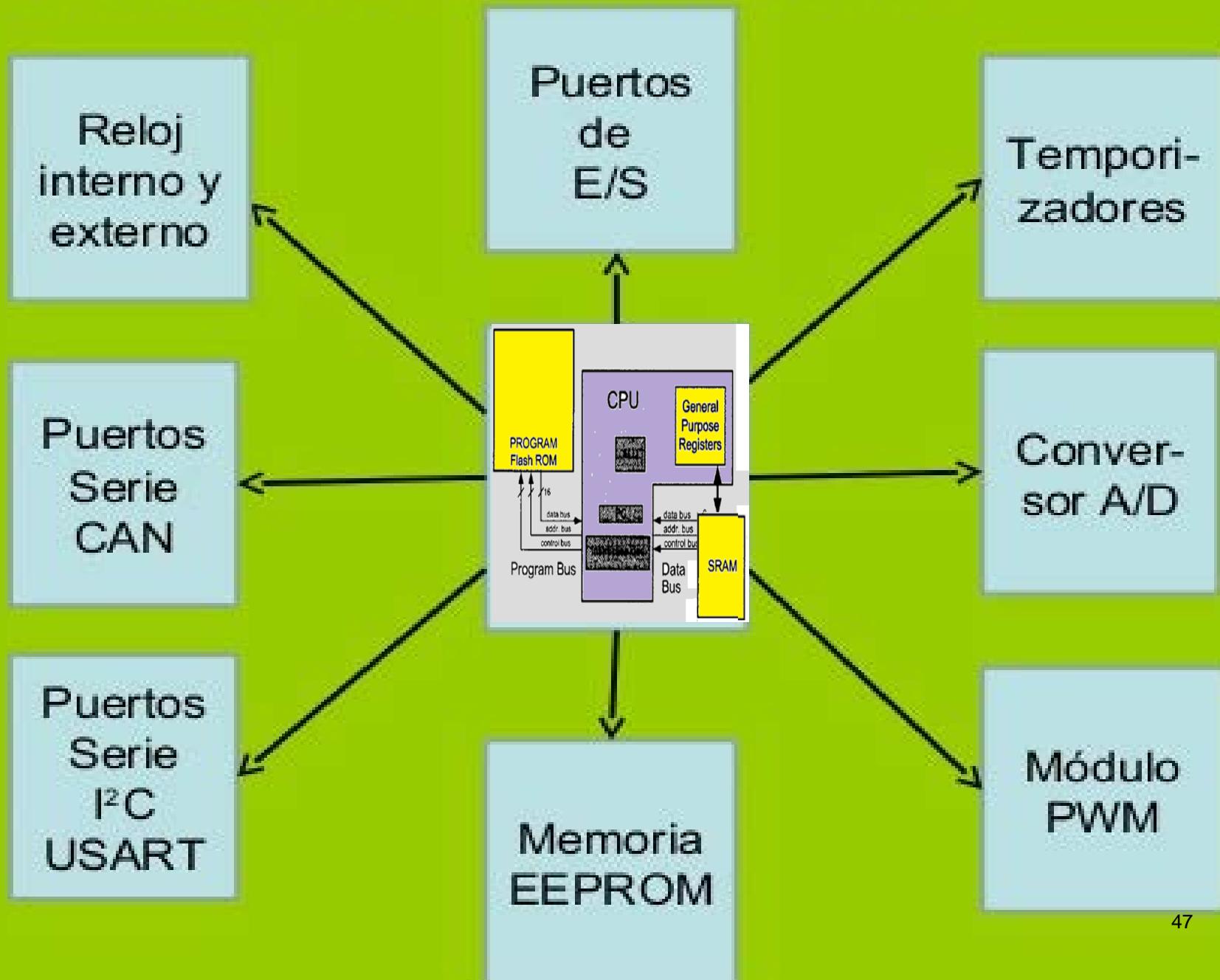
**La arquitectura RISC funciona de modo muy diferente a la Arq. CISC su objetivo no es ahorrar esfuerzos externos por parte del software con sus accesos a la memoria . RAM sino facilitar que las instrucciones sean ejecutadas lo mas rápidamente posible**

# Características principales

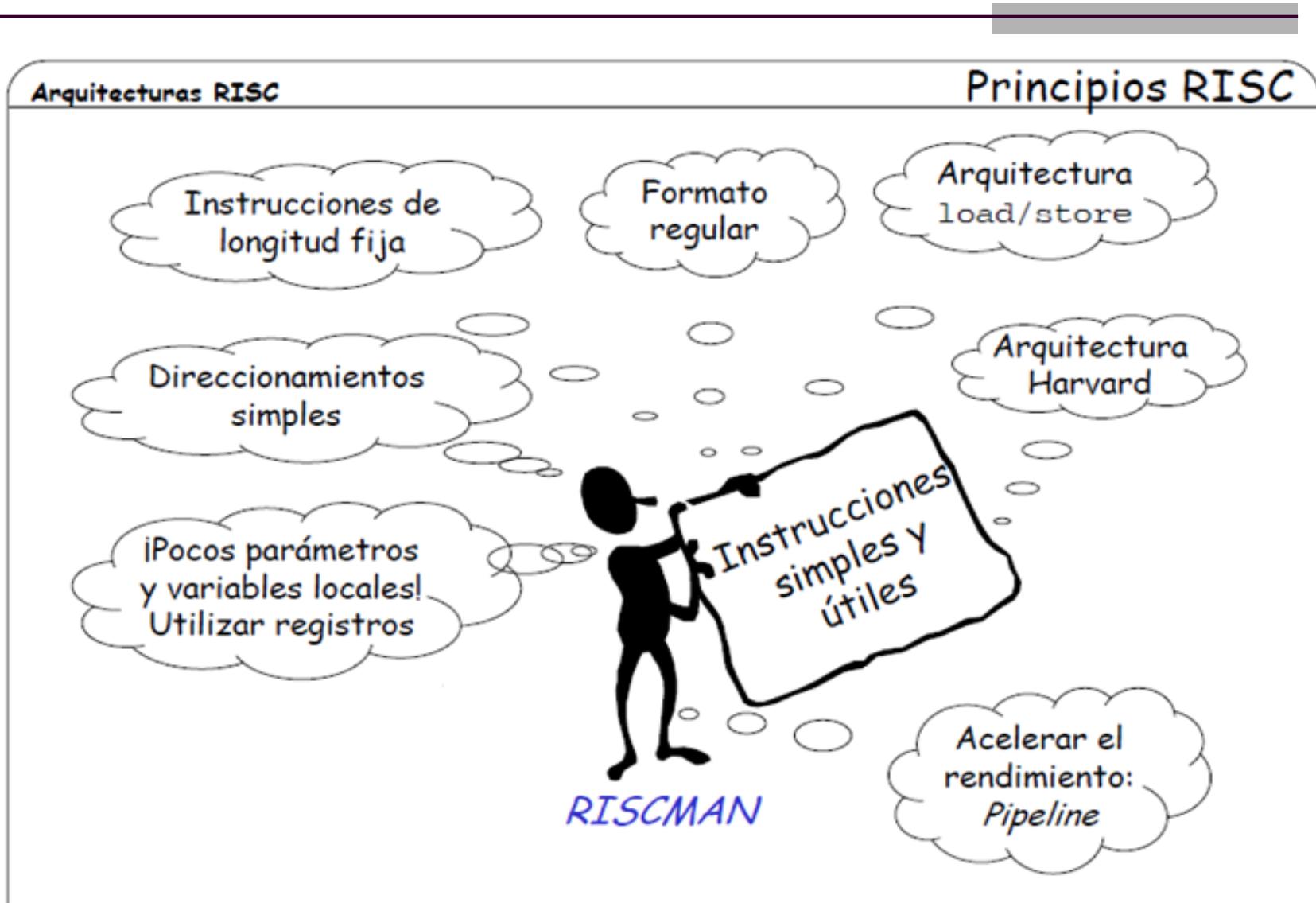


- El hecho de que la estructura simple de un procesador RISC conduzca a una notable reducción de la superficie del circuito integrado, se aprovecha con frecuencia para ubicar en el mismo, funciones adicionales

# *Periféricos de un microcontrolador*



El hecho de que la estructura simple de un procesador RISC conduzca a una notable reducción de la superficie del circuito integrado, se aprovecha con frecuencia para ubicar en el mismo, funciones adicionales



# Arq. RISC características

---

- **Reducción del conjunto de instrucciones** a instrucciones básicas simples, con la que pueden implantarse todas las operaciones complejas.
- **Arquitectura del tipo load-store** (carga y almacena). Las únicas instrucciones que tienen acceso a la memoria son 'load' y 'store'; registro a registro, con un menor número de acceso a memoria.
- **Casi todas las instrucciones pueden ejecutarse dentro de un ciclo de reloj.** Con un control implantado por hardware (con un diseño del tipo load-store), casi todas las instrucciones se pueden ejecutar cada ciclo de reloj, base importante para la reorganización de la ejecución de instrucciones por medio de un compilador.
- **Pipeline** (ejecución simultánea de varias instrucciones). Posibilidad de reducir el número de ciclos de máquina necesarios para la ejecución de la instrucción, ya que esta técnica permite que una instrucción pueda empezar a ejecutarse antes de que haya terminado la anterior.

# Arq. RISC características

---

- Repertorio simple y ortogonal.
- Formatos de instrucción uniformes.
- Pocos tipos de datos y de modos de direccionamiento, ~~sistemas~~ siempre los más sencillos.

Una instrucción por circunstancia.

— Instrucciones complejas.

Lenta ejecución de la instrucción.

Pocas instrucciones por trabajo

Pequeñas instrucciones.

Instrucciones simples.

Ejecución rápida de las instrucciones.

Varias instrucciones por trabajo específico.

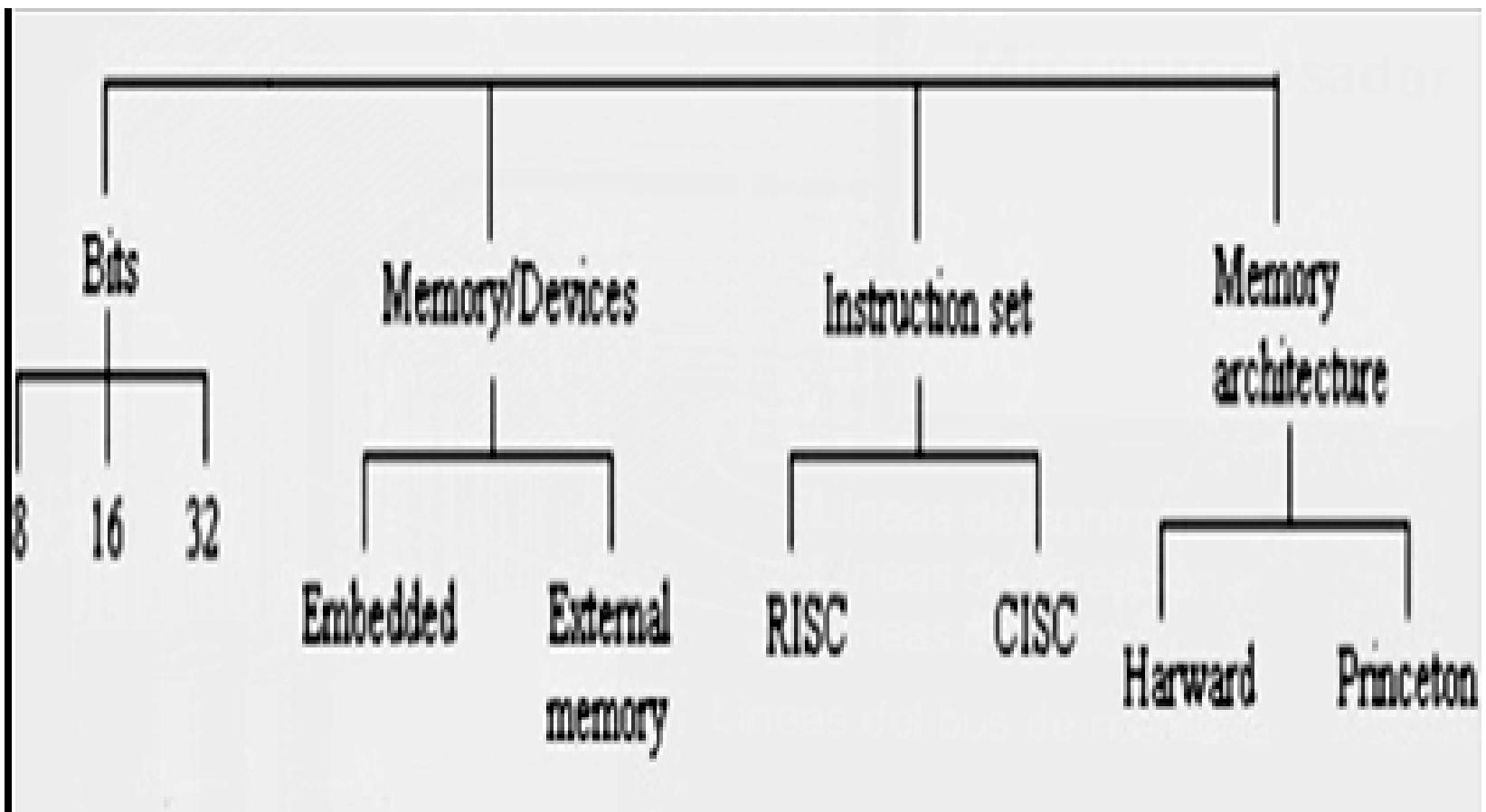
CISC



RISC



# Microcontroladores



# Microcontroladores

**Table 1-6: Some of the Companies that Produce Widely Used 8-bit Microcontrollers**

Company	Web Site	Architecture
Atmel	<a href="http://www.atmel.com">http://www.atmel.com</a>	AVR and 8051
Microchip	<a href="http://www.microchip.com">http://www.microchip.com</a>	PIC16xxx/18xxx
Intel	<a href="http://www.intel.com/design/mcs51">http://www.intel.com/design/mcs51</a>	8051
Philips/Signetics	<a href="http://www.semiconductors.philips.com">http://www.semiconductors.philips.com</a>	8051
Zilog	<a href="http://www.zilog.com">http://www.zilog.com</a>	Z8 and Z80
Dallas Semi/Maxim	<a href="http://www.maxim-ic.com">http://www.maxim-ic.com</a>	8051
Freescale Semi	<a href="http://www.freescale.com">http://www.freescale.com</a>	68HC11/HCS08

See <http://www.microcontroller.com> for a complete list.

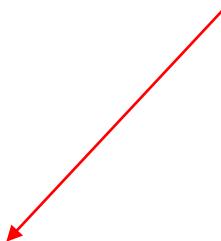
# ATMEL-AVR

---

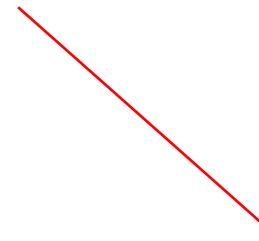
**Microcontrolador**



**Arquitectura basada**



**8-bit- RISC**



**Harvard**

# COMPUTADORA DIGITAL

INSTRUCCIONES

SISTEMA  
DE RELOJ

UNIDAD DE  
MEMORIA

DATOS Y  
RESULTADOS

UNIDAD  
DE  
CONTROL

UNIDAD DE  
ARITMÉTICA Y  
LÓGICA

SISTEMA DE  
ENTRADA Y  
SALIDA

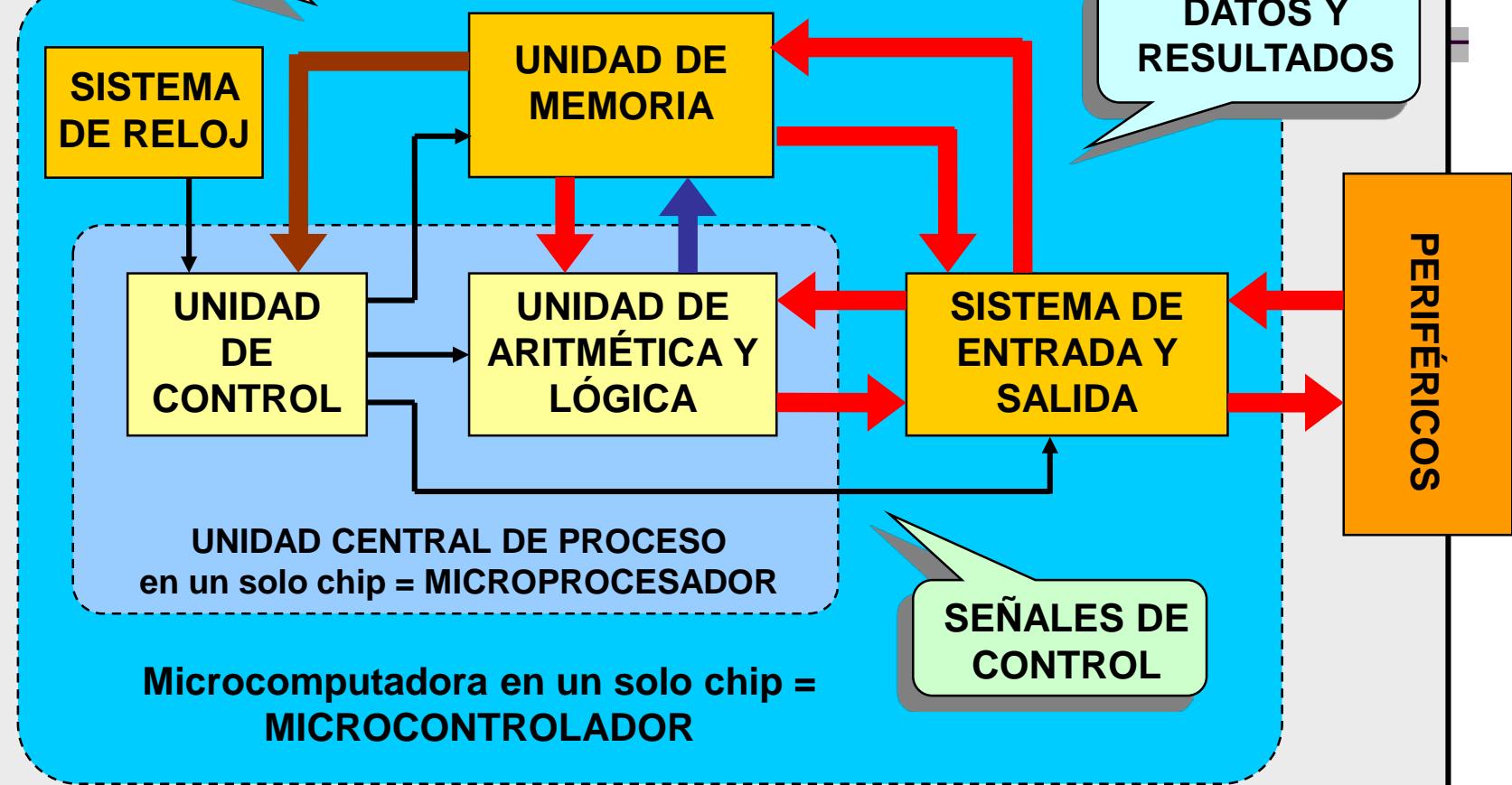
PERIFÉRICOS

SEÑALES DE  
CONTROL

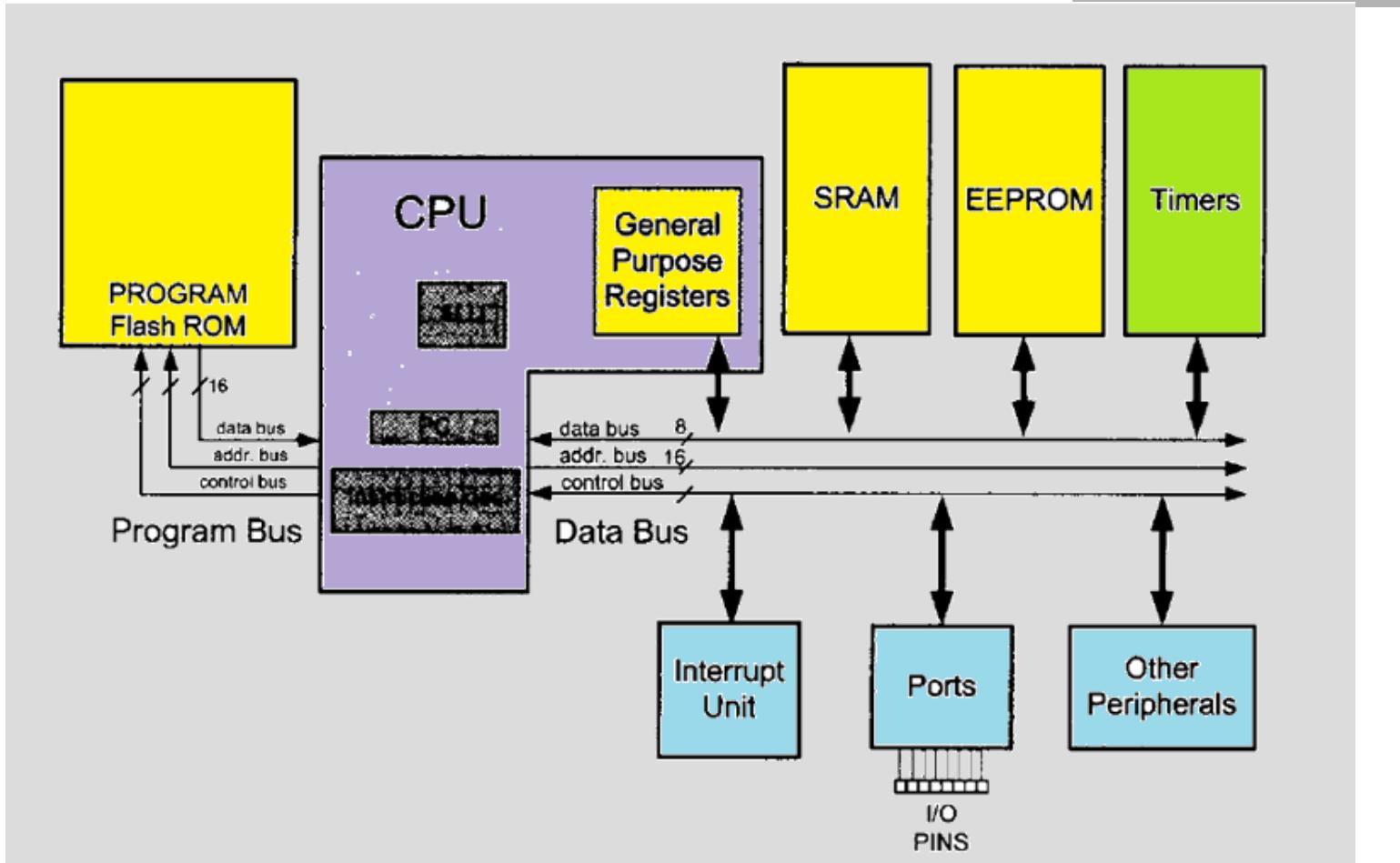
UNIDAD CENTRAL DE PROCESO  
en un solo chip = MICROPROCESADOR

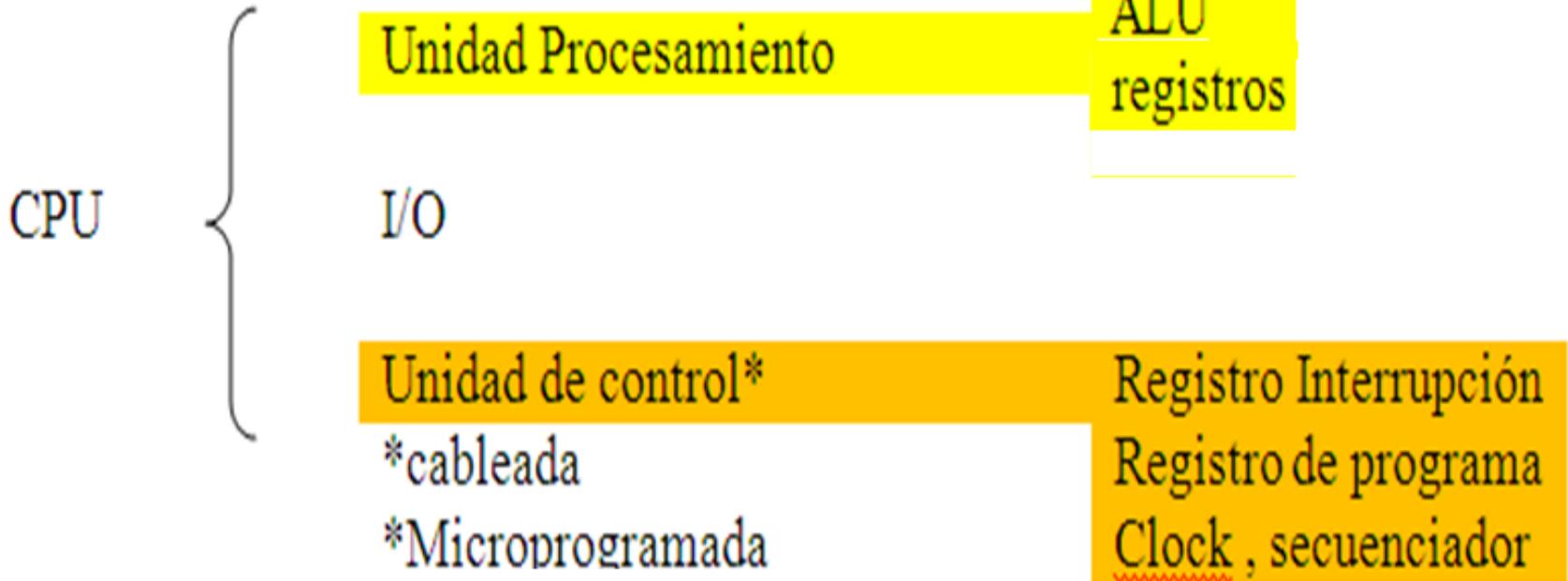
# COMPUTADORA DIGITAL

INSTRUCCIONES

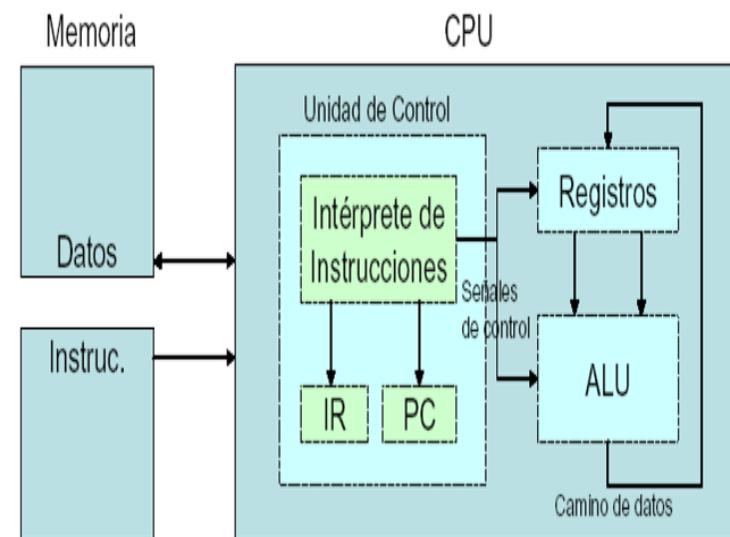


# ATMEL-AVR

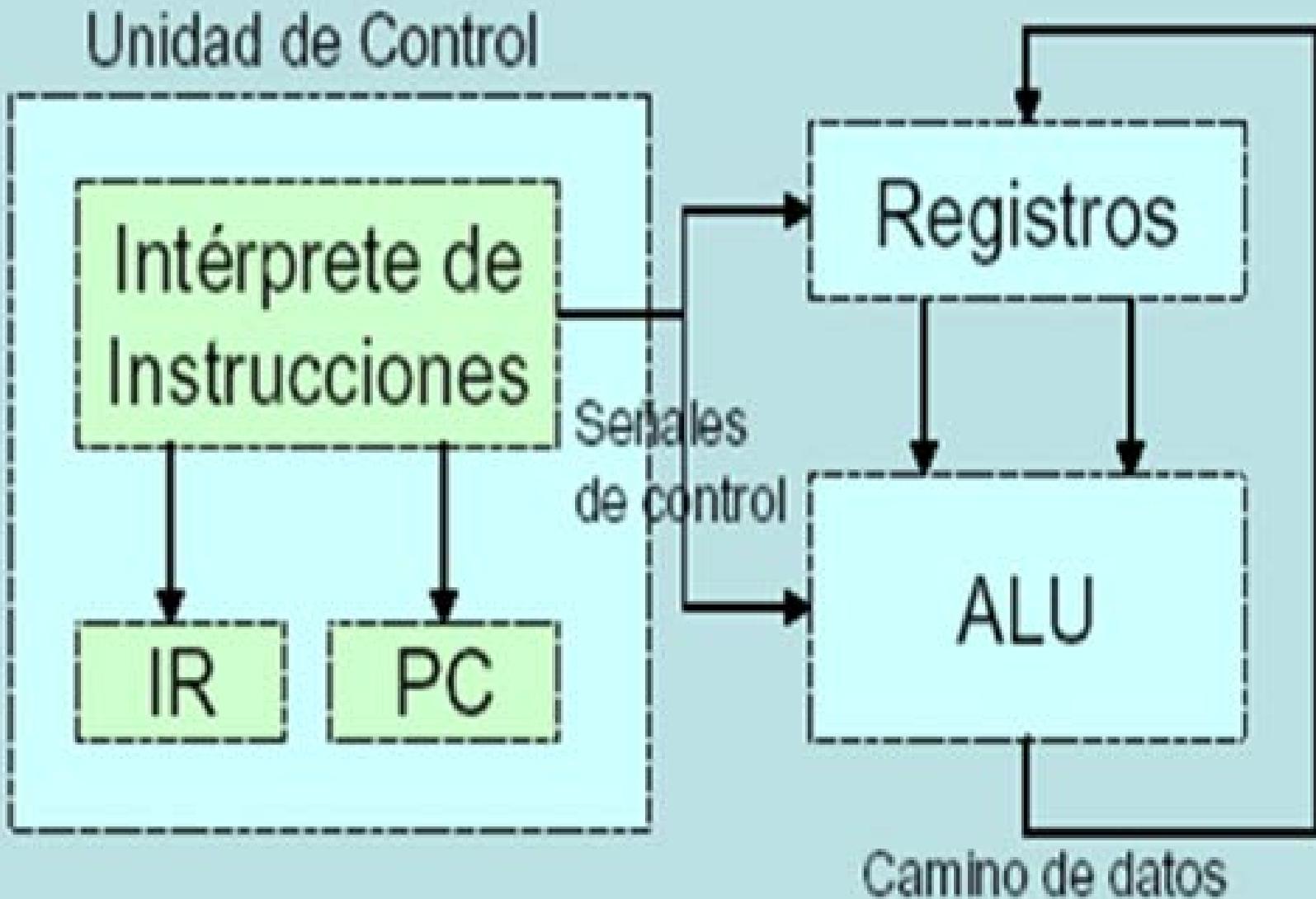




## ■ Bus ■ Memorias ■ I/O



# CPU



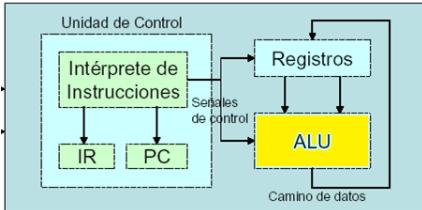
# Unidad Lógica Aritmética

---

## ALU

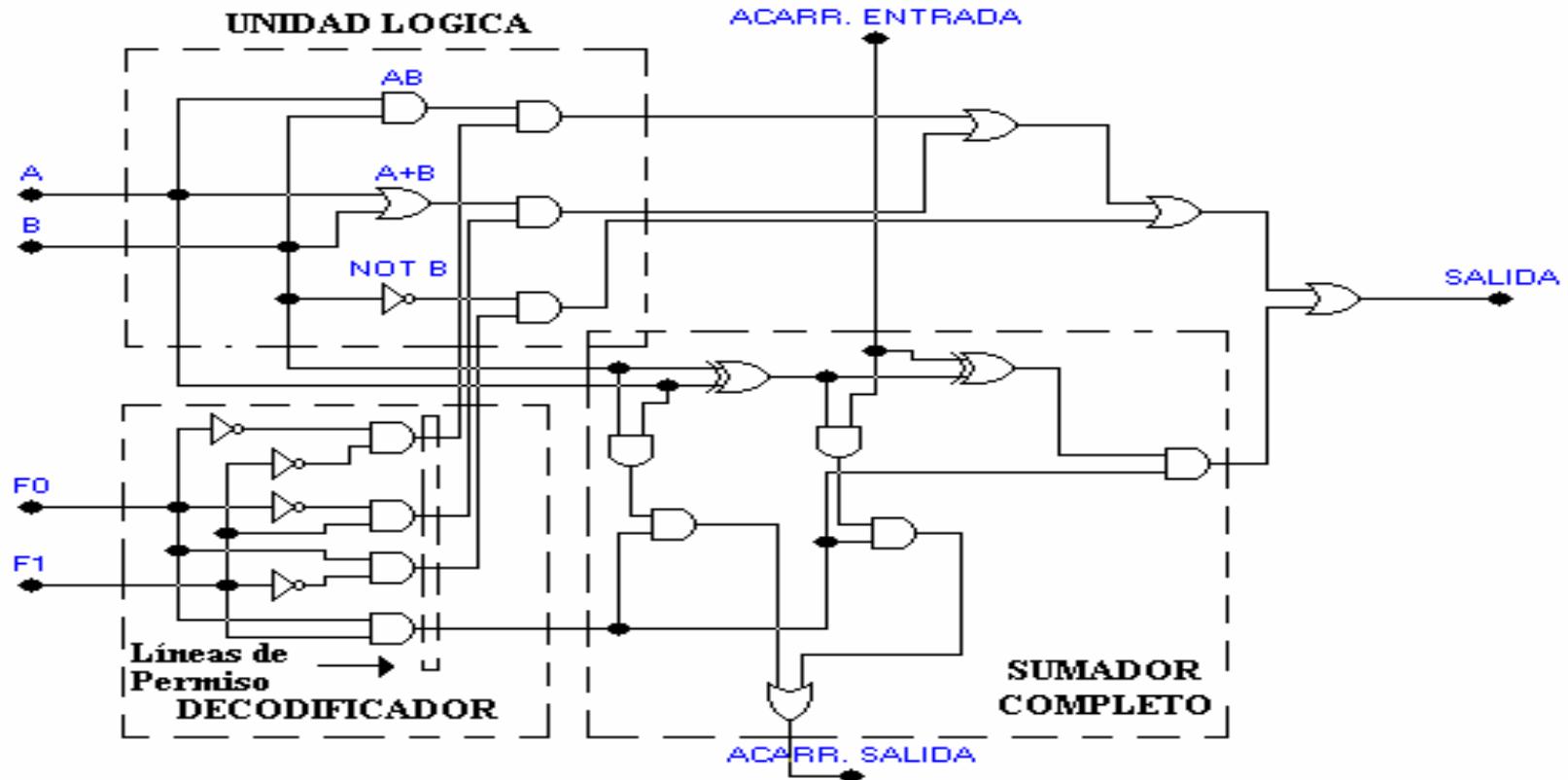
Desempeña funciones aritméticas /lógicas

- Suma resta
- Lógicas ( AND, OR)

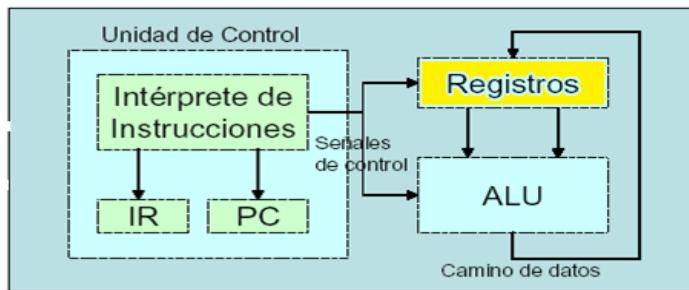


# UNIDAD ARITMÉTICA LÓGICA

GENERIC



■ **UNIDAD DE ARITMÉTICA Y LÓGICA** :Conjunto de circuitos digitales que realizan las operaciones tanto lógicas como aritméticas.

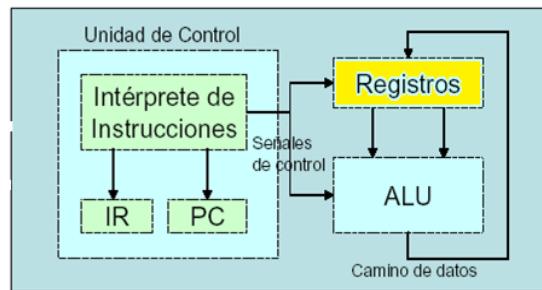


# Registros

- Direcciones de la memoria (MAR)
- De datos (  $R_0, \dots, R_{31}$  )
- De instrucciones (IR)
- SREG ( Estado de registro )
- ■ PSW ( Palabra de estado de programa )

Bit	D7								D0
SREG	I	T	H	S	V	N	Z	C	

C – Carry flag	S – Sign flag
Z – Zero flag	H – Half carry
N – Negative flag	T – Bit copy storage
V – Overflow flag	I – Global Interrupt Enable



# Registros

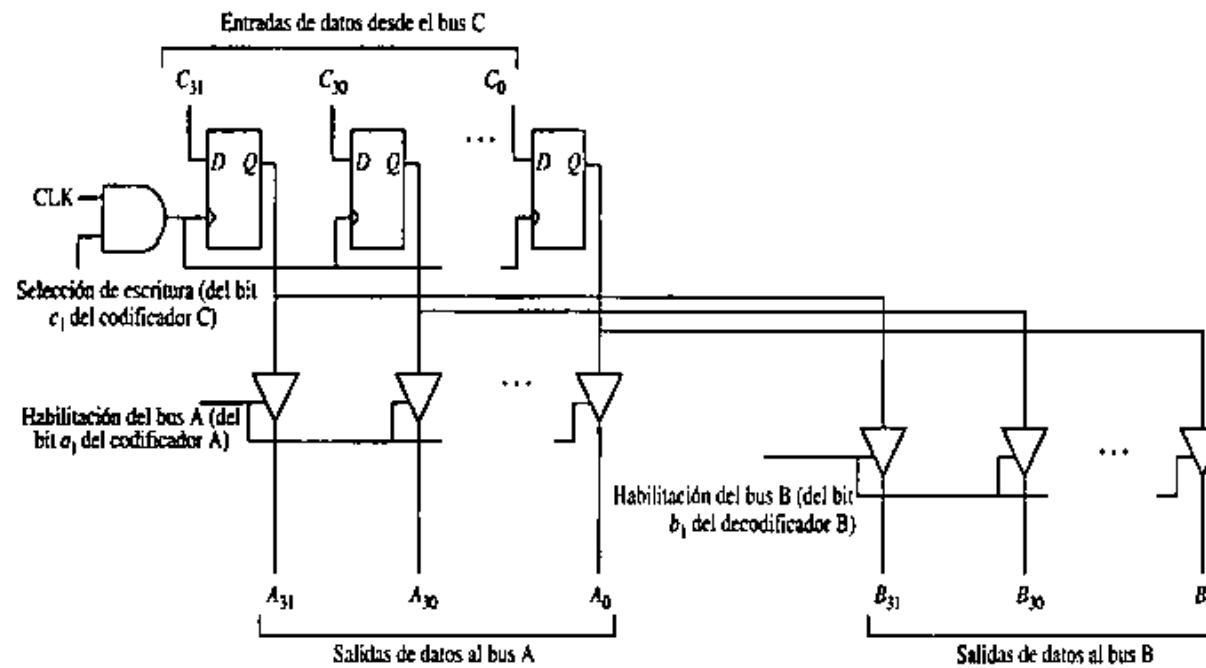
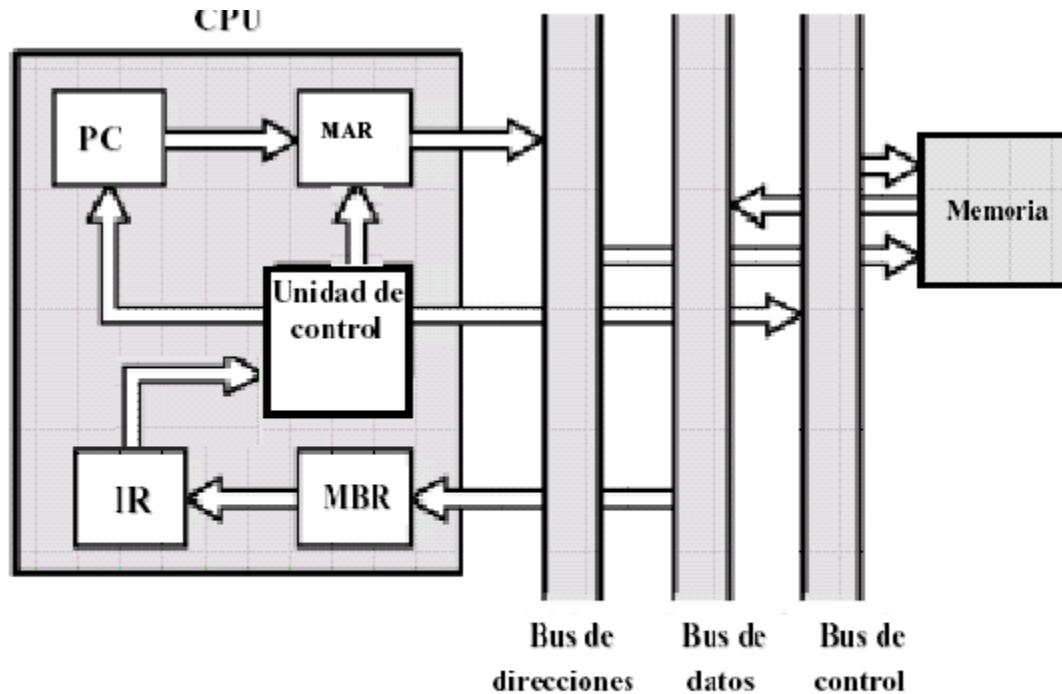


Figura 6.8 • Diseño del registro 32x1.

# Registros y Buses



**MBR** = Registro intermedio de memoria

**MAR** = Registro de dirección de memoria

**IR** = Registro de instrucción

**PC** = Contador de programa

■ **BUS** : Es un grupo de conductores que permite llevar la información entre los distintos bloques de la computadora digital.

## ■ INSTRUCCIÓN

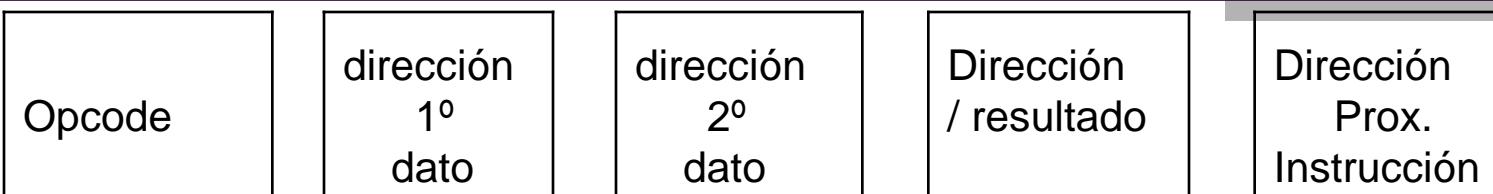
**Es una orden o comando para que la ALU realice alguna operación.**

## ■ PROGRAMA

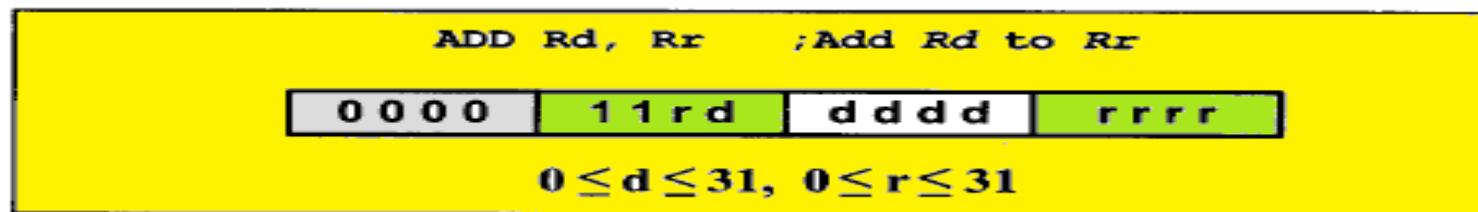
**Secuencia de instrucciones que permiten a la ALU procesar datos y obtener resultados, con algún fin específico.**

Aritmética	Bit & otros	Transferencia
ADD Rd, Rr	BSET s	MOV Rd, Rr
ADC Rd, Rr	BCLR s	MOVW Rd, 1:Rd, Rr + 1:Rr
ADIW Rd + 1:Rd, K6	OSE IO5, b CBI IO5, b	EN Rd, IO6
SUB Rd, Rr	BST Rd, b	CABO IO6, Rr
SUBI Rd, K8	BLD Rd, b	
SBC Rd, Rr		EMPUJE Rr
SBCI Rd, K8	NOP	POP Rr
SBIW Rd + 1:Rd, K6	DESCANSO DORMIR	LDI Rd, K8
INC Rd	WDR	LDS Rd, D16
DEC Rd		
Y Rd, Rr		LD Rd, X LD Rd, -X
ANDI Rd, K8		LD Rd, X +
O Rd, Rr		
ORI Rd, K8		LDD Rd, Y + δD6
EOR Rd, Rr		LD Rd, -Y LD Rd, Y +

# Instrucción de programa



(i)	<b>Syntax:</b> STS k,Rr	<b>Operands:</b> $0 \leq r \leq 31, 0 \leq k \leq 65535$								
	<b>32-bit Opcode:</b> <table border="1"><tr><td>1001</td><td>001d</td><td>dddd</td><td>0000</td></tr><tr><td>kkkk</td><td>kkkk</td><td>kkkk</td><td>kkkk</td></tr></table>	1001	001d	dddd	0000	kkkk	kkkk	kkkk	kkkk	<b>Program Counter:</b> $PC \leftarrow PC + 2$
1001	001d	dddd	0000							
kkkk	kkkk	kkkk	kkkk							



↔ 16 bits

# Ciclo de *máquina*

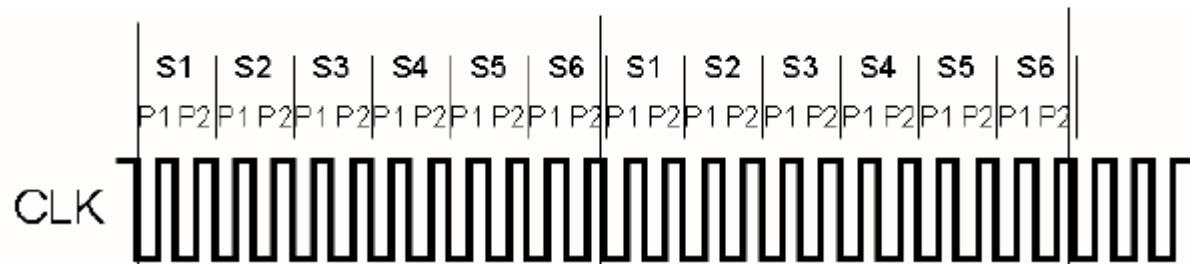
- Las operaciones del microprocesador son sincronizadas por un osc.
- Un ciclo de instrucción consiste de uno o más ciclos de Máq.
- Un ciclo de Máq. Puede variar de 1(avr) a 12 pulsos de clock (8051)



# Ciclo de máquina en el 8051

Ciclo máquina: 12 ciclos del oscilador ( $12\text{MHz} \rightarrow 1\mu\text{s}$ )

Las instrucciones se ejecutan generalmente en 1 o 2 ciclos



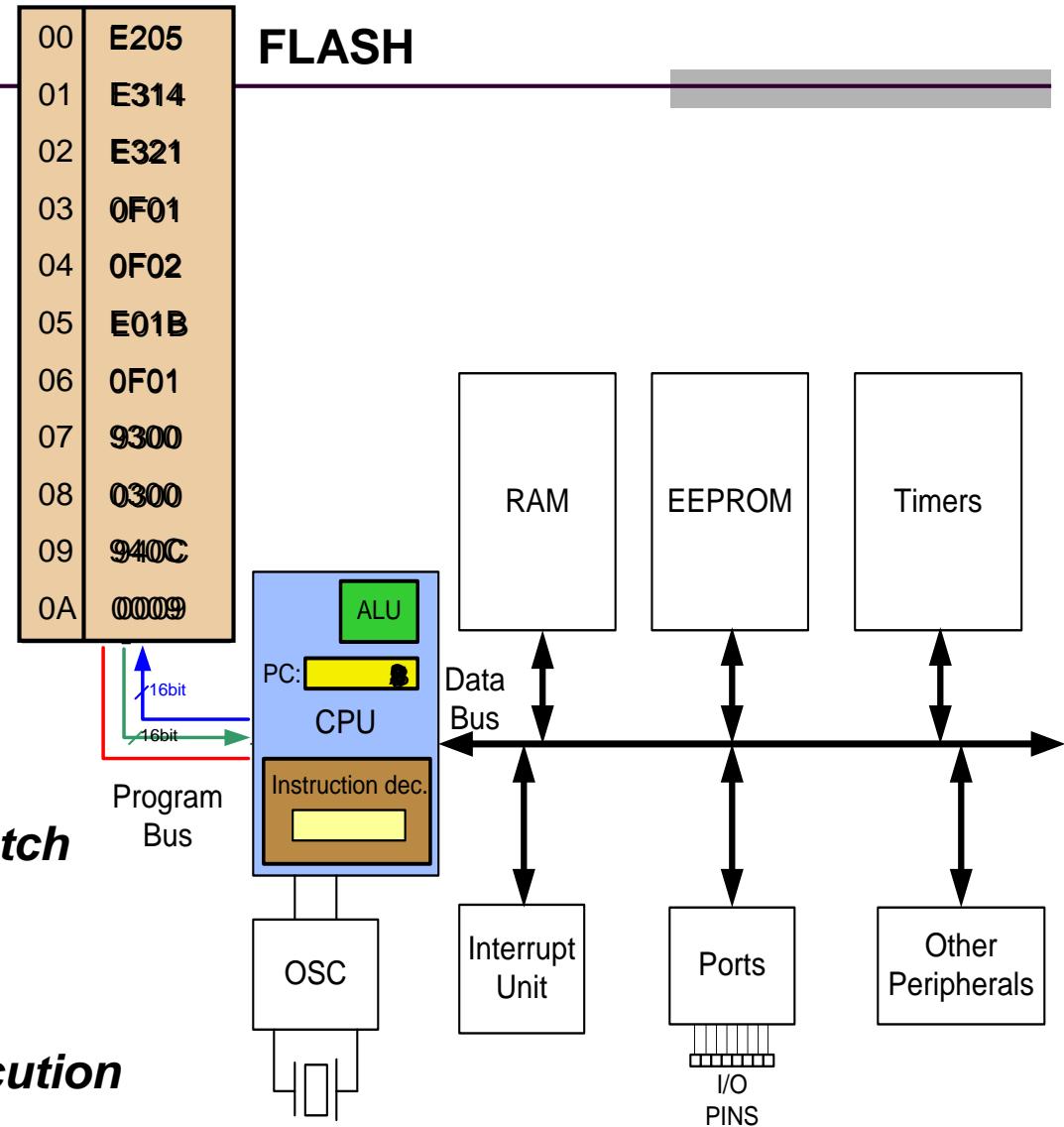
# . Ciclo de una instrucción (FETCH+EJEC.)

---

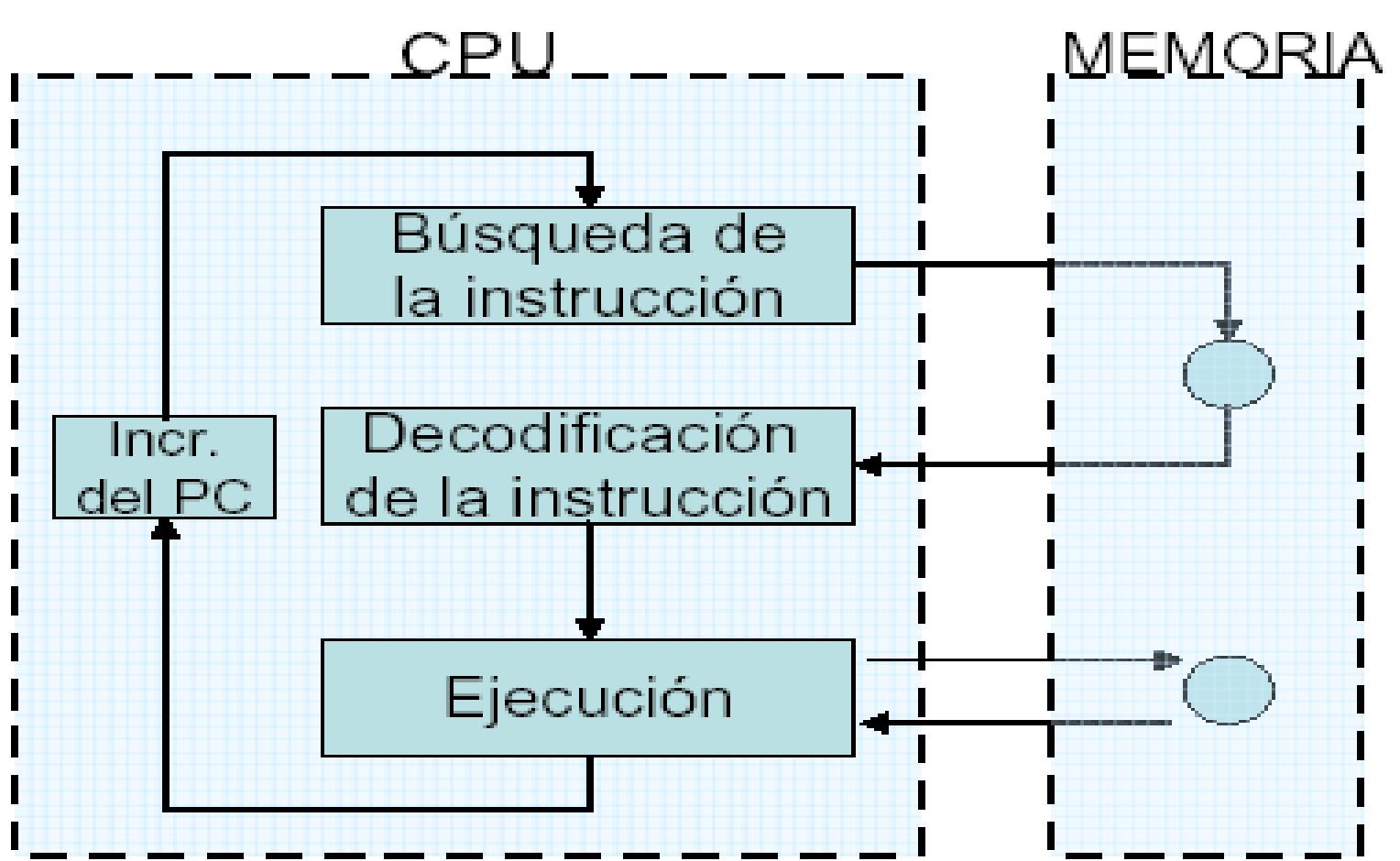
- 1. *Subciclo de búsqueda (fetch subcycle)*
  
- 2. *Subciclo de ejecución (execution subcycle).*

# Ciclo de una instrucción (FETCH+EJEC.)

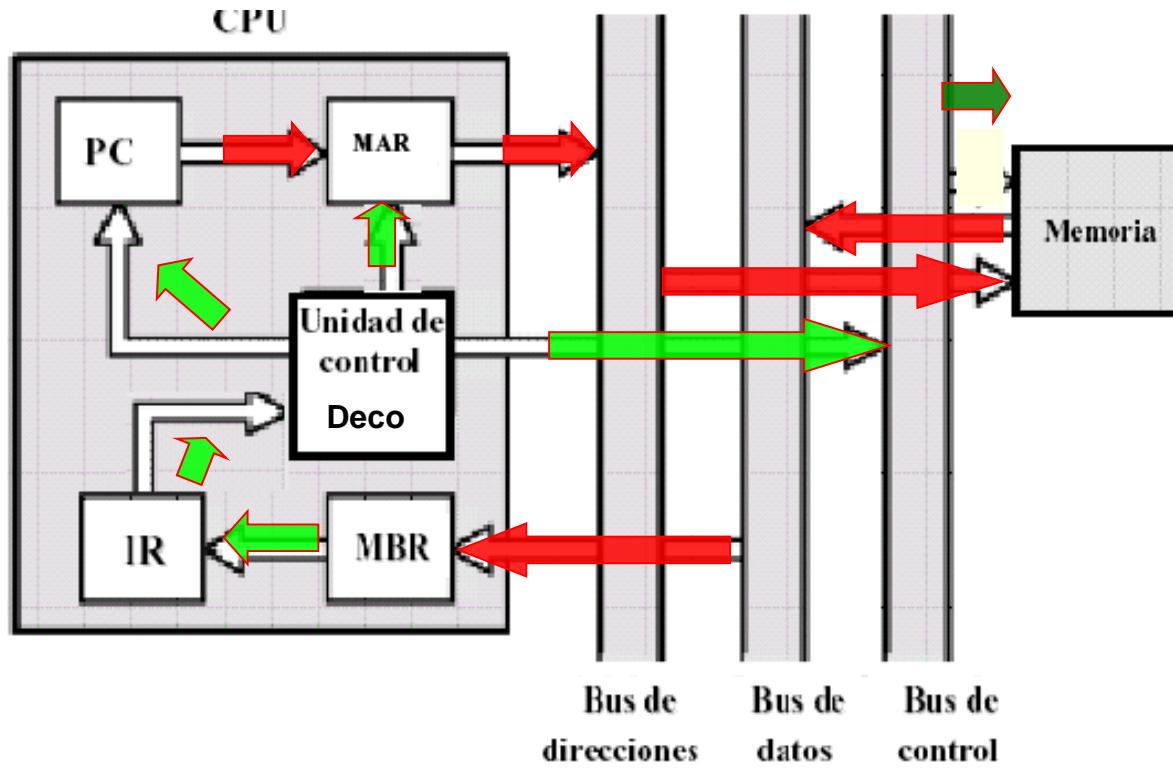
LDI R16, 0x25  
LDI R17, \$34  
LDI R18, 0x31  
ADD R16, R17  
ADD R16, R18  
LDI R17, 11  
ADD R16, R17  
STS SUM, R16  
**HERE : JMP HERE**



# Ciclo de una instrucción (FETCH+EJEC.)



# FETCH + EJEC

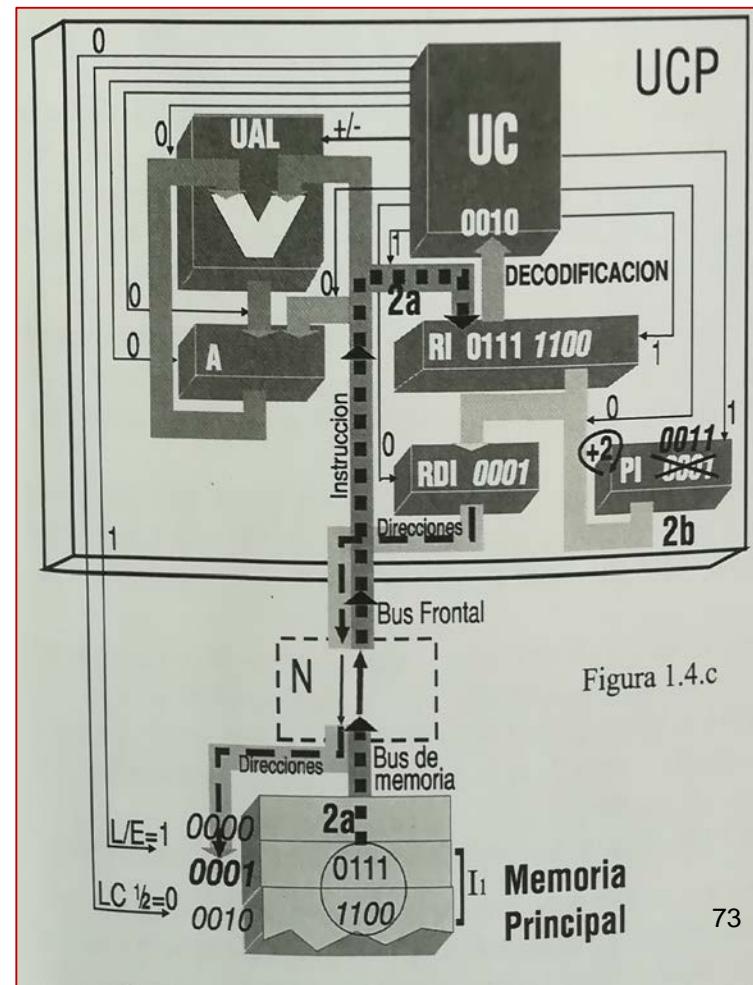
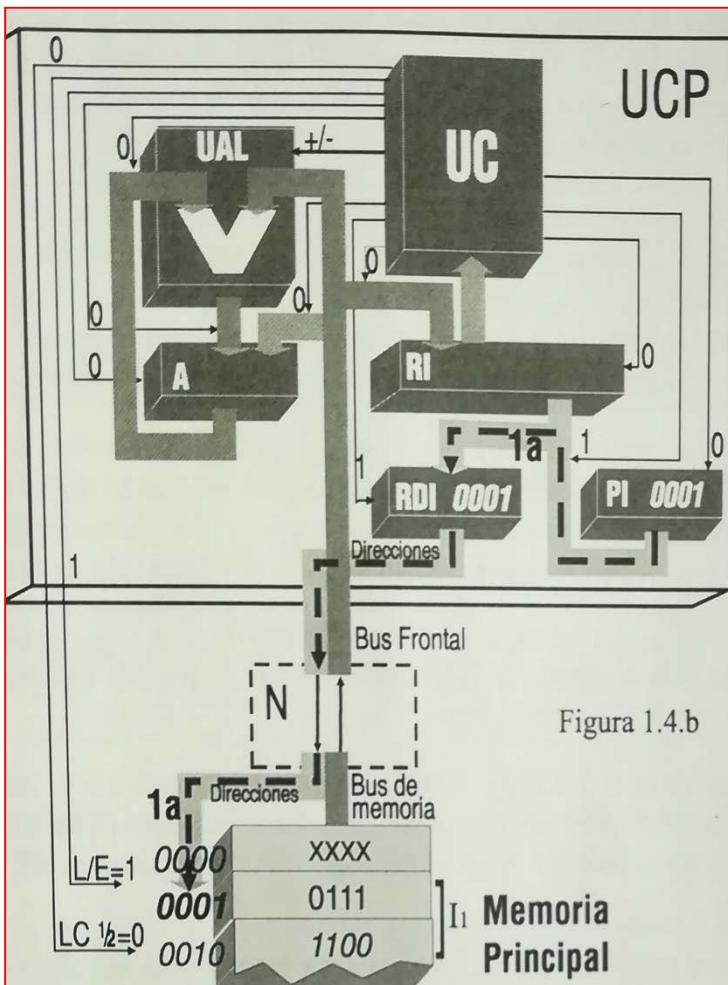
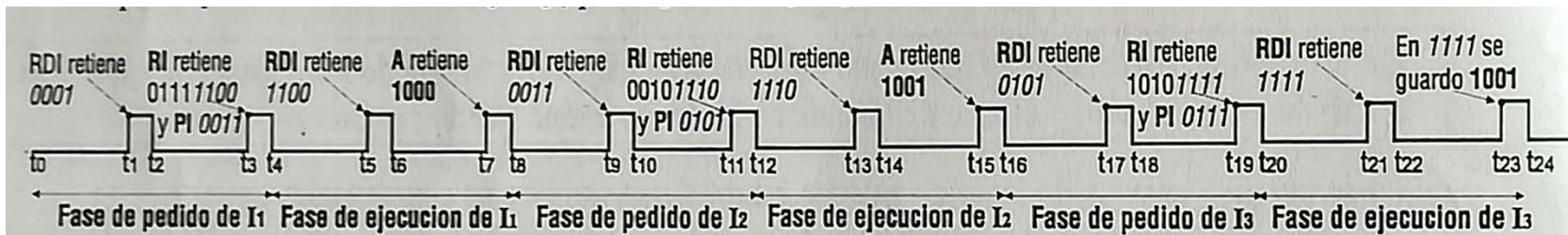


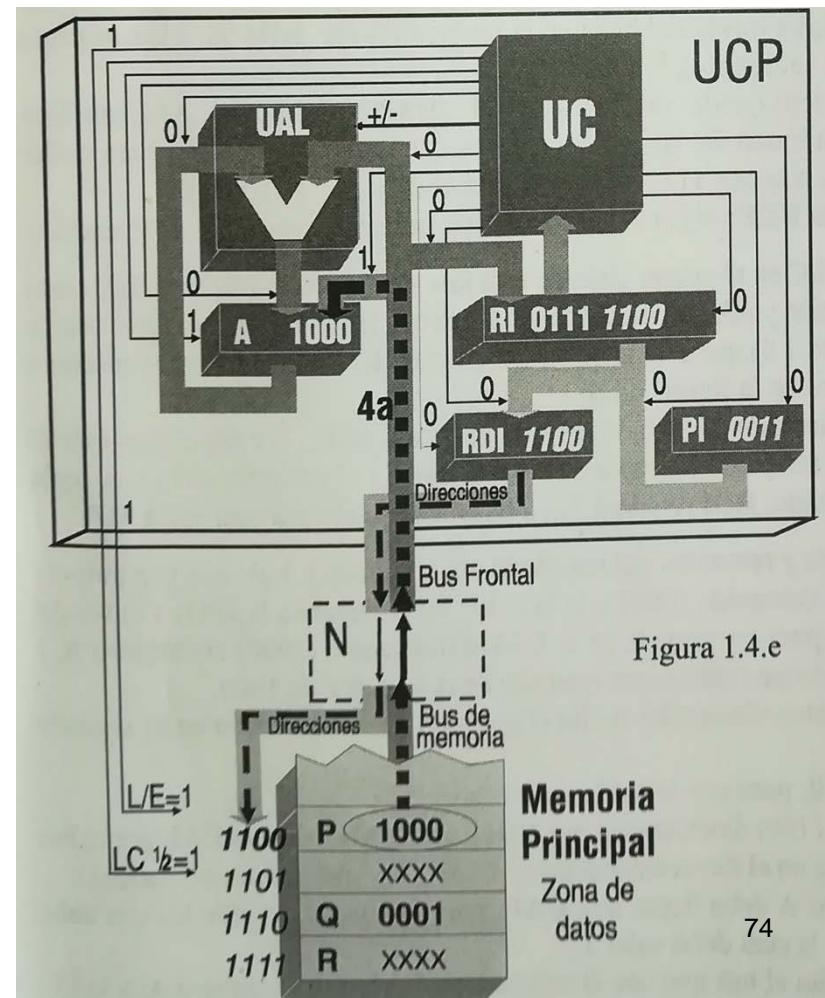
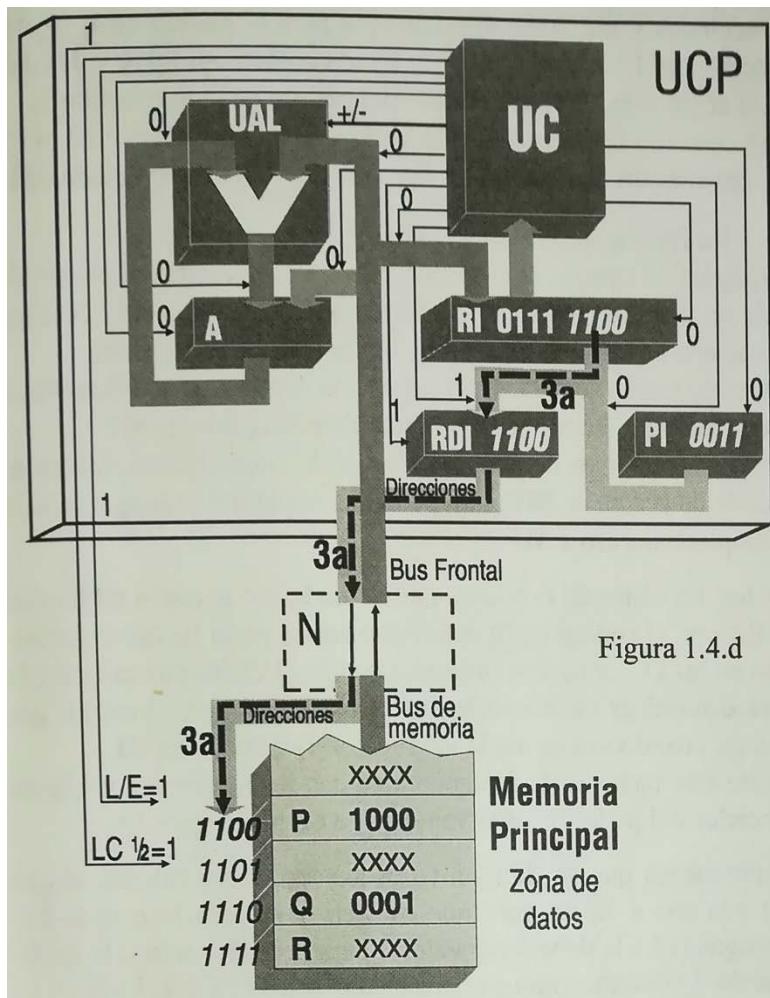
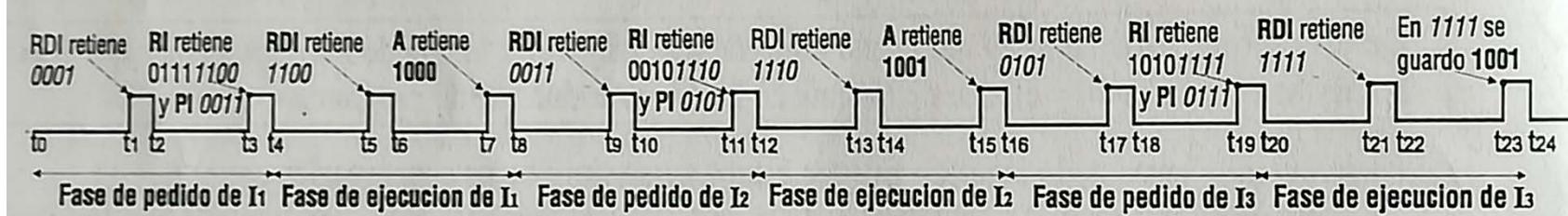
**MBR = Registro intermedio de memoria**

**MAR = Registro de dirección de memoria**

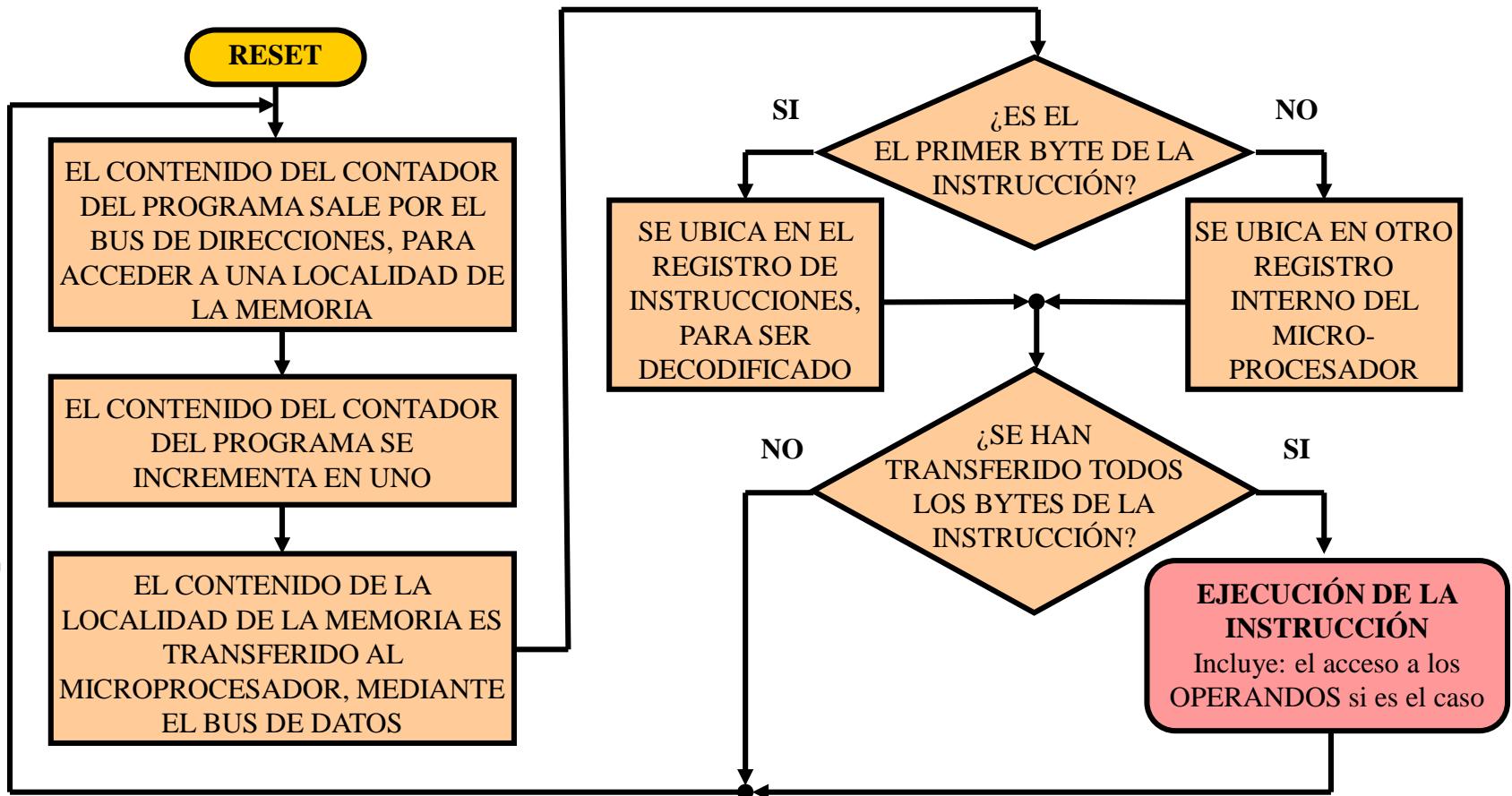
**IR = Registro de instrucción**

**PC = Contador de programa**



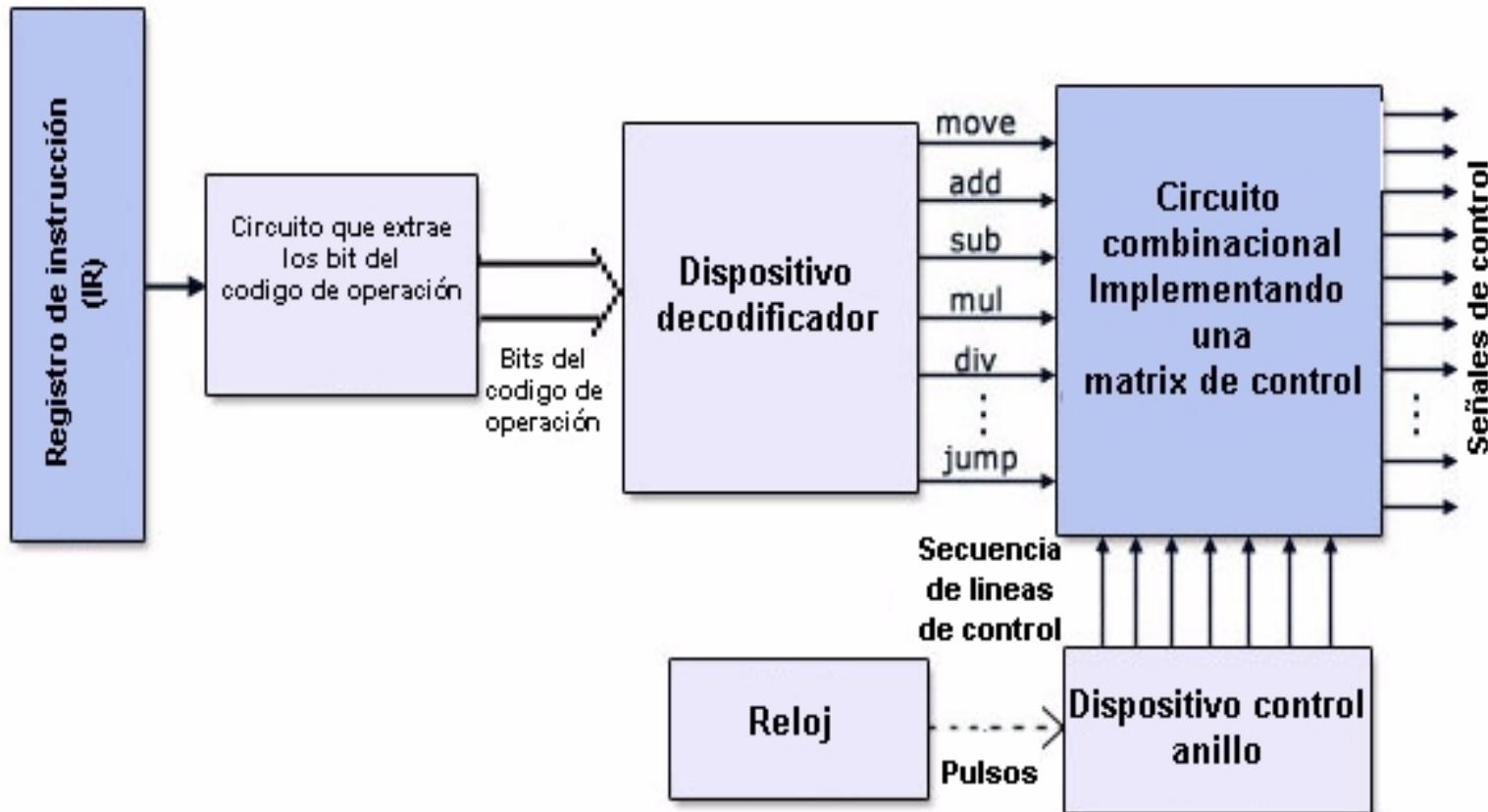


# DIAGRAMA DE FLUJO DEL CICLO DE LAS INSTRUCCIONES



# Unidad de Control Cableado

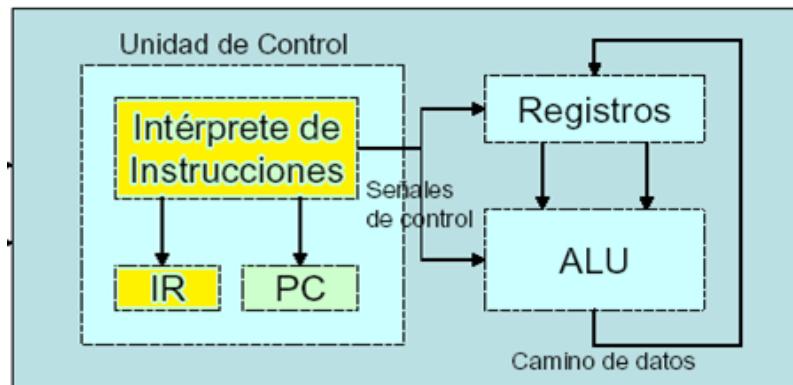
- El control cableado consiste de una máquina de estados y un circuito combinacional
- Es un autómata que se mueve de un estado a otro dependiendo de la entrada recibida.



# Unidad de control

## 1. LÓGICA CABLEADA (PUERTAS LÓGICAS).

- Diseño muy laborioso y puesta a punto muy complicada.
- Una vez construida, es muy difícil modificarla.  
Mas rápida.
- Lo usan los computadores que deben ser extremadamente potentes.



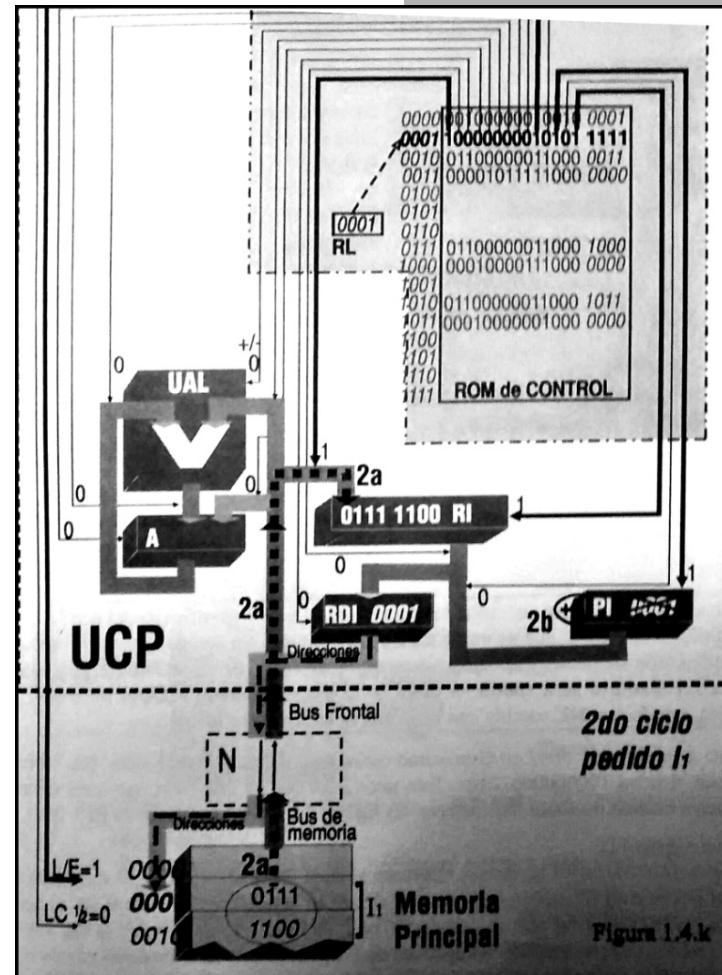
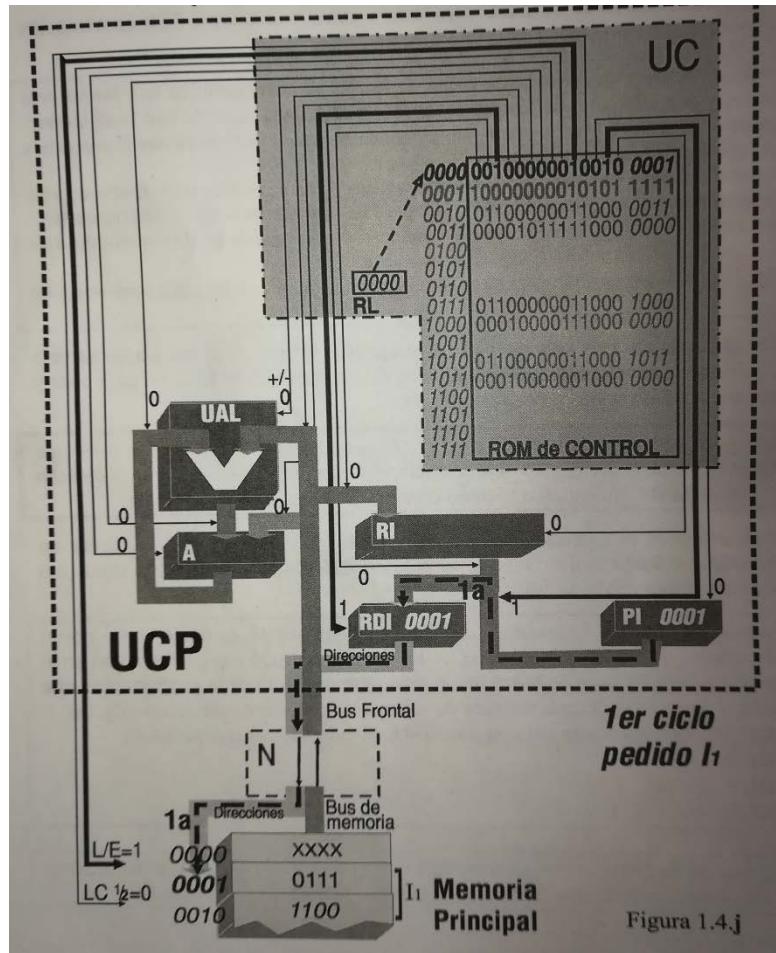
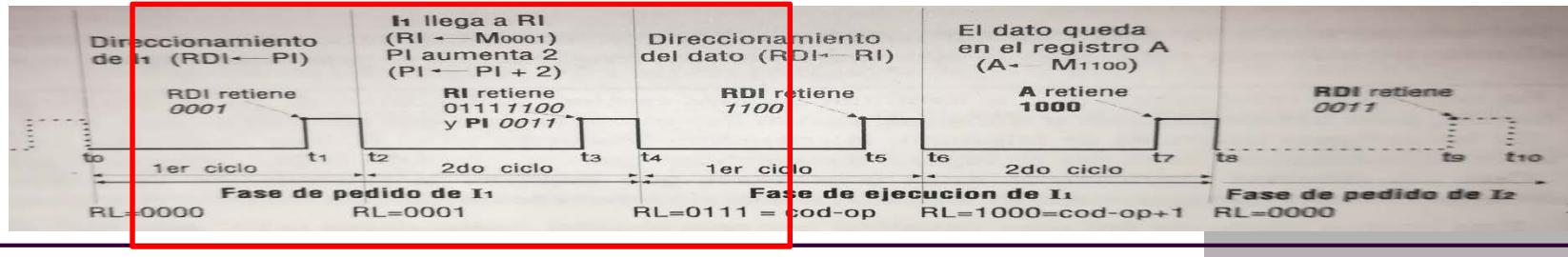
## Unidad de Control Microprogramado

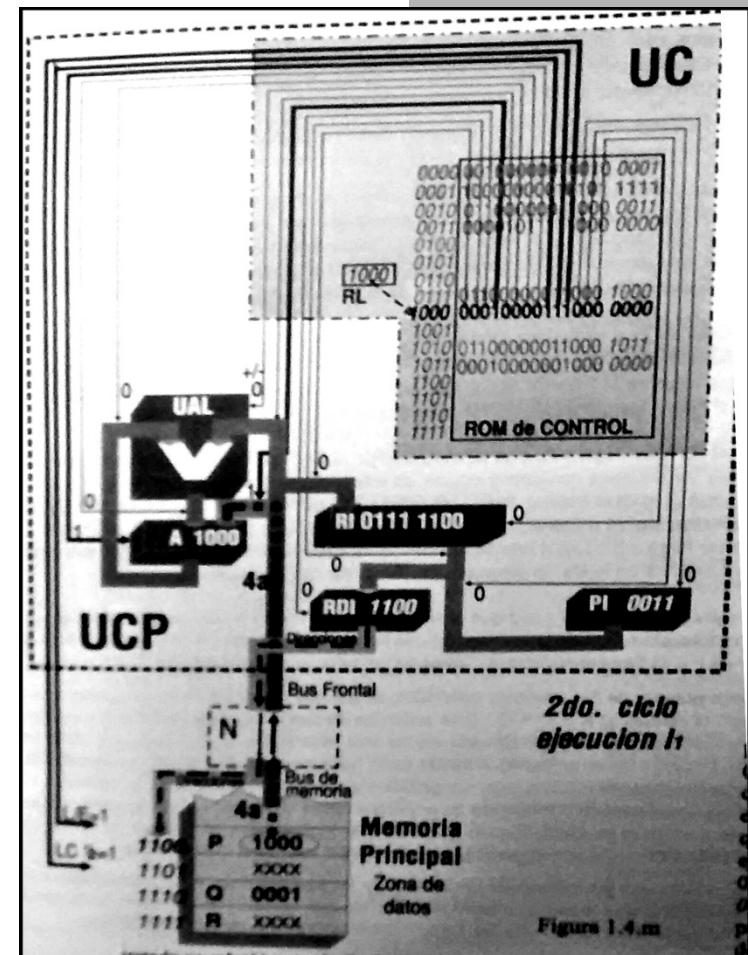
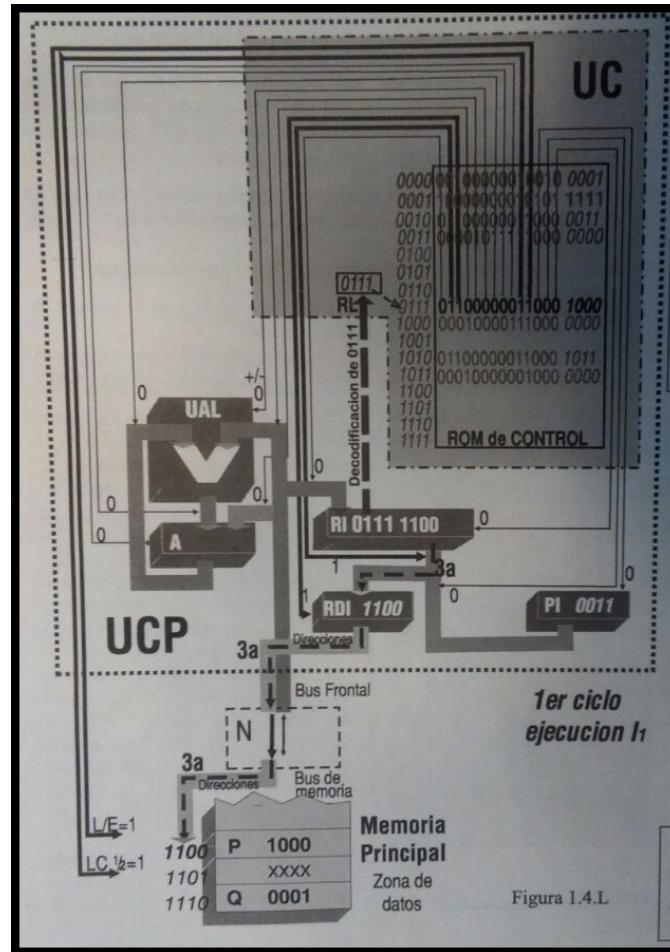
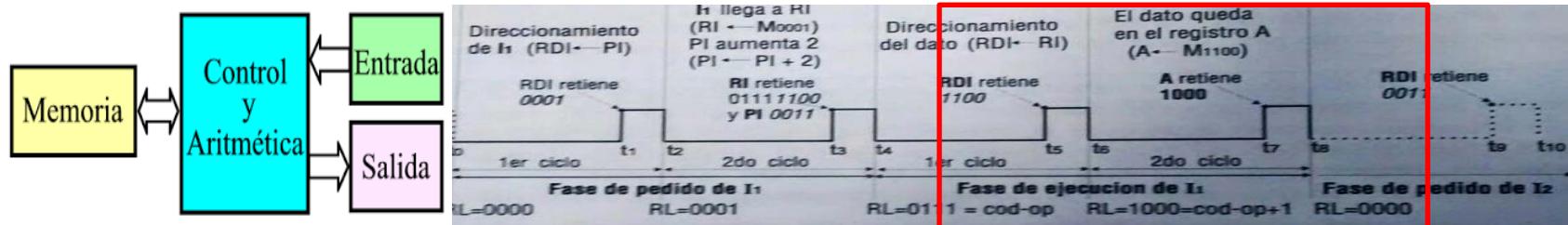
---

- Cuando la CPU ejecuta una instrucción, realmente ejecuta un conjunto de microinstrucciones llamadas micro\_rutinas
- Cada microinstrucción requiere de un ciclo de CPU para ejecutarse
- Cada Micro-rutina está almacenada generalmente en una memoria de sólo lectura también llamada ROM de control

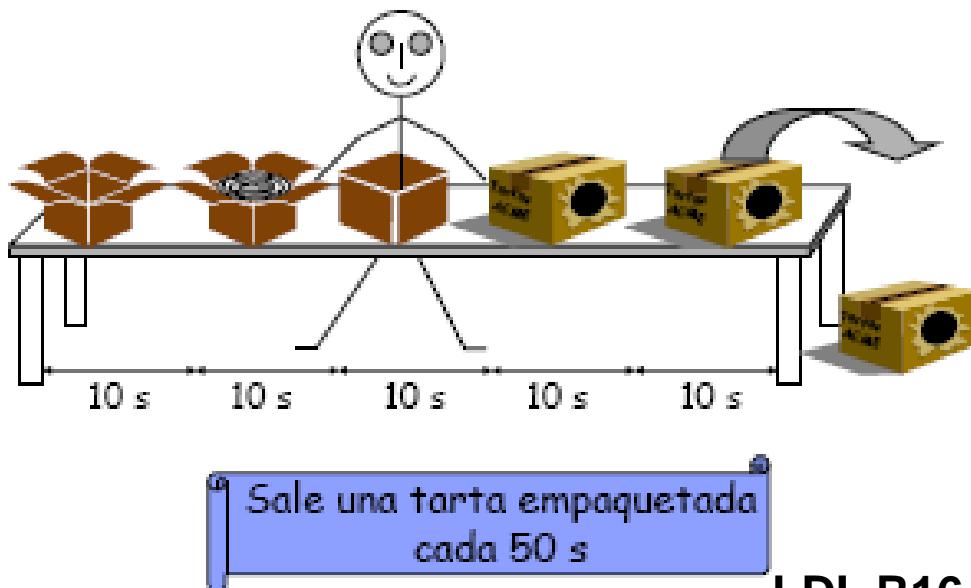
- Las características del **Control Microprogramado** son:
- Permite crear infinitas instrucciones complejas que puedan ser descompuestas en las microinstrucciones básicas
- Cuando se necesitan nuevas instrucciones no hay que modificar la parte física de un cpu sólo hay que programar la instrucción con sus equivalentes microinstrucciones.
- En el control cableado cada nueva instrucción requiere ser agregada físicamente en el hardware

# LÓGICA MICROPGRAMADA

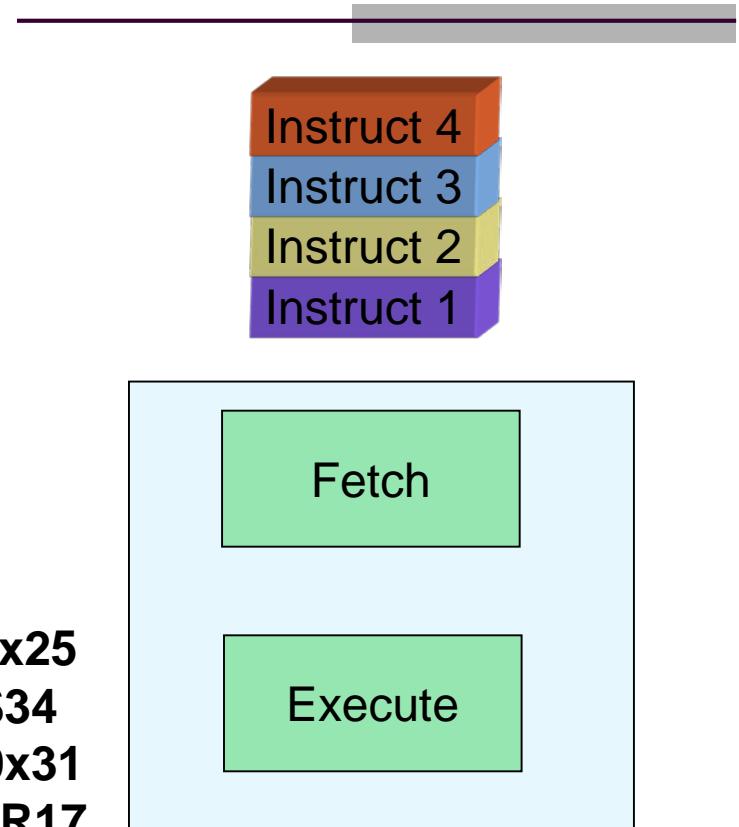




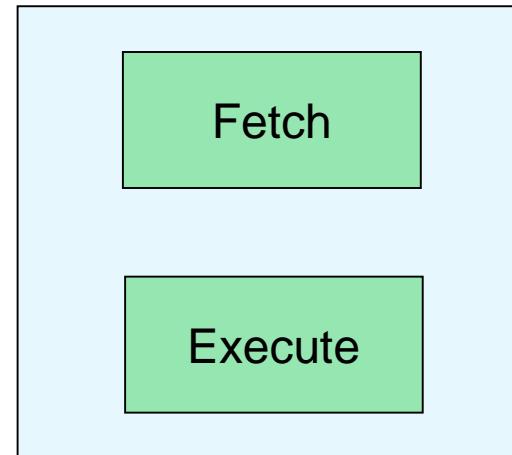
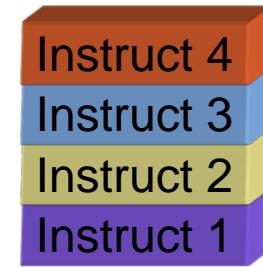
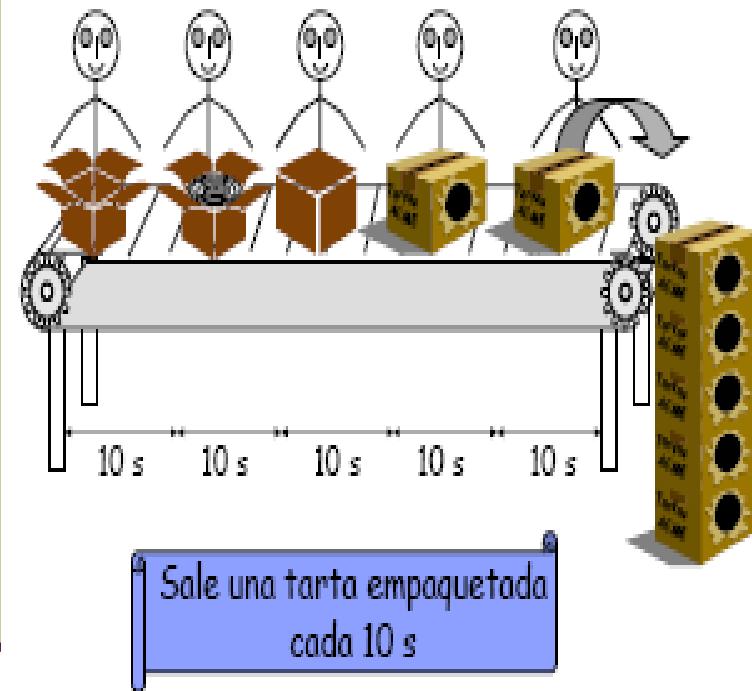
# Pipeline Segmentación



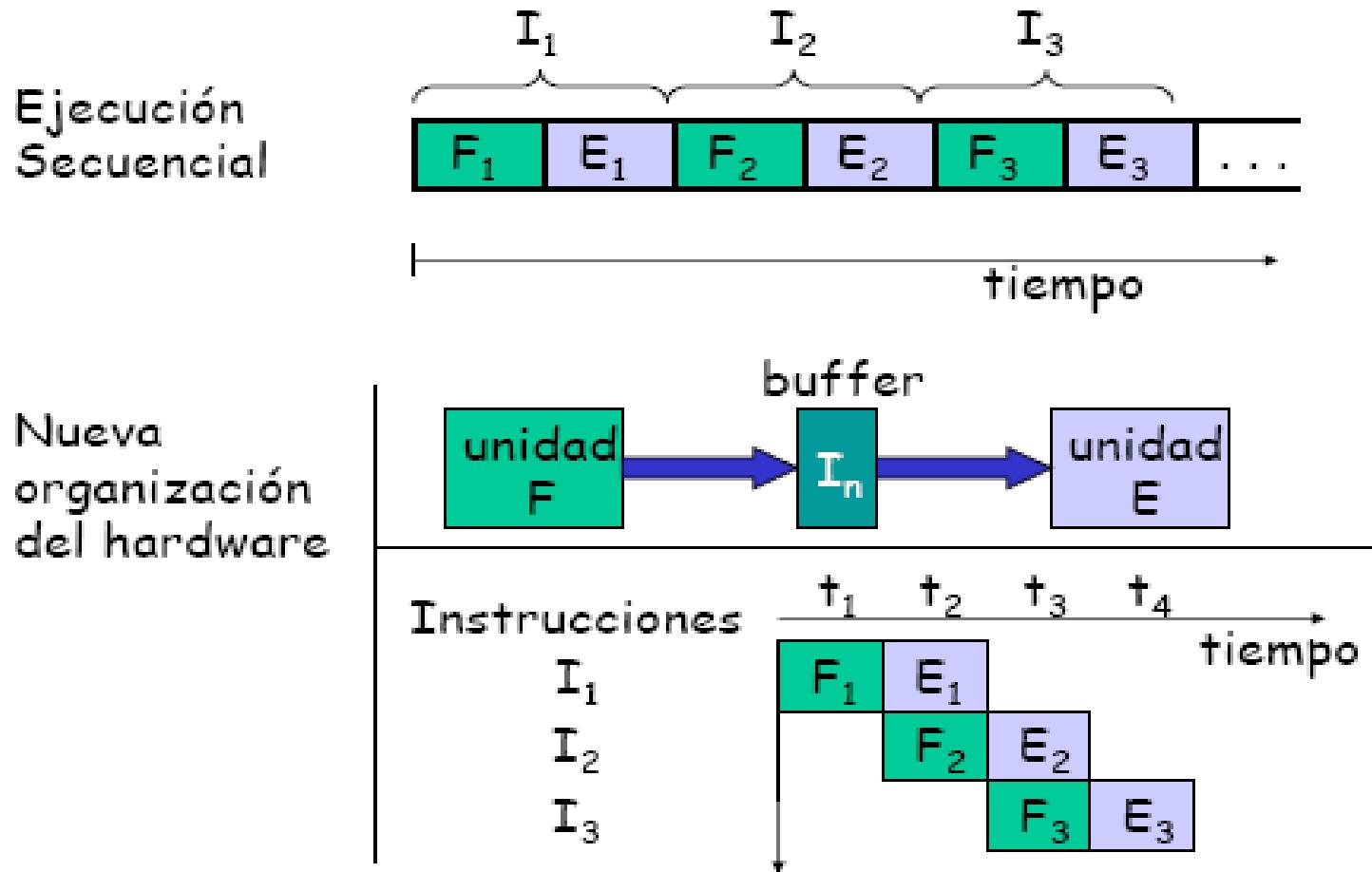
```
LDI R16, 0x25  
LDI R17, $34  
LDI R18, 0x31  
ADD R16, R17  
ADD R16, R18  
LDI R17, 11  
ADD R16, R17  
STS SUM, R16  
HERE : JMP HERE
```



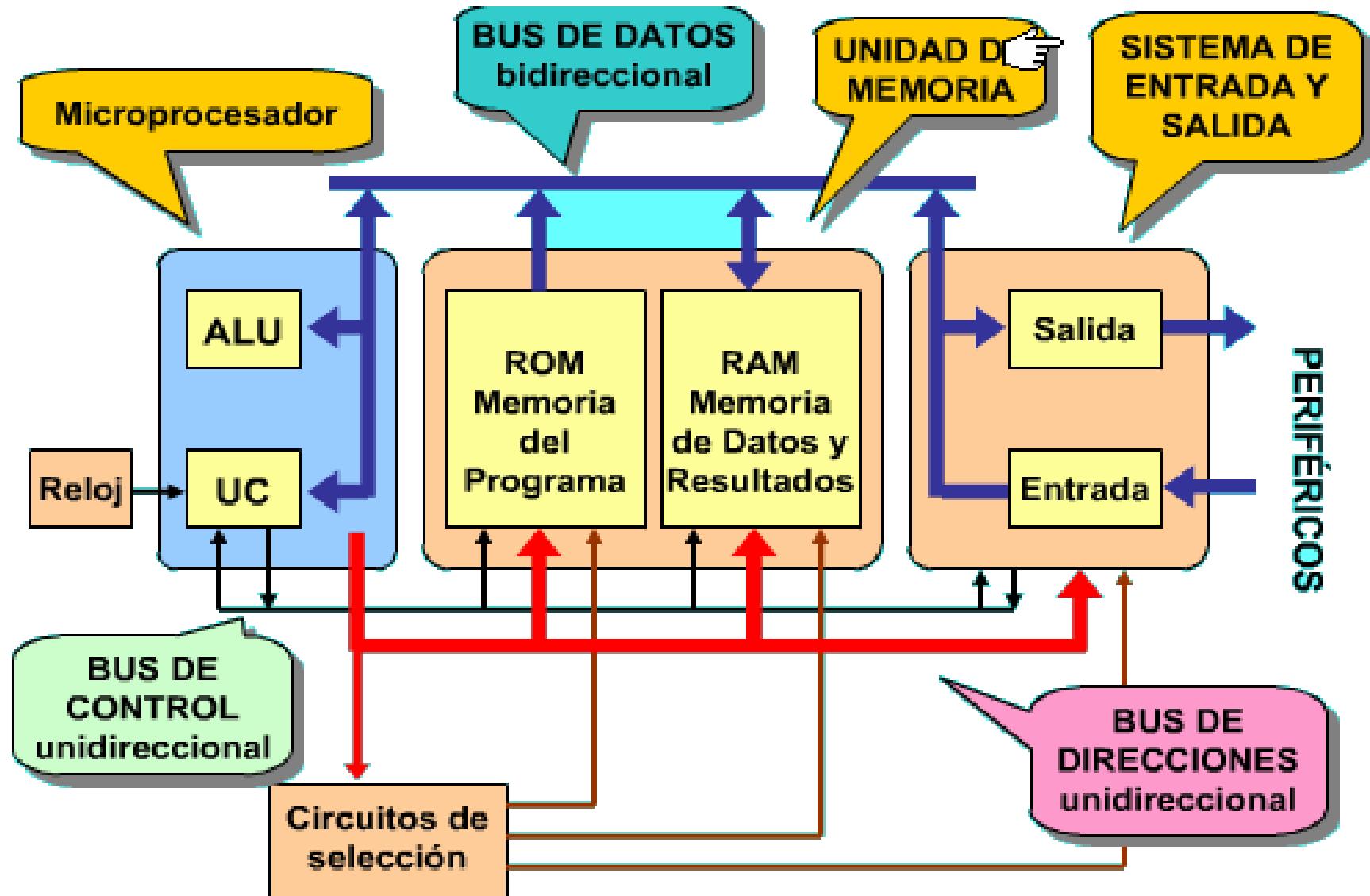
# Pipeline Segmentación



# Pipelined instruction architecture



# Resumen



# Resumen

C  
P  
U  
I

Instrucciones

Registros

PC  
MAR  
MDR  
IR  
PSW

Instrucciones  
Aritmética y Lógica

OR  
NOR  
AND  
NAND  
Desplazamiento  
Rotación

Movimientos  
de datos

Reg. a Reg.  
Reg. a Mem.  
Mem. A reg.  
Mem a mem.

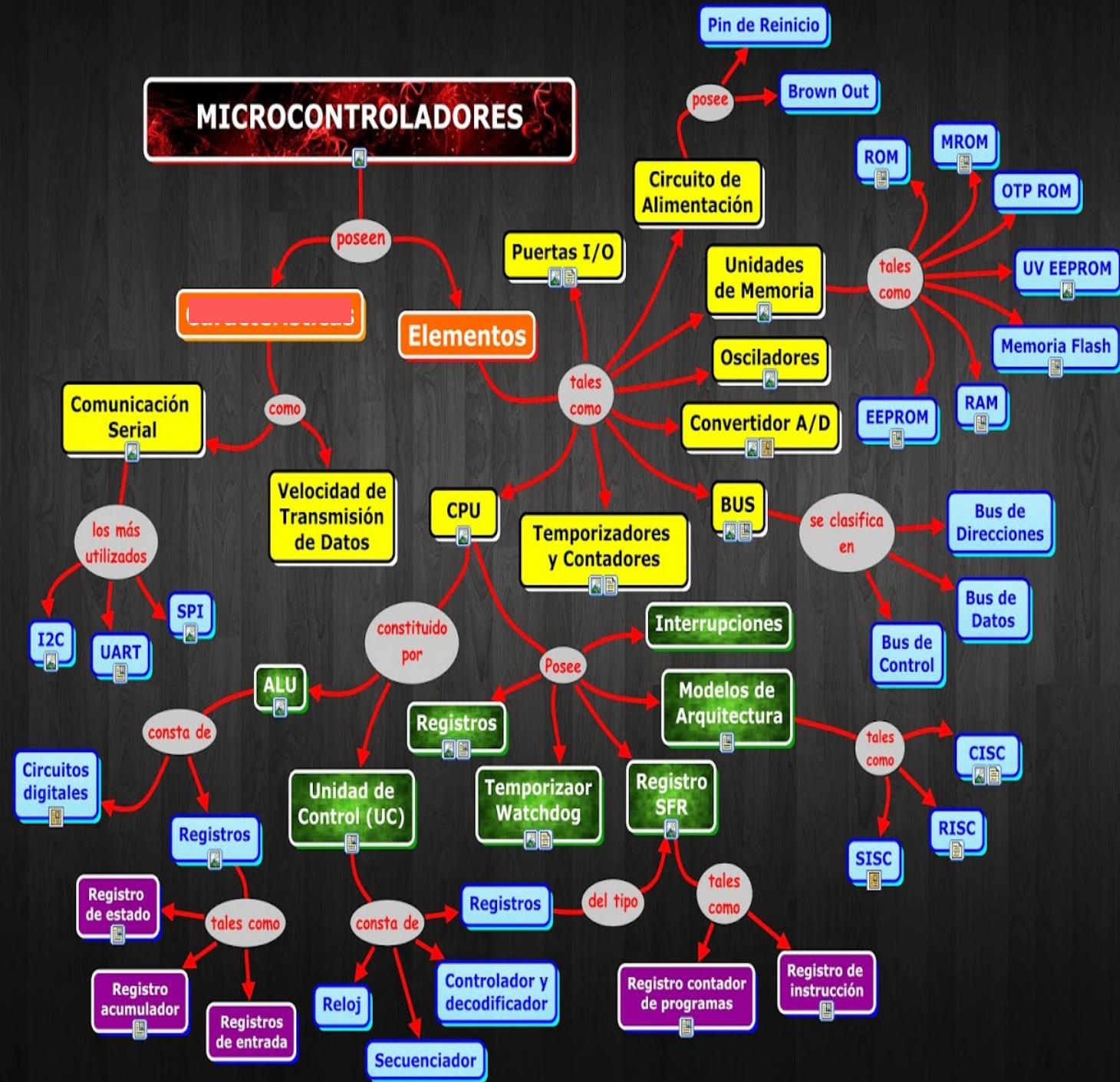
Operaciones de datos en bloque

Instrucciones de control de programa

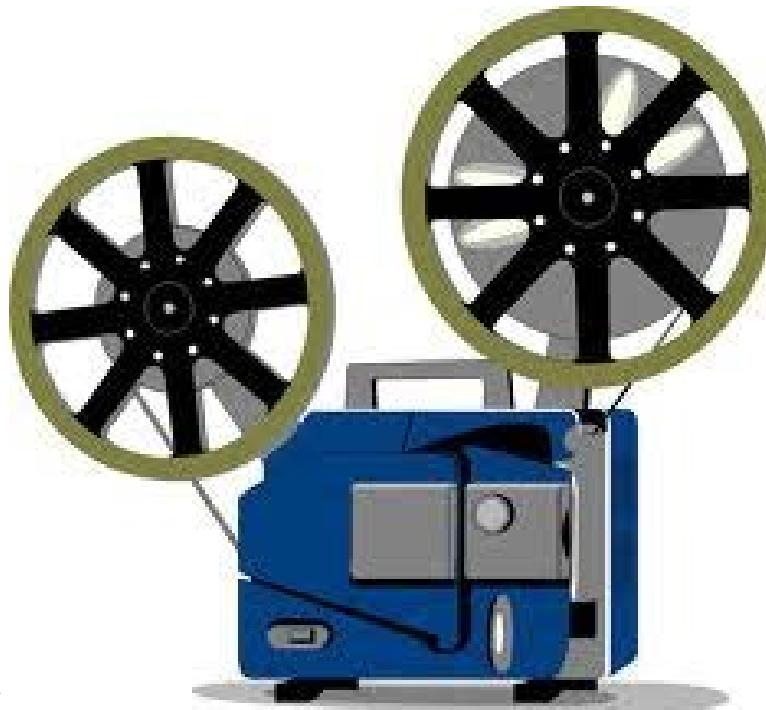
Instrucciones de Entrada-Salida

Reg. estados y control  
Reg. De información

# **MICROCONTROLADORES**

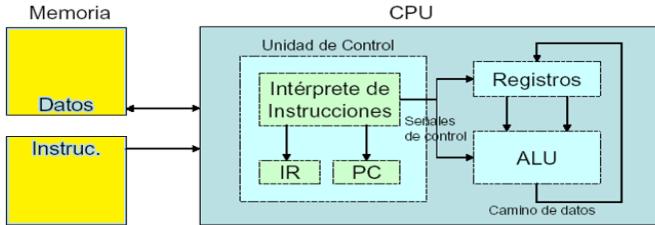


# ¿Qué podemos hacer?



[8051](#)  
[Microcontroller](#)  
[Buggy Robot.flv](#)

[Wall follower](#)  
[using iBOT.flv](#)



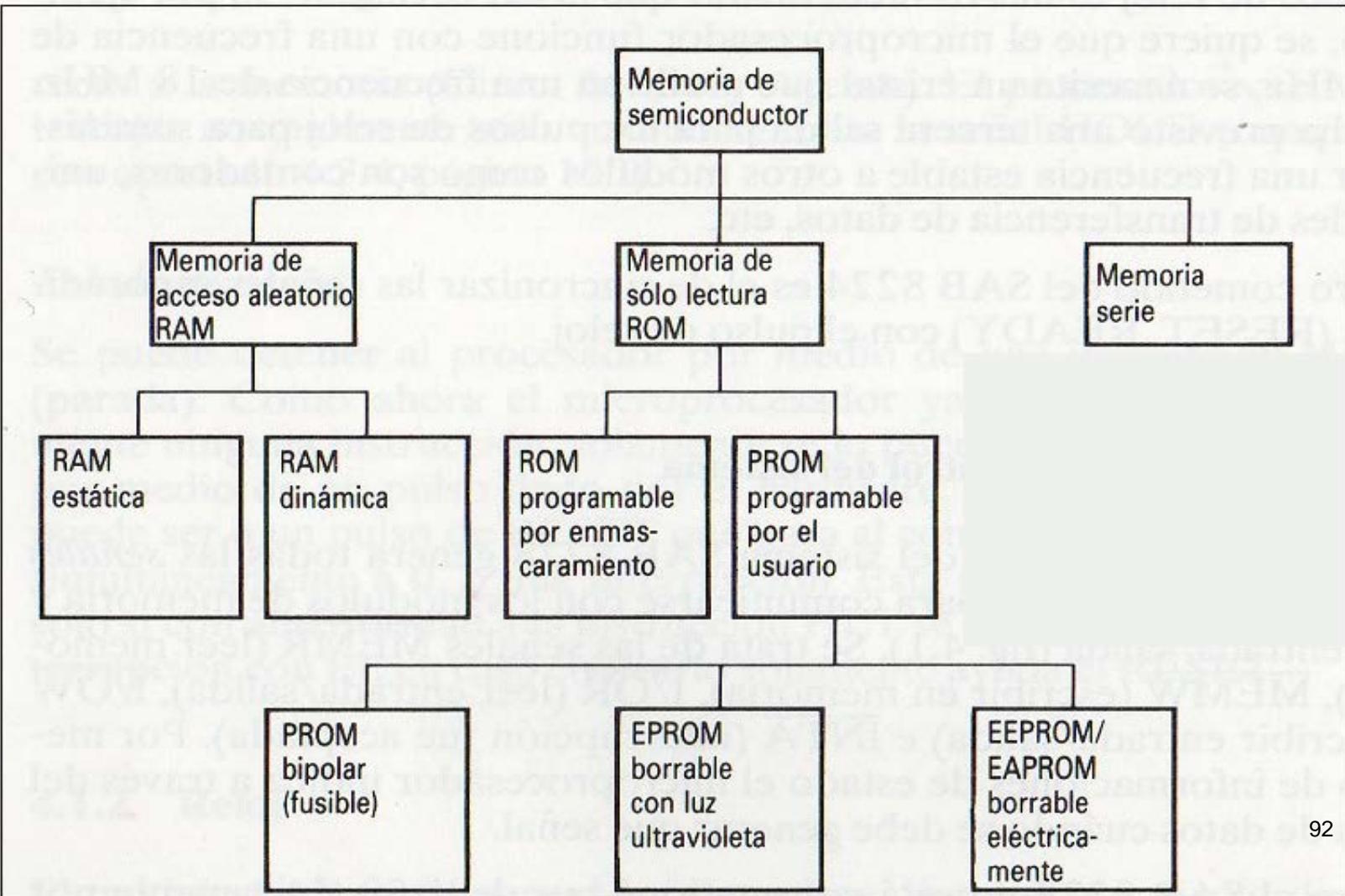
■ **UNIDAD DE MEMORIA :** Es el recurso donde se almacenan las instrucciones de los programas (memoria solo de lectura) y los datos y resultados que se procesan (memoria de lectura y escritura).

# Memoria

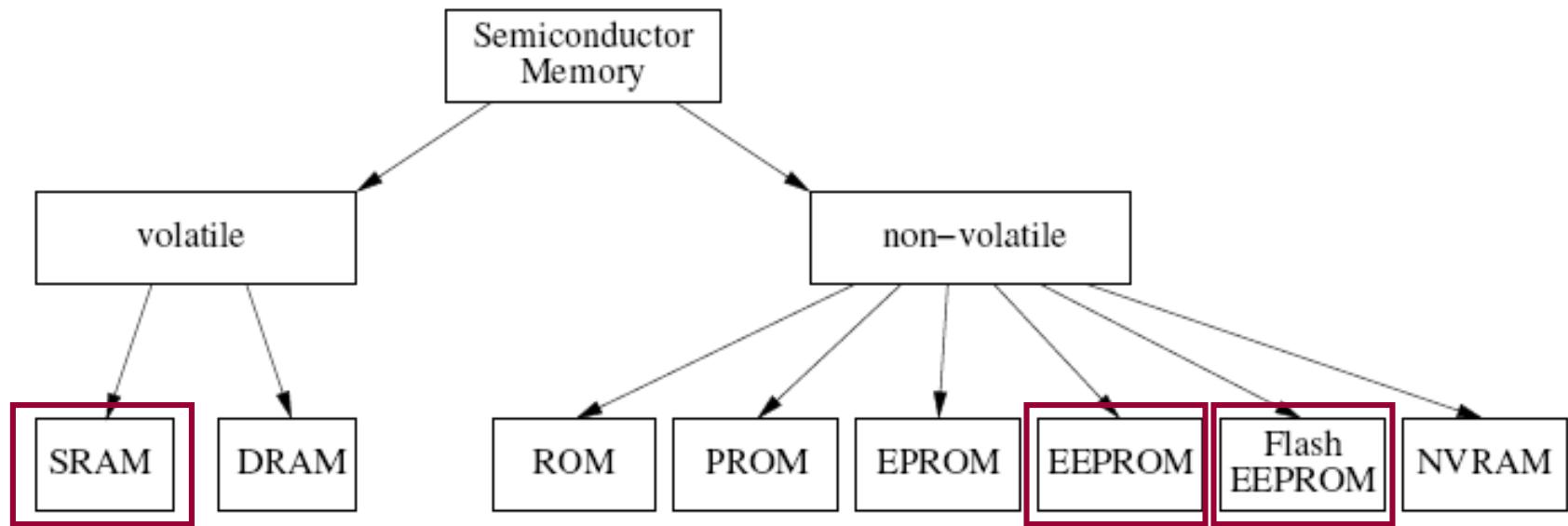
---

- **Registros** (memoria de corto plazo):
  - Pequeña (relativamente)
  - Almacenamiento temporario p/CPU
- **Memoria de datos**
  - Relativamente Grande
  - Almacena datos mientras el MCU funciona
- **Memoria de programa**
  - Relativamente Grande
  - De preferencia, mantiene el programa incluso con el MCU apagado.

# MEMORIAS



# Memoria: Tipo físicos



**S/DRAM: sin límite de escrituras**

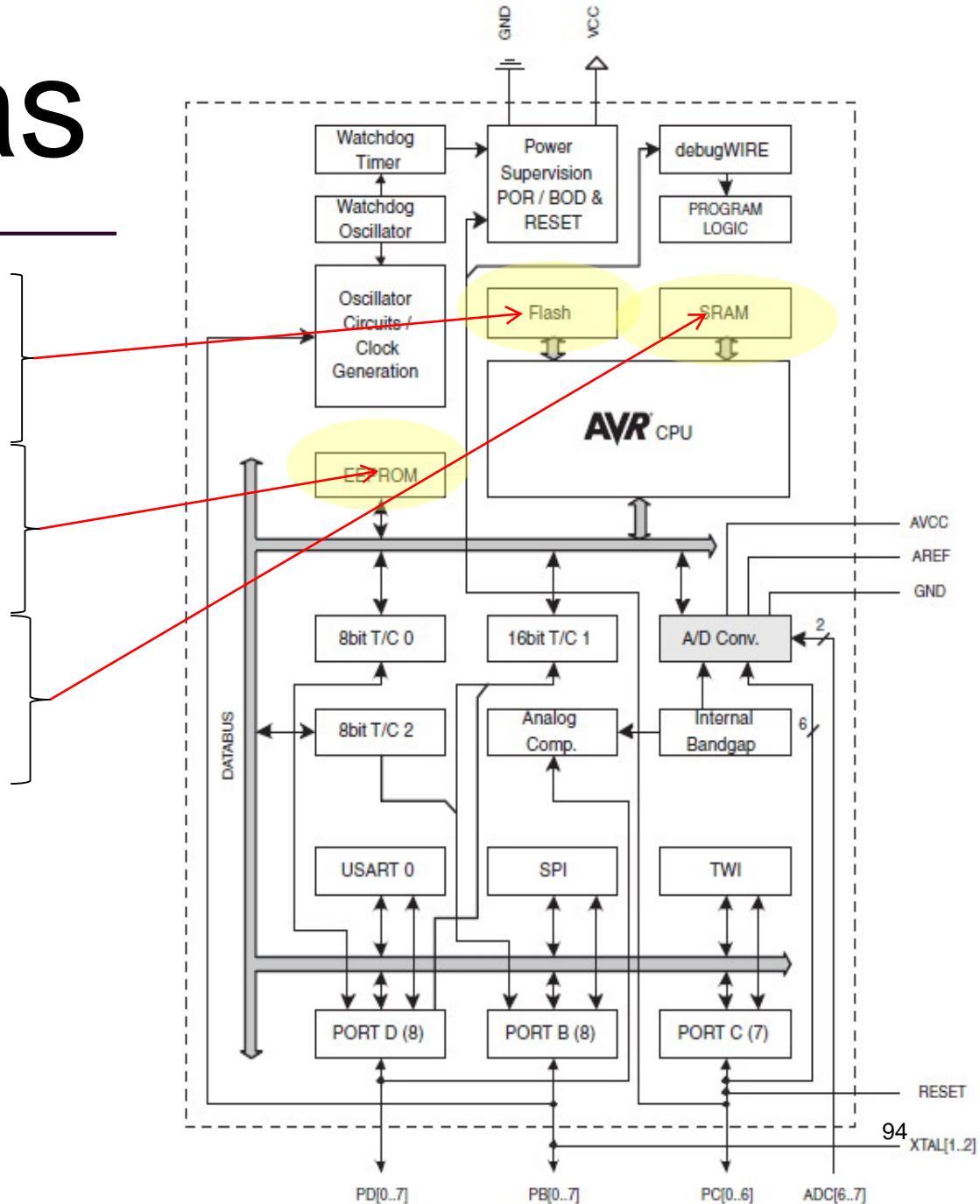
**EEPROM: 100.000 ciclos de borrado**

**Flash: 10.000 ciclos de borrado**

# Memorias

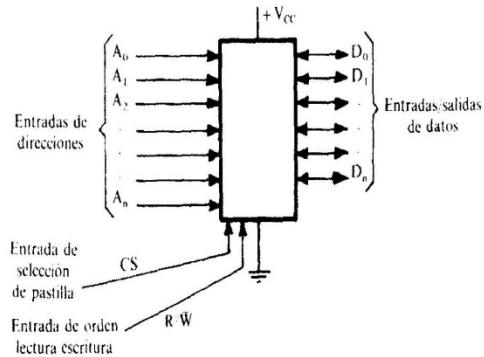
- 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
- 256/512/512/1KBytes EEPROM
  - 512/1K/1K/2KBytes Internal SRAM

Atmega328



## EL CHIP DE MEMORIA

- **Terminales**



- **Capacidad**

$$\text{Nº palabras (posiciones)} = 2^{\text{Nº terminales dirección}}$$

$$\text{Nº bits/palabra} = \text{Nº terminales datos}$$

$$\text{Nº bits (células)} = \text{Nº palabras} \times \text{Nº bits/palabra}$$

- Las pastillas suelen ser de 1, 4, 8 bits/palabra
- La capacidad se suele expresar en Kpalabras (2K = 2048 palabras)

## ESTRUCTURA INTERNA

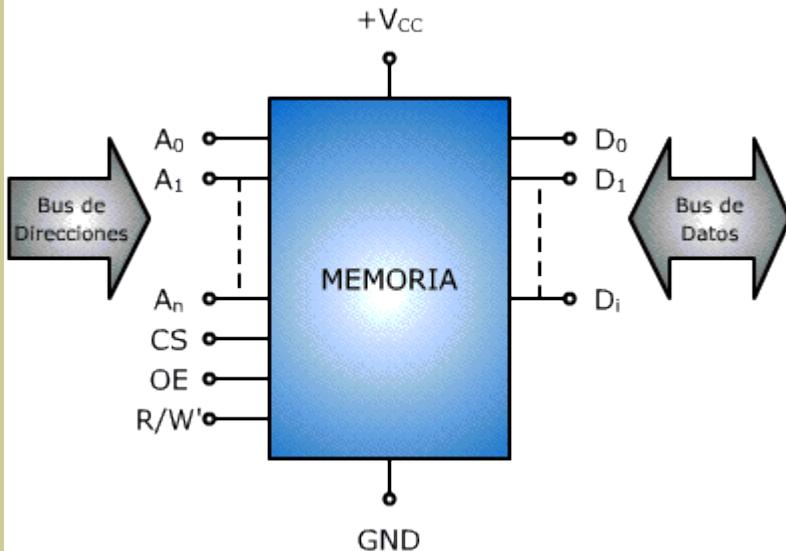
- **Elementos:**

- Matriz de células básicas
  - Organización por filas y columnas
  - Facilita el diseño con muchas células
- Decodificadores de filas y columnas
  - Permiten la selección de una posición
  - A veces se usan multiplexores
- Circuitos de lectura/escritura
  - Por donde salen/entran los datos
- Buffers de entrada y de salida

- **Opcionalmente:**

- Lógica de selección
  - Circuitos adicionales que conectados al bus de direcciones permiten seleccionar (activar) otros chips a través de CS ó CE
- Terminales de E/S
  - Usan buffers bidireccionales triestado
  - Permiten reducir el número de terminales
  - Estado de alta impedancia si el chip no está activado (CS)

# MEMORIAS



- Matriz de células básicas
  - Organización por filas y columnas
  - Facilita el diseño con muchas células
- Decodificadores de filas y columnas
  - Permiten la selección de una posición
  - A veces se usan multiplexores
- Circuitos de lectura/escritura
  - Por donde salen/entran los datos
- Buffers de entrada y de salida

- A<sub>0</sub>...A<sub>n</sub> (Bus de direcciones): Estos pines son las entradas para seleccionar la posición de memoria a escribir o leer y su cantidad define la capacidad de palabras que puede almacenar, dada por la expresión  $2^{**n}$ , donde n es el número de pines.
- D<sub>0</sub>...D<sub>i</sub> (Bus de Datos): Corresponde a los pines de entrada y salida de datos.
- CS (Chip Select): Este pin se utiliza para seleccionar el chip de memoria que se desea acceder. Esto en el caso del usar dos o más memorias similares.
- OE (Output Enable): Utilizado para habilitar la salida de datos. Cuando se encuentra en estado activo las salidas tiene alta impedancia o actúan como entradas.
- R/W' (Read/Write'): Entrada utilizada en las memorias RAM para seleccionar la operación de lectura o escritura
- VCC y GND (Alimentación): Corresponden a los pines de alimentación del circuito integrado.

# Memorias

---

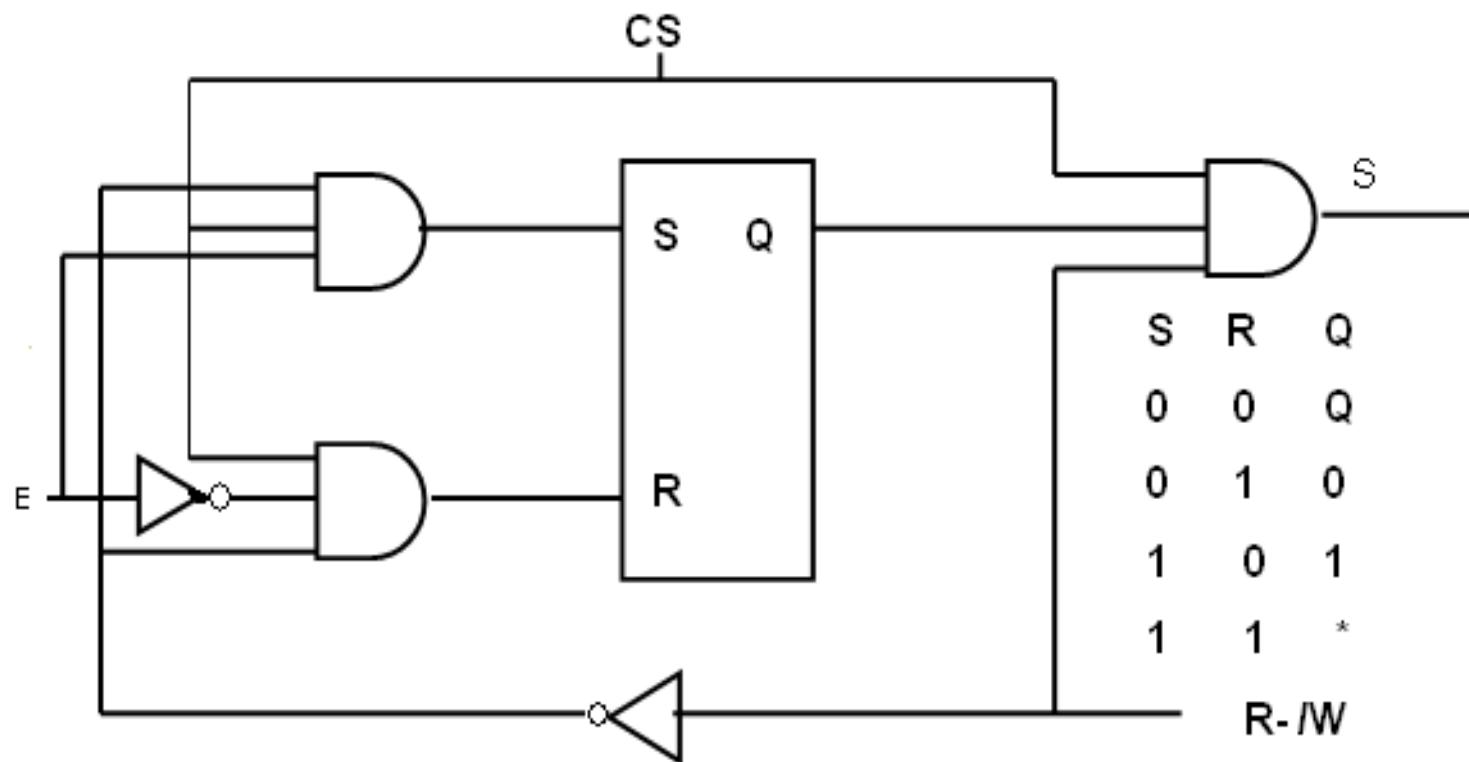
Las *RAM* estáticas tienen los siguientes inconvenientes:

- Densidad limitada
- Alto consumo de energía
- Conforme se incrementa la densidad de la memoria, los circuitos sensores se vuelven más complejos y requieren mayor energía.

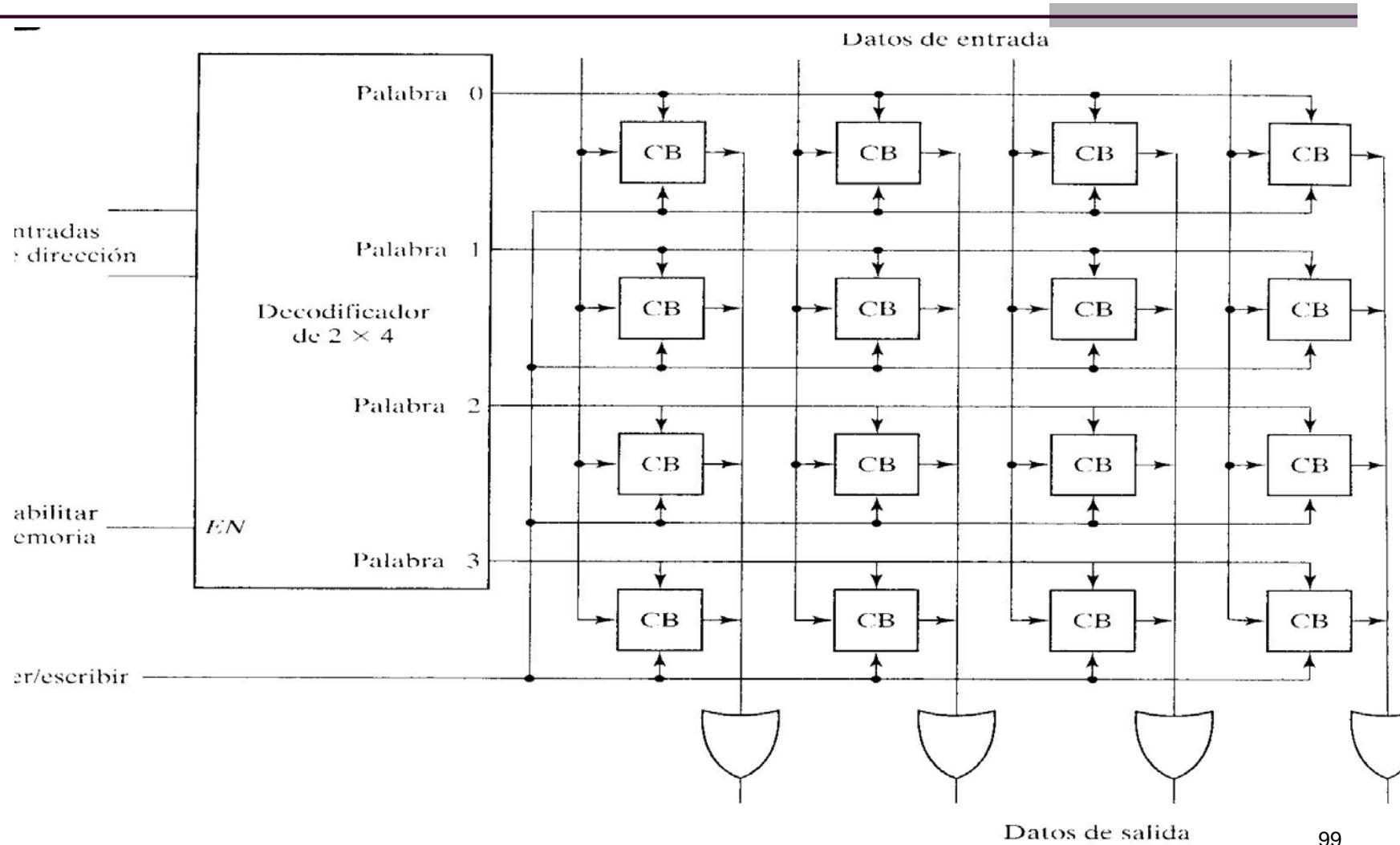
Las *RAM* dinámicas son de tecnología MOS y tienen la ventaja de que las celdas son de dimensiones menores a las bipolares y consumen poca energía, lo cual hace posible el que se puedan empaquetar muchas celdas en un circuito integrado.

La desventaja de las *RAM* dinámicas con respecto a las *estáticas* es la velocidad de acceso.

# Célula básica de memoria



# Diagrama lineal de RAM



# Arreglo Matricial

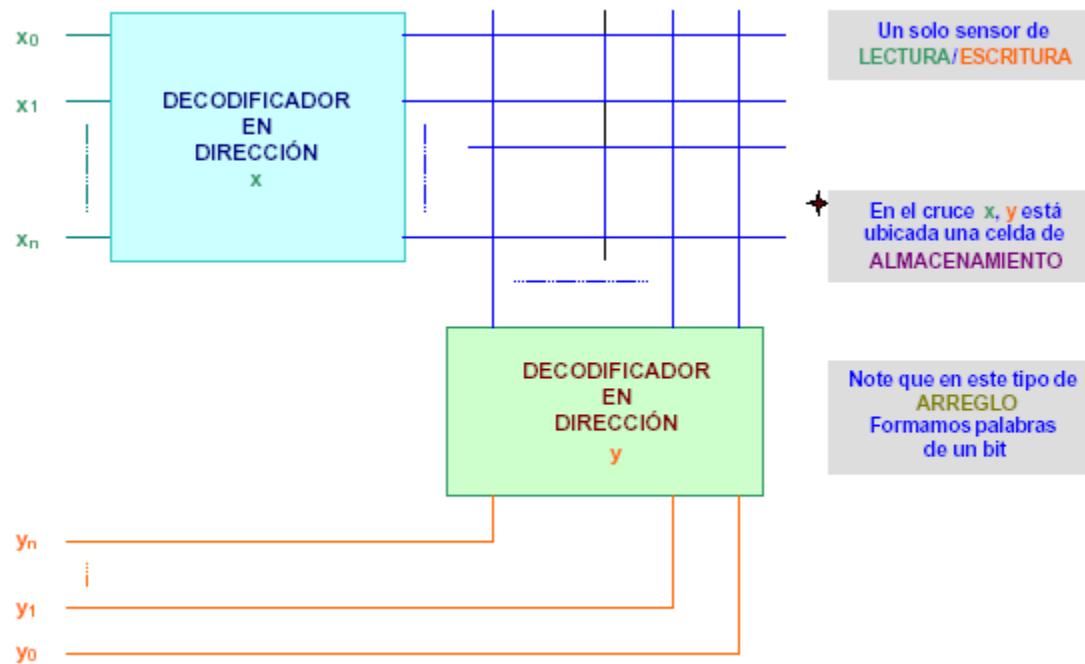


FIGURA 1.13. Diagrama a bloques de un arreglo matricial.

## MEMORIAS DINÁMICAS (DRAM)

### • Célula básica

- Pequeño condensador integrado MOS
- Cargado = "1"
- Descargado = "0"
- Ventajas:
  - Muy alto nivel de integración
  - Menor coste a partir de 256K
- Problema: descarga por fugas de corriente
- Solución: recarga periódica (refresco)

### • Proceso de refresco

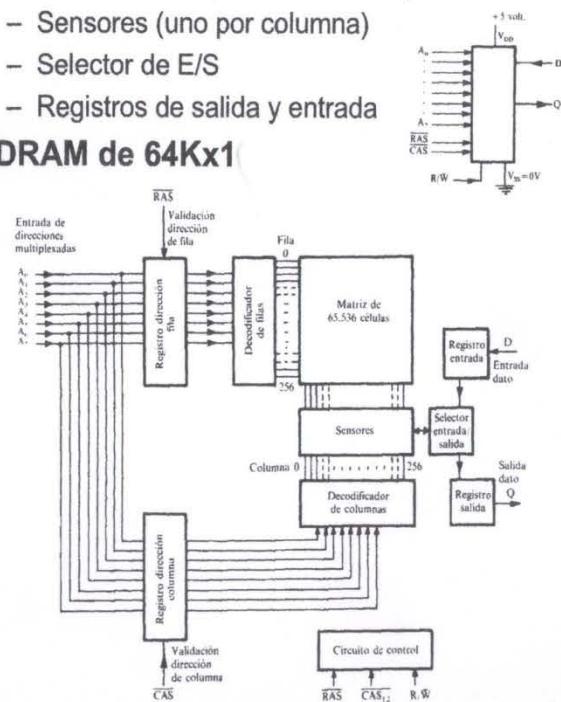
- Proceso: leer y, si está cargada, recargar
- Necesita circuitos adicionales e implica que la DRAM no está disponible todo el tiempo
- Intervalo de refresco típico: cada 2 mseg.
- Se refrescan a la vez filas o columnas gracias a la estructura matricial
- Cada fila o columna se puede considerar como una página de memoria que se procesa a la vez

## ELEMENTOS DE UNA DRAM (1)

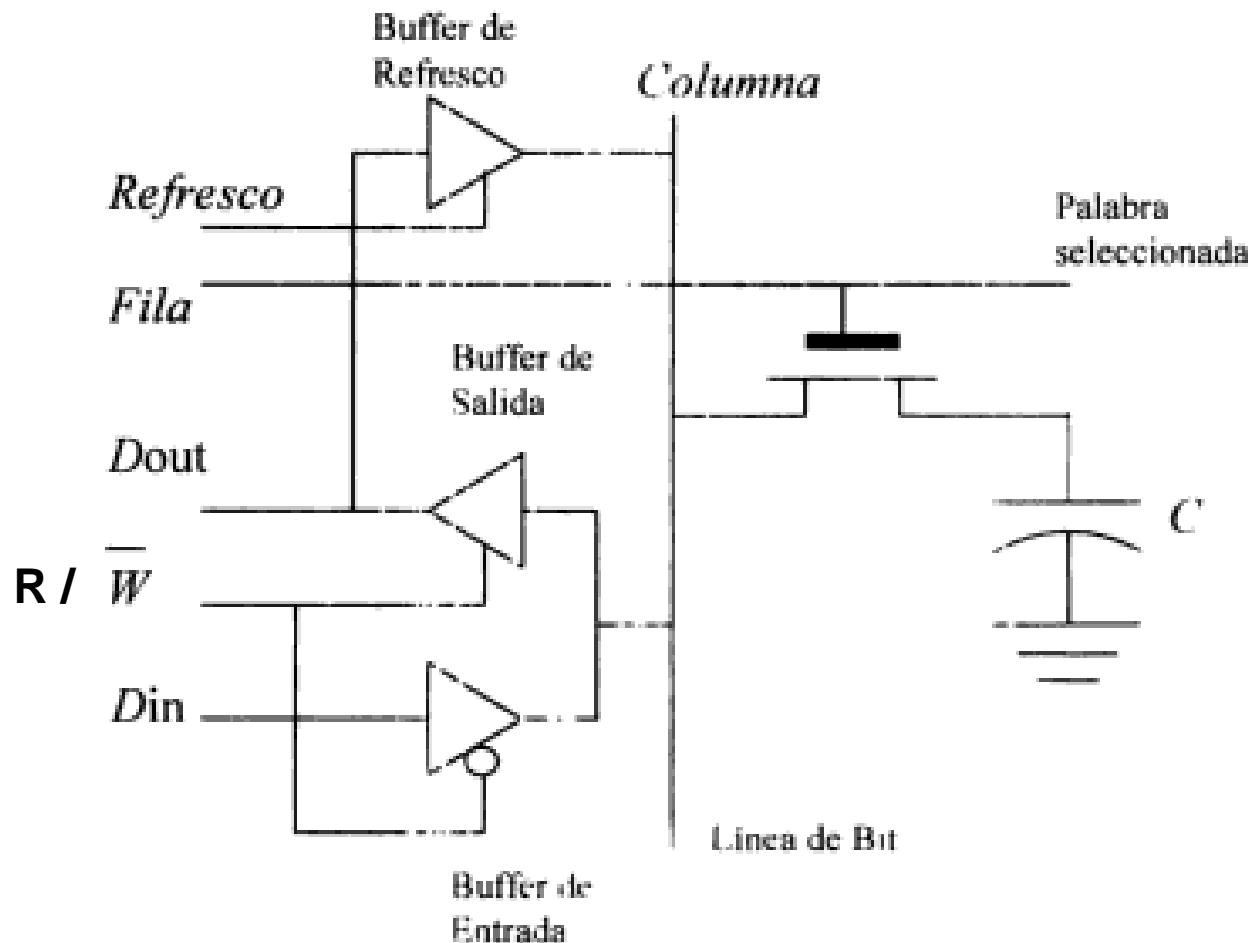
### • Elementos

- Registro de dirección de fila
- Registro de dirección de columna
- Decod de filas y decod de columnas
- Sensores (uno por columna)
- Selector de E/S
- Registros de salida y entrada

### • DRAM de 64Kx1

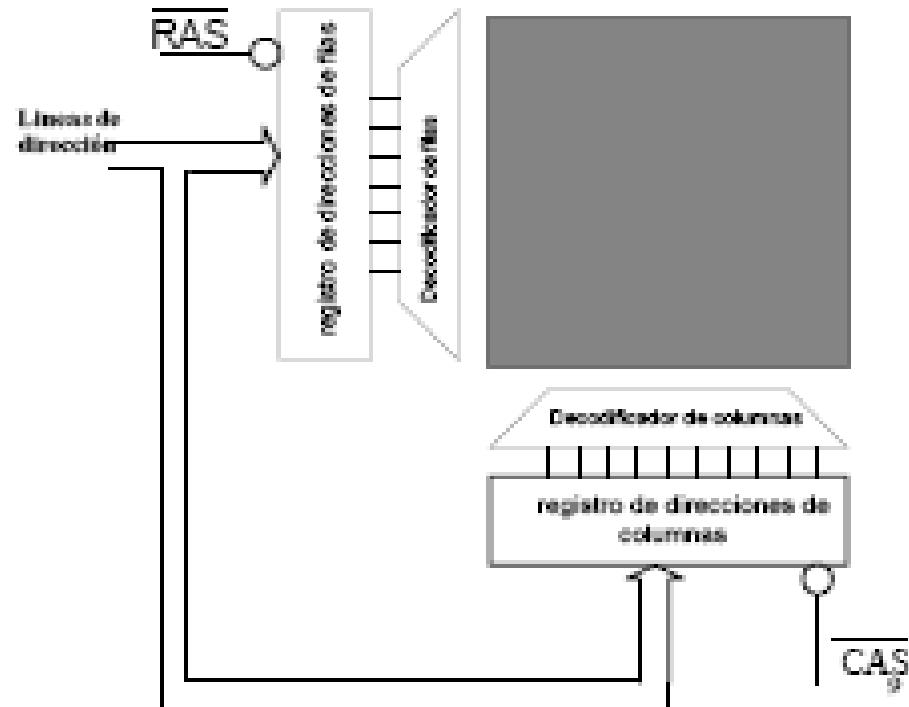


# Funcionamiento DRAM

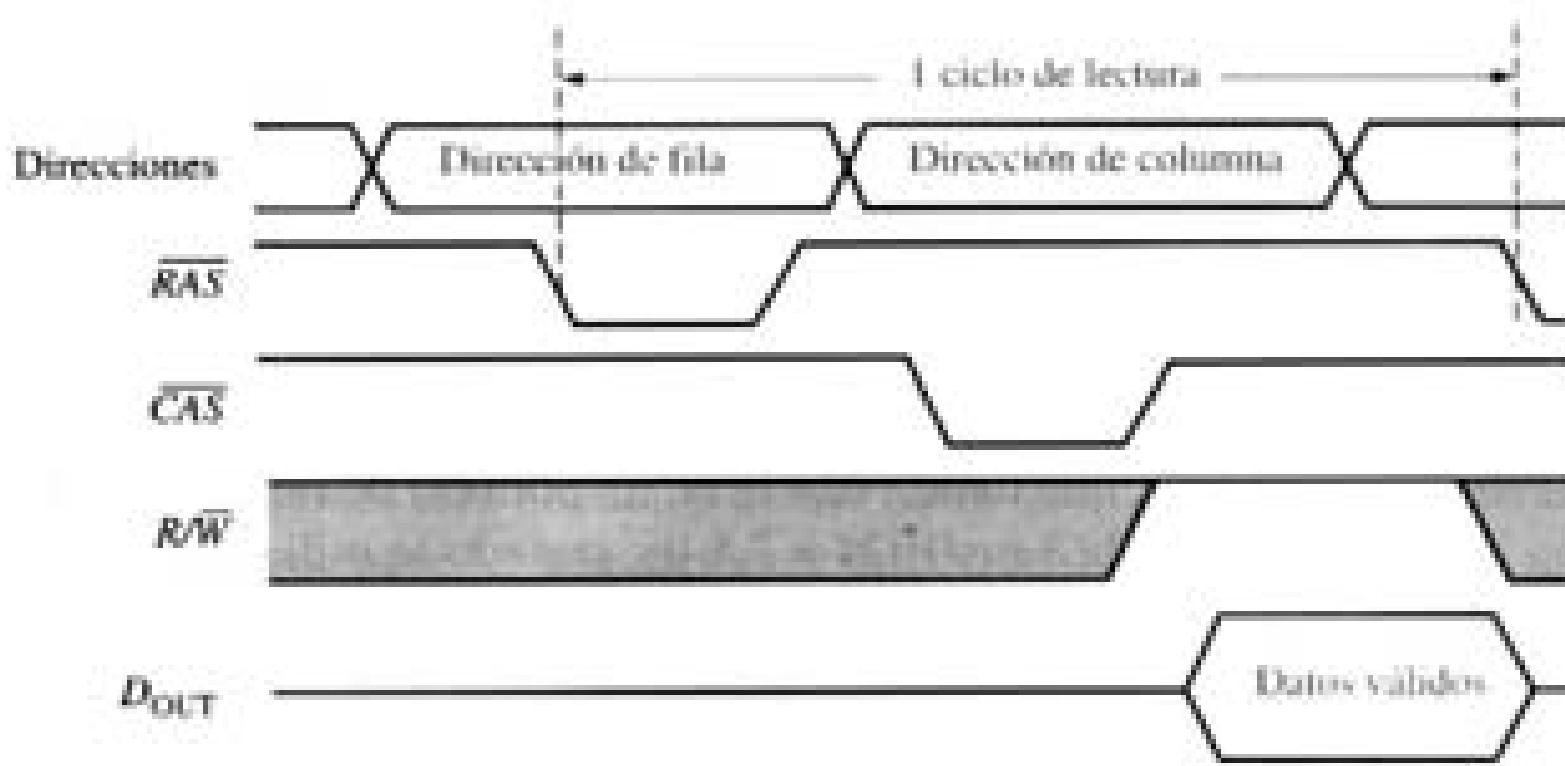


## Memorias DRAM

Las DRAM utilizan una técnica de **multiplexación del bus de direcciones** permitiendo reducir el número de líneas del bus.



# CICLO DE LECTURA DRAM



# Comparación SRAM \_ DRAM

Memoria	Ventajas	Desventajas
SRAM	La velocidad de acceso es alta	Menor capacidad, debido a que cada celda de almacenamiento requiere mas transistores
	Para retener los datos solo necesita estar energizada	Mayor costo por bit
	Son mas fáciles de diseñar	Mayor consumo de Potencia
DRAM	Mayor densidad y capacidad	La velocidad de acceso es baja
	Menor costo por bit	Necesita recargar de la información almacenada para retenerla
	Menor consumo de potencia	Diseño complejo

## **MEMORIAS DE SOLO LECTURA (ROM) CLASIFICACION**

---

- - Memorias ROM (Read Only Memory)
- - Memorias PROM (Programable ROM)
  
- - **Memorias RPROM (Reprogramable ROM):** Es posible reprogramarlas borrando el contenido previamente. Según la forma de realizar el borrado, se contempla una subclasificación adicional:
  1. - Memorias EPROM (Erasable PROM):
  2. - Memorias EEPROM o E2PROM (Electrically EPROM):

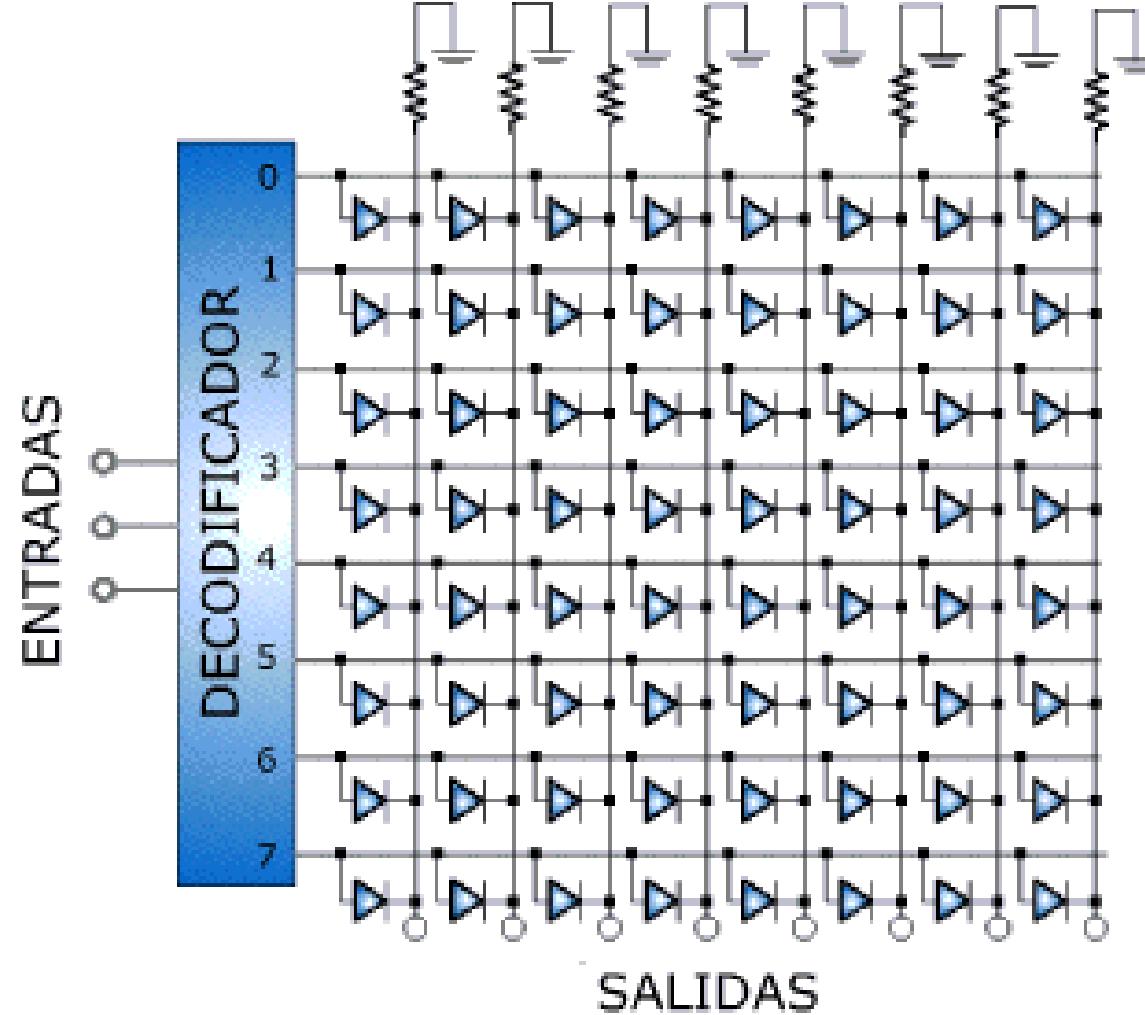
# Algunas Aplicaciones

---

- - Conversores de código
- - Generadores de caracteres
- - Func. aritméticas complejas (trigonométricas, logarítmicas)
- - Secuenciales de propósito general
- - Unidades de control microprogramadas
- - Almacenamiento de partes del sistema operativo.
- **COMPARACIÓN con las memorias RAM de lectura/escritura.**
- - La circuitería de direccionamiento es igual (uso de decodificadores)
- - El bloque de E/S se simplifica (sólo buffers de salida).
- - Las líneas de control quedan reducidas a CS (Chip Select).
- - Son no volátiles.

# ROM

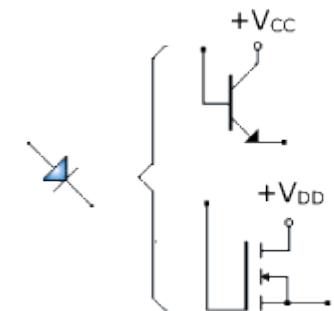
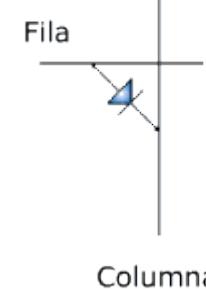
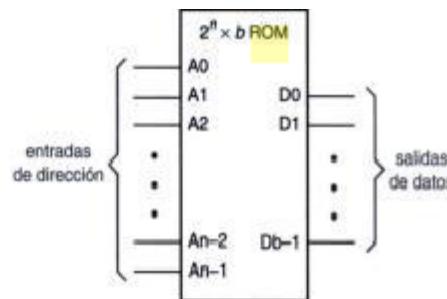
El contenido se establece en el proceso de fabricación.  
(Mascara)



# PROM

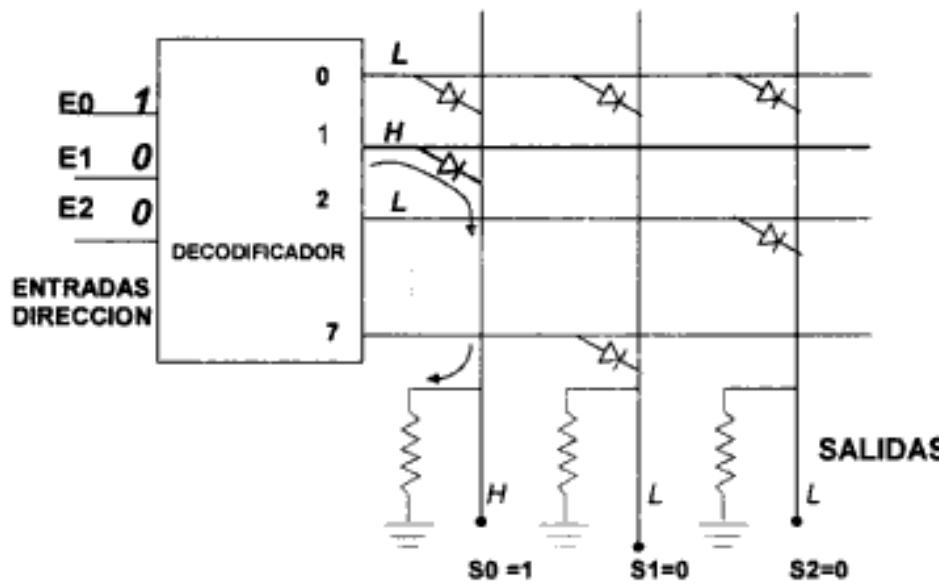
Son memorias ROM programables en un equipo especializado. El contenido es inalterable desde el momento de la programación.

Figura 10-1  
Estructura básica de una ROM de  $2^n \times b$ .



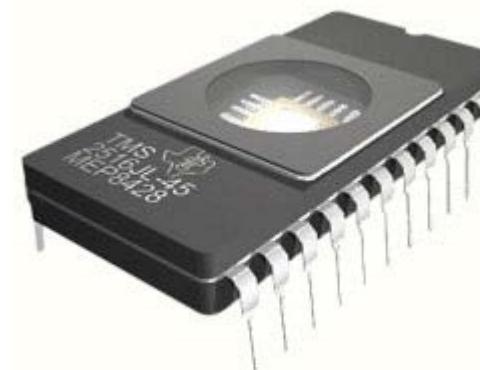
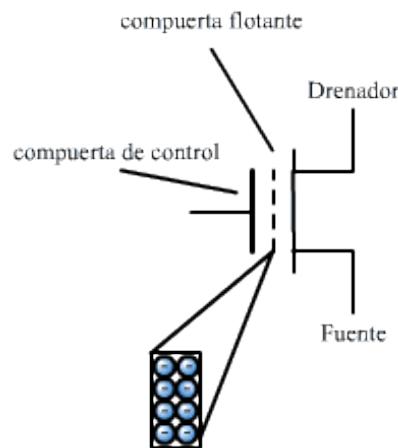
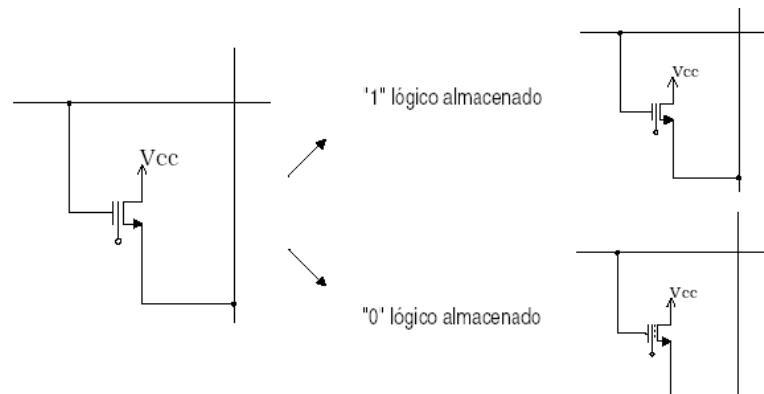
TTL

MOS



# EPRON (FLIA 2700 EJ. 2764)

- Necesita programador  $V_{pp}=12$  a 25 volt
- Borrado por luz ultravioleta
- Tiempo de borrado entre 10 30 minutos
- Borrado total



1. La grabación se realiza en equipos especiales. El borrado se realiza mediante la exposición del integrado a radiación ultravioleta.

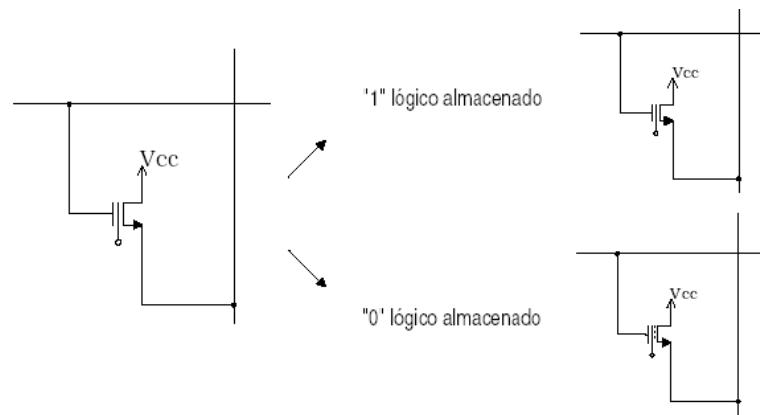
# EEPROM (28C64A) 8K x 8

- La programación y el borrado pueden realizarse sin la necesidad de una fuente de luz UV y unidad programadora de PROM
- Las palabras almacenadas en memoria se pueden borrar de forma individual.
- Para borrar la información no se requiere luz ultravioleta.
- El tiempo de borrado total se reduce a 10ms en circuito donde su antepasado inmediato requería media hora bajo luz ultravioleta externa.
- El tiempo de programación depende de un pulso por cada palabra de datos de 10 ms, versus los 50 ms empleados por una ROM programable y borrible.
- Se pueden reescribir aproximadamente unas 1000
- Para reescribir no se necesita hacer un borrado previo.

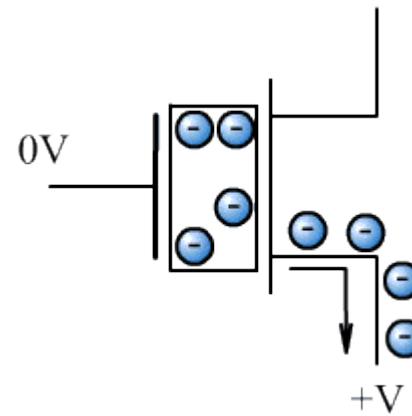
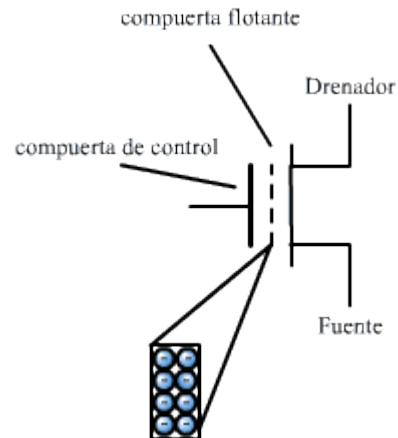
CE = CHIP ENABLE: Permite Activar el Circuito Integrado

OE = OUTPUT ENABLE: Permite Activar La Salida Del Bus De Datos

	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$
LECTURA	0	0	1
ESCRITURA	0	1	0



# Memoria Flash



La lectura se efectúa aplicando una tensión positiva a la compuerta de control de la celda de memoria, en cuyo caso el estado lógico almacenado se deduce con base en el cambio de estado del transistor:

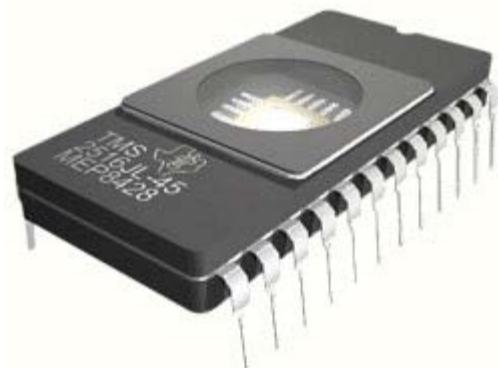
Si hay un 1 almacenado, la tensión aplicada será lo suficiente para encender el transistor y hacer circular corriente del drenador hacia la fuente.

Si hay un 0 almacenado, la tensión aplicada no encenderá el transistor debido a que la carga eléctrica almacenada en la compuerta aislada.

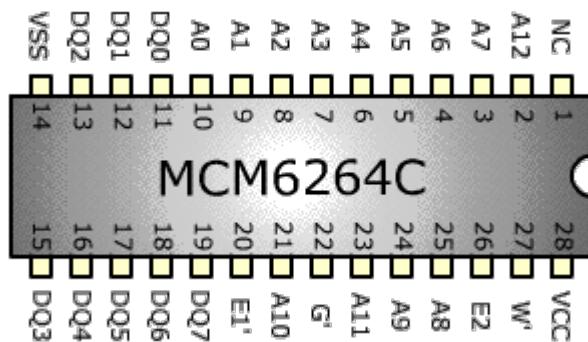
Para determinar si el dato almacenado en la celda es un 1 ó un 0, se detecta la corriente circulando por el transistor en el momento que se aplica la tensión en la compuerta de control. El borrado consiste en la liberación de las cargas eléctricas almacenadas en las compuertas aisladas de los transistores. Este proceso consiste en la aplicación de una tensión lo suficientemente negativa que desplaza las cargas como se indica en la figura 2

# Resumen

TIPO	CATEGORIA	BORRADO	ALTERABLE POR BIT
ROM	Solo R	imposible	NO
PROM	Solo R	impossible	NO
EPROM	PRIN. R	UV	NO
EEPROM	PRIN. R	ELECTRICO	SI
FLASH	R\W	ELECTRICO	NO PAG



# Ejemplos de Memorias Comerciales I



## Características Técnicas

**Referencia**

MCM6264C

**Tipo**

SRAM

**Capacidad (bits)**

8192 X 8

**Tipo de salida**

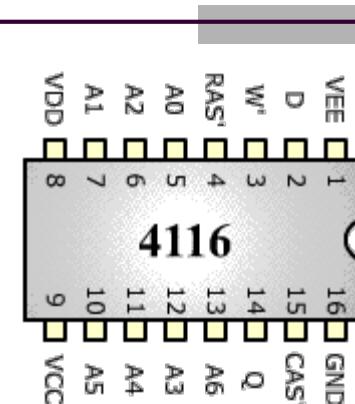
5V

**Tiempos de Acceso**

12/15/20/25/35 ns

**Encapsulado**

DIL-28



## Características Técnicas

**Referencia**

4116

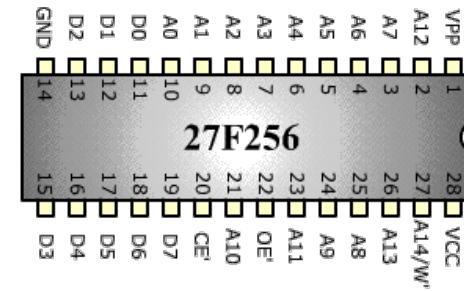
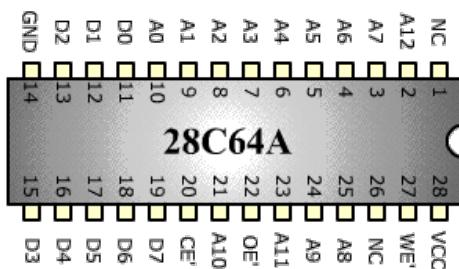
**Tipo**

DRAM

**Capacidad (bits)**

16384 X 1

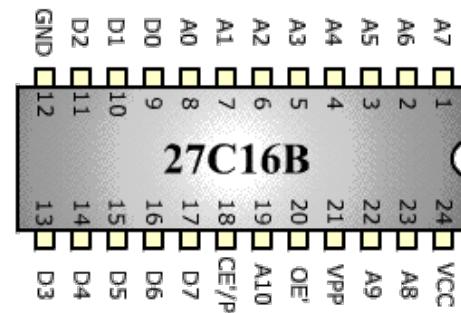
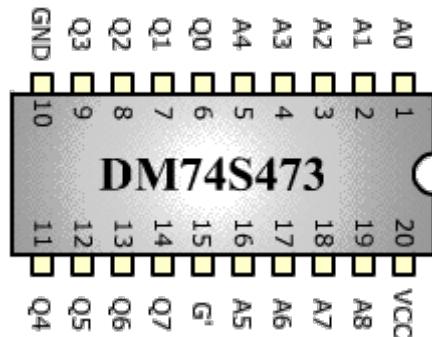
# Ejemplos de Memorias Comerciales II



- Características Técnicas
- Referencia      **28C64A**
- Tipo              **EEPROM CMOS**
- Capacidad (bits) **8192 X 8**
- Tiempos de Acces **120/150/200 ns**
- Encapsulado **DIL-28 y PLCC-32**

- Características Técnicas
- Referencia      **28F256**
- Tipo              **FLASH EEPROM**
- Capacidad (bits) **32768 X 8**
- Tiempos de Acceso **90/100/120/150 ns**
- Encapsulado      **DIL-28**

# Ejemplos de Memorias Comerciales III



- **Características Técnicas**
- **Referencia**
- 74S473
- **Tipo PROM**
- **Capacidad (bits)** 512 X 8
- **Tiempos de Acceso** 60 ns
- **Encapsulado** DIL-20

- **Características Técnicas**
- **Referencia**
- 27C16B
- **Tipo EPROM CMOS**
- **Capacidad (bits)** 2048 X 8
- **Tiempos de Acceso** 150/250 ns
- **Encapsulado**
- DIL-24

# Memorias Serie

## Características

### 1.- **Memorias Serie**

interesantes para ahorrar espacio y/o recursos.

### 2. **Número de pines muy reducido:**

alimentación, entrada/salida de datos en serie,  
control/selección de chip.

### 3. **Interfaces serie de muy diversa naturaleza:**

1Wire, SPI, I2C, custom ...

Lentas por su naturaleza.

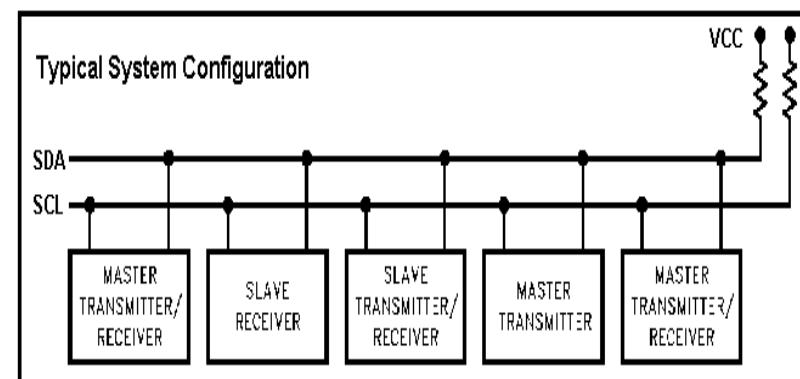
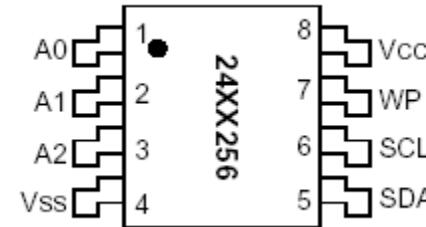
### 4. **Idóneas para guardar datos de acceso eventual**

En consonancia, la mayor parte de ellas son No  
Volátiles: **EEPROM, FLASH, etc.**

# Memorias serie 24XX256

## EEPROM

- Es una memoria de 256 Kbits organizada es a 32 k \*8
- Soporta protocolo de transmisión de datos en bus de dos líneas.
- El bus se controla mediante un **master** que se encarga de generar el reloj ( **SCL** ), el control de acceso al bus y además genera las condiciones de **START** y **STOP** mientras el **24lc25616B** actua como **slave**.
- El rango de Vcc 1.8-5.5 V .
- Max frecuencia del clock 400 khz.



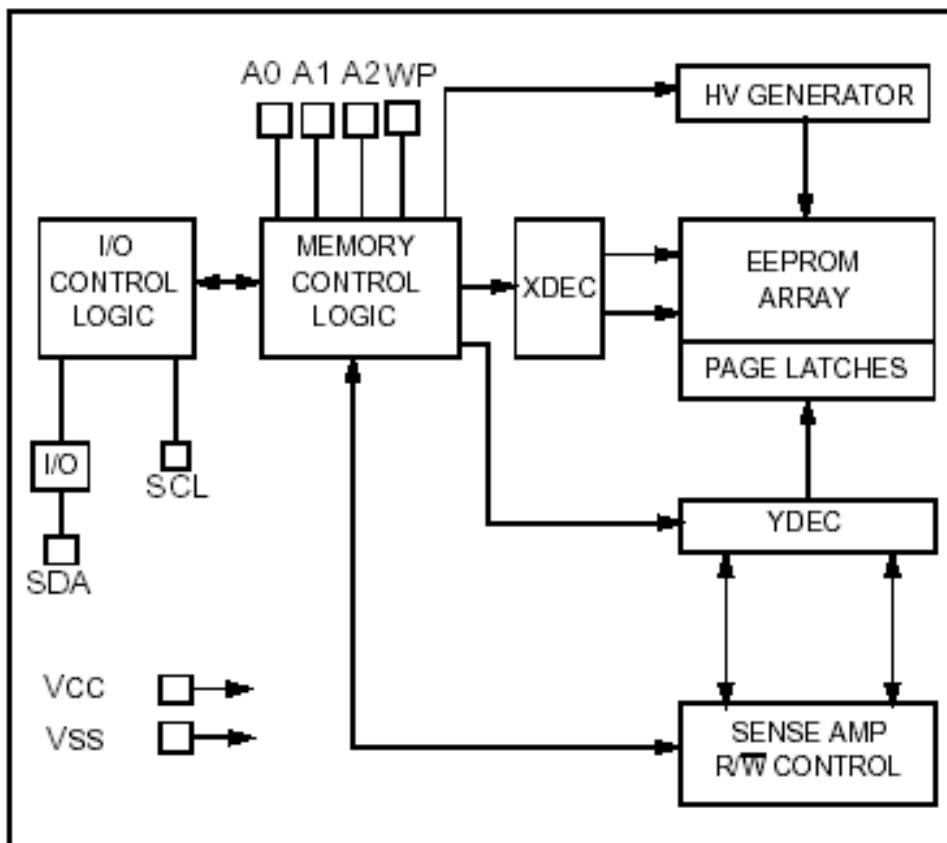
# Memorias serie 24XX256

## Descripción de los pines

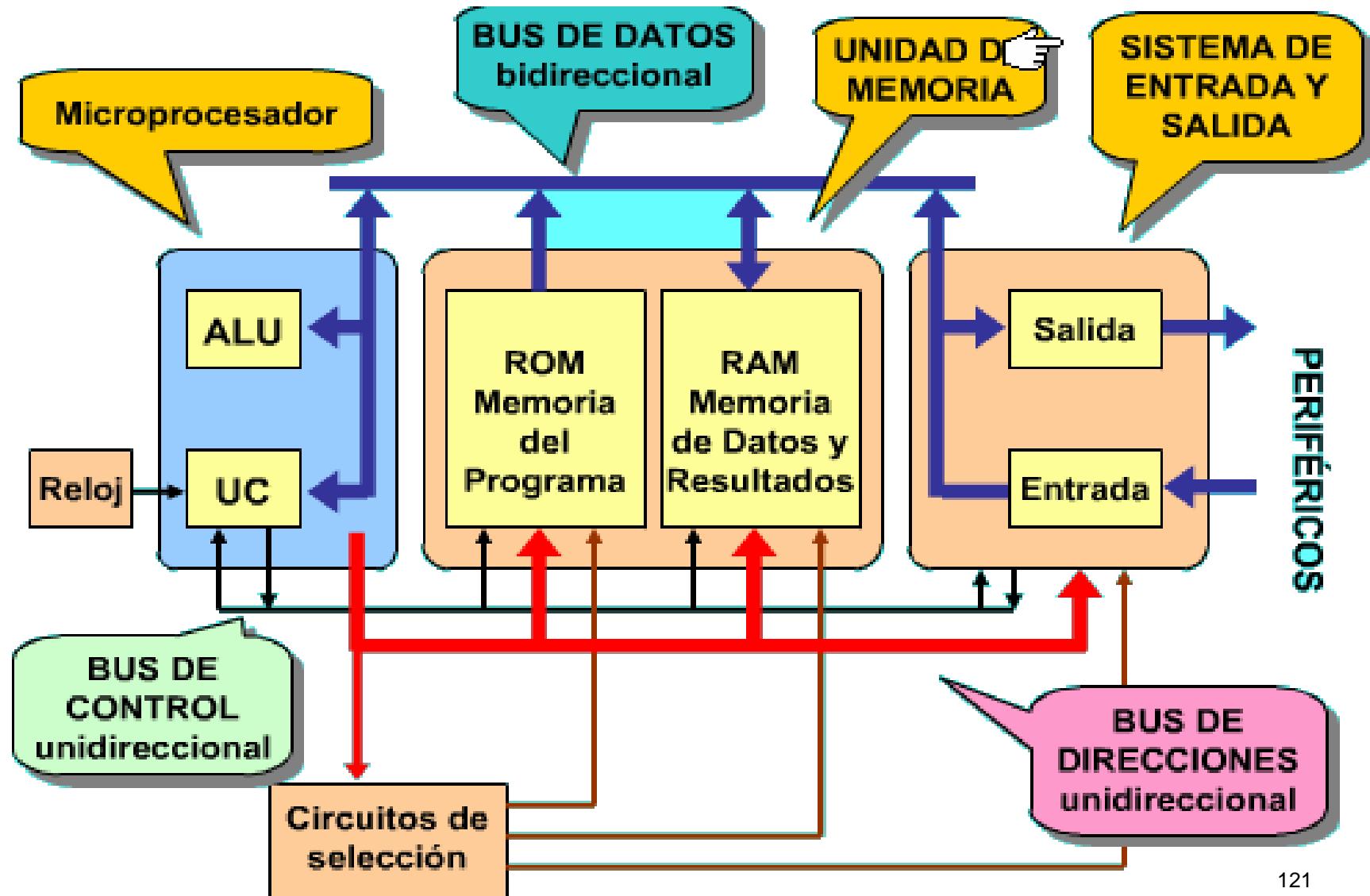
- **AO, A1,A2** son los pines de entrada, la cual es usada por el 24C256 para multiples operaciones. Los niveles de entrada son comparados con los bits correspondientes en la dirección del esclavo. El chip selecciona si la comparación es verdad.
- **SDA Data serial.** Es un pin bidireccional (entrada y salida de data). Para habilitar el pin de SDA debe tener conectado una resistencia del pin (SDA) a VCC ( 10 kΩ desde 100 khz y 2kΩ desde 400 khz y 1 khz.). SDA puede estar habilitado solo cuando el pin SCL esta en nivel bajo. La cual puede trabajar en dos condiciones START o STOP.
- **SCL SERIAL CLOCK.** Sincroniza la transferencia de datos del dispositivo.
- **WP.** Es de selección de escritura y lectura va conectado a tierra o ha Vcc. Puede leer o escribir desde 0000- hasta 7FFF.

# Memorias serie 24XX256

## Diagrama de bloque

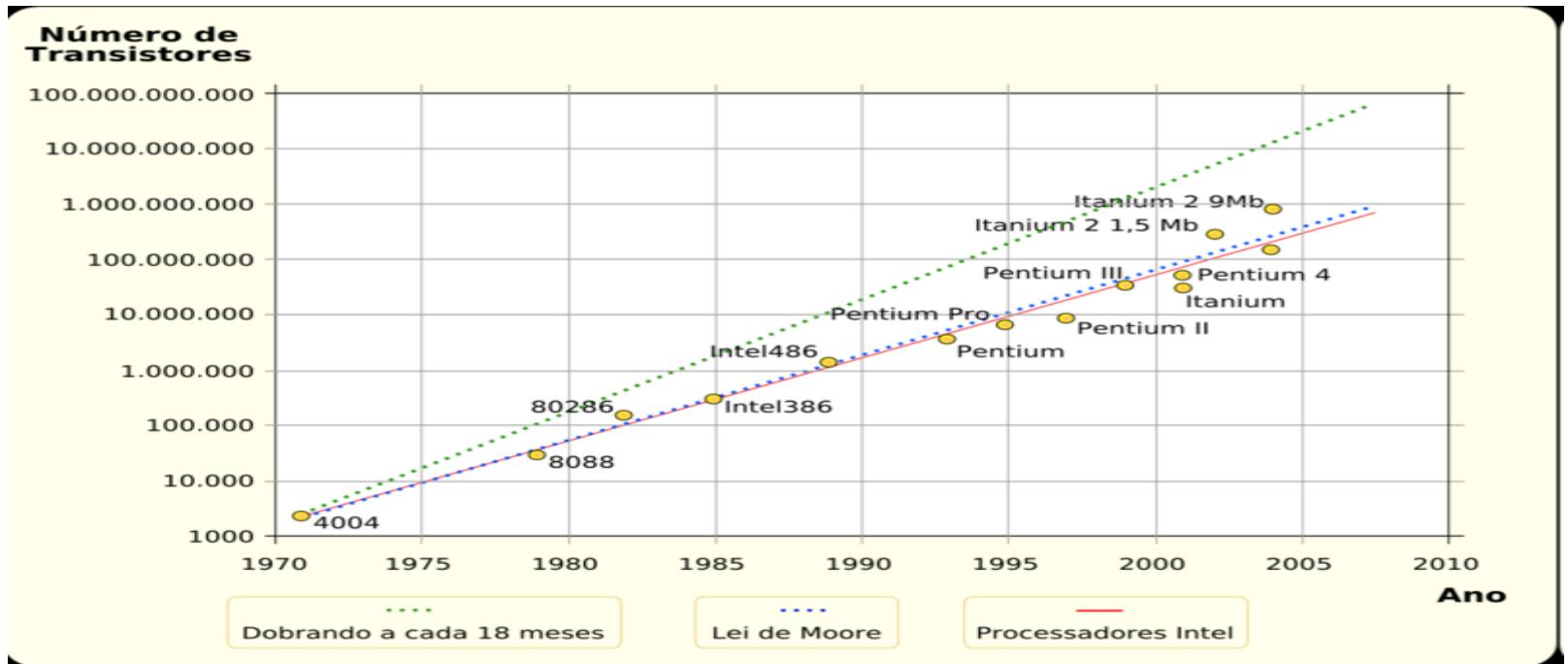


# Resumen



# Ley de Moore

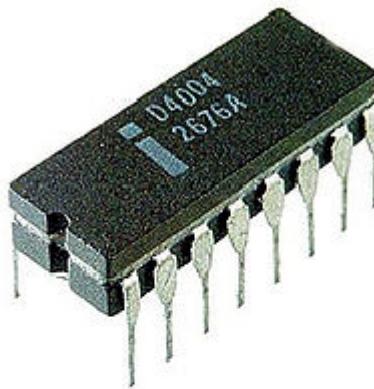
Expresa que aproximadamente cada dos años se duplica el número de transistores en un circuito integrado



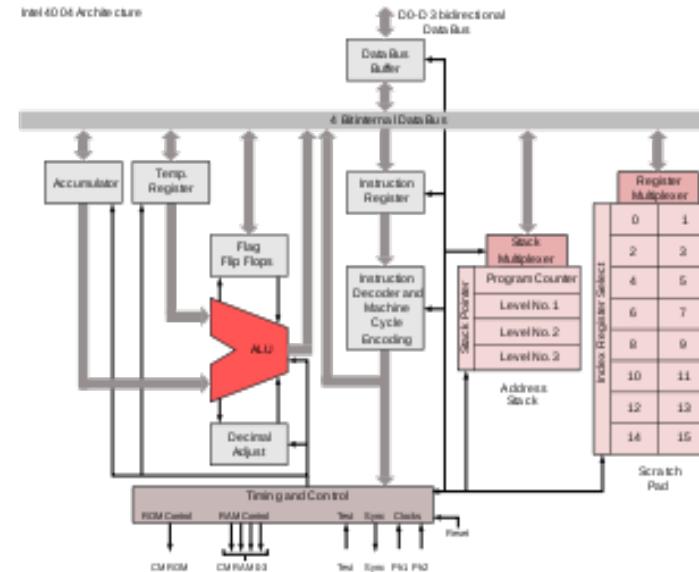
La ley de Moore expresa que aproximadamente cada dos años se duplica el número de transistores en un microprocesador. A pesar de que la ley originalmente fue formulada para establecer que la duplicación se realizaría cada 1 año, posteriormente Moore redefinió su ley y amplió el periodo a dos años.

# Intel 4004

CPU de 4bits, fue el primer microprocesador en un chip, así como el primero disponible comercialmente

Intel 4004	
Microprocesador	
	Microprocesador Intel 4004
Producción	Finales de 1971 — 1981
Fabricante(s)	Intel
Frecuencia de reloj de CPU	740 kHz
Conjunto de instrucciones	pre x86
Package(s)	16 pin DIP
[editar datos en Wikidata]	

- Microprocesador de 4 bits
- Contiene 2.300 transistores
- Encapsulado de 16 pines
- Máxima velocidad del reloj 740 KHz
- Usa Arquitectura Harvard,
  - 12 bits de direcciones (direccinando hasta 4 KB)
  - Instrucciones de 8 bits de ancho, que no deben ser colocadas en la misma memoria de datos de 4 bits de ancho.
- El conjunto de instrucciones está formado por 46 instrucciones
- 16 registros de 4 bits cada uno



# Intel 4004

---

- Intel fabrica el primer microprocesador (el 4004) de tecnología PMOS. Este era un microprocesador de 4 bits y fue fabricado por Intel a petición de Datapoint Corporation con el objeto de sustituir la CPU de terminales inteligentes que eran fabricadas en esa fecha por Datapoint mediante circuitería discreta. El dispositivo fabricado por Intel resultó 10 veces más lento de lo requerido y Datapoint no lo compró, de esta manera Intel comenzó a comercializarlo. El 4004 era un microprocesador de 4 bits, contenía 2,300 transistores y corría a 108 KHz podía direccionar sólo 4096 (4k) localidades de memoria de 4 bits, reconocía 45 instrucciones y podía ejecutar una instrucción en 20 µseg en promedio. Este procesador se utilizó en las primeras calculadoras de escritorio.

**1972**

---

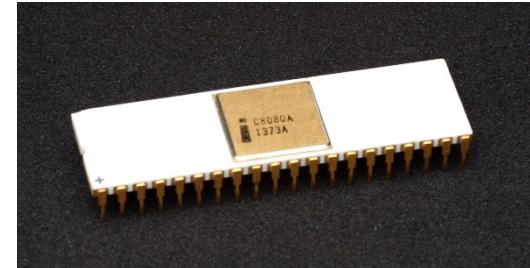
- Las aplicaciones del 4004 estaban muy limitadas por su reducida capacidad y rápidamente Intel desarrolló una versión más poderosa (el 8008), el cual podía manipular bytes completos, por lo cual fue un microprocesador de 8 bits. La memoria que este podía manejar se incrementó a 16 kbytes, sin embargo, la velocidad de operación continuó igual.

# i8051



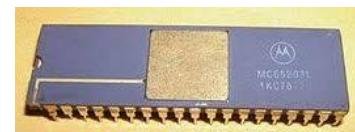
- El 8051 es un microcontrolador de 8 bits desarrollado en su origen por Intel en 1980. Es el microcontrolador más popular del mundo, hecho por varios fabricantes independientes. ¡Tan solo en 1993 se vendieron 126 millones de 8051s (y variantes) !  
Un típico 8051 contiene:
  - CPU con procesador booleano
  - 5 o 6 Interrupciones:
    - 2 son externas
    - 2 niveles de prioridad
  - 2 o 3 temporizadores/contadores de 16 bits
  - puerto serie full-duplex programable
    - (con la velocidad en baudios provista por uno de los timers)
  - 32 líneas de entrada - salida (cuatro puertos de 8 bits)
  - RAM
  - En algunos modelos, ROM/EPROM

# Intel 8080



- El **Intel 8080** fue un microprocesador diseñado y fabricado por Intel.
- La CPU de 8 bits fue lanzado en abril de 1974.
- Corría a 2 MHz, y generalmente se le considera el primer diseño de CPU microprocesador verdaderamente usable.
- Varios fabricantes importantes fueron segundas fuentes para el procesador, entre los cuales estaban AMD, Mitsubishi, NatSemi, NEC, Siemens, y Texas Instruments.

# Motorola MC6800 (6800)



- Es un microprocesador fabricado por Motorola que fue lanzado al mercado en, 1975
- El nombre le viene de contener aproximadamente 6800 transistores.
- Su conjunto de instrucciones está formado por 78 de estas compatibles con 8008 y agregado el manejo de punteros
- El 6800 normalmente se fabricaba en un encapsulado DIP de 40 PINES
- Partiendo del 6800 se crearon varios procesadores derivados, siendo uno de los más potentes el Motorola 6809, que fue usado en el sistema de videojuego Vectrex y en el ordenador Tandy TRS-80, entre otros. También se han producido varios microcontroladores basados en el 6800.

# Zilog Z80 ( 1976)



- Es un microprocesador de 8 bits cuya arquitectura se encuentra a medio camino entre la organización de acumulador y de registros generales.
- Si consideramos al Z80 como procesador de arquitectura de registros generales, se sitúa dentro del tipo de registro-memoria.
- Fue lanzado al mercado en julio de 1976 por la compañía Zilog, y se popularizó en los años 80 a través de ordenadores como el Sinclair ZX Spectrum, Amstrad CPC