

# MICRONTROLADORES

# INTRODUCCIÓN

- Los microcontroladores son circuitos integrados que son capaces de ejecutar ordenes grabadas previamente en su memoria.
- El termino microcontrolador esta dado por dos palabras que son “Micro”-“Controlador” las cuales tienen por significado “pequeño (en tamaño)” y “maniobrar o controlar (función principal)” procesos los cuales son definidos mediante la programación.
- Se componen por diferentes bloques funcionales, que cumplen una tarea especifica, son dispositivos que operan uno o mas procesos.

# INTRODUCCIÓN

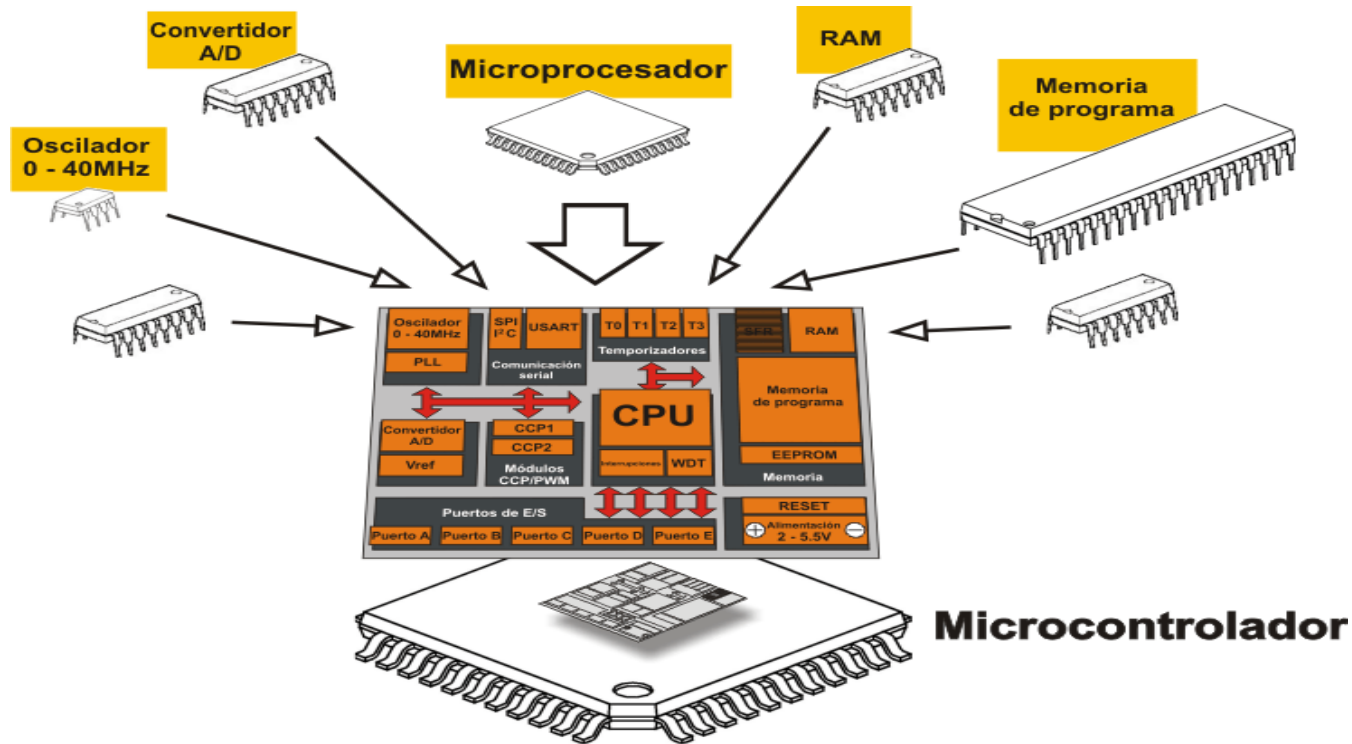
Los micro controladores esta constituido en su interior por las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada y salida.

Un micro controlador es un sistema completo, con prestaciones limitadas que no pueden modificarse y lleva a cabo las tareas para las que ha sido programado de forma autónoma.

# DIFERENCIAS ENTRE MICROCONTROLADOR Y MICROPROCESADOR

- FUNCIONALIDAD.
- COMPONENTES REQUERIDOS PARA SU FUNCIONAMIENTO.
- CAPACIDAD DE CALCULO.
- REQUERIMIENTOS DE CIRCUITOS ESPECIALES.
- CONEXIÓN CON PERIFÉRICOS.
- USO GENERAL Y ESPECIFICO.

# DIFERENCIAS ENTRE MICROCONTROLADOR Y MICROPROCESADOR



## DESARROLLO HISTORICO.

- Los ingenieros Gary Boone y Michael Cochran desarrollan el primer microcontrolador (TMS 1000) en 1971, comercializado en 1974. Incluía memoria ROM, RAM, un microprocesador y una fuente de reloj en un mismo chip.
- 1975: Primeros PIC (PIC1650). Desarrollado en General Instrument. Usaba microcodigo almacenado en ROM.
- 1976: Intel lanza el 8048. Aquitectura Harvard modificada, memoria ROM y entradas-salidas con espacio propio en memoria.

## DESARROLLO HISTORICO.

- Intel desarrollo el 8048 en 1977, diseñado para aplicaciones de control <sup>1</sup>.
- 8051 de Intel, desarrollado para el uso en sistemas embebidos (1980).
- Tercera generación de micro controladores, el Intel 80c196 de 16 bits. Unidad de multiplicación, división, canales de direccionamiento de alta velocidad y control de interrupciones programables.

<sup>1</sup>[http://web.archive.org/web/20121202001529/http://archive.computerhistory.org/resources/access/text/Oral\\_History/102658328.05.01.acc.pdf](http://web.archive.org/web/20121202001529/http://archive.computerhistory.org/resources/access/text/Oral_History/102658328.05.01.acc.pdf)

## DESARROLLO HISTORICO.

- Usaban memorias EPROM (*Erasable Programmable Read-Only Memory*)Y PROM (*programmable read-only memory*).
- 1993: Uso de memorias EEPROM en un microcontrolador (PIC16x18).
- 1993: ATMEL lanza un microcontrolador con memoria FLASH.
- 1994: Intel introduce 80386EX, diseñado como revisión del 386 y uso en sistemas embebidos.
- 2001: PICs wireless.



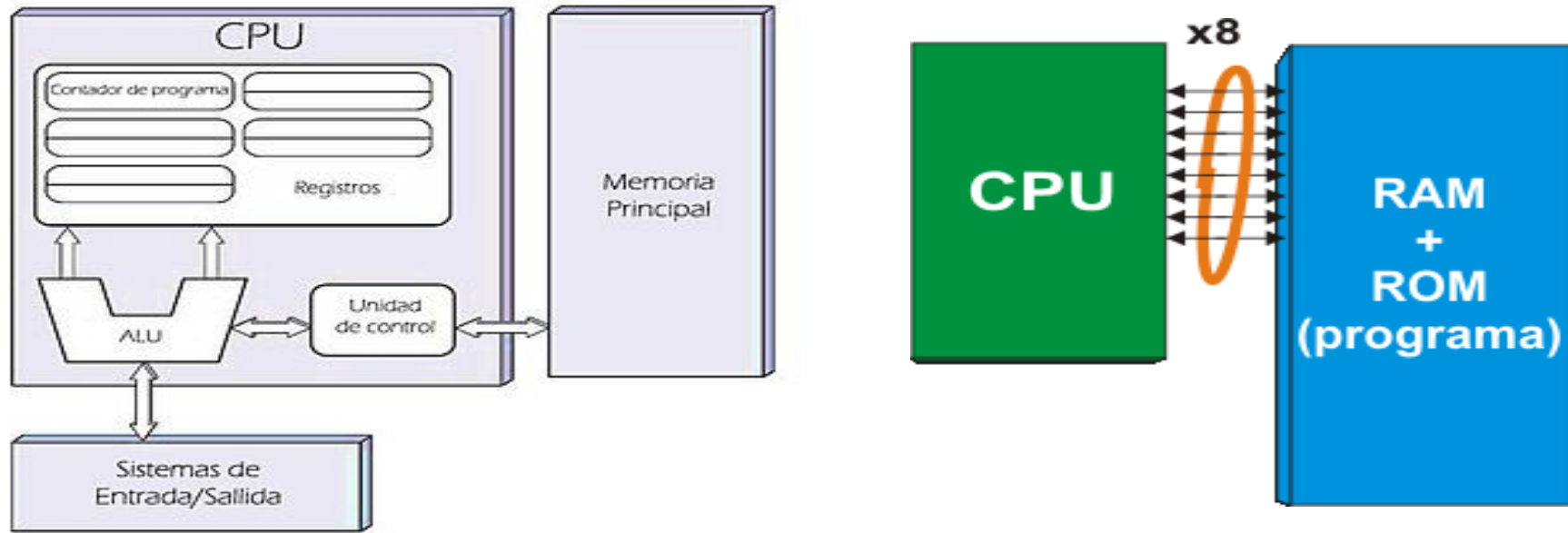
## DESARROLLO HISTORICO.

- 2003: Microchip lanza los DSPic. Usan un bus de 16 bits e incluyen todas las características desarrolladas por las familias anteriores de microcontroladores. Añaden hardware para operaciones DSP.
- 2007: Lanzamiento de los PIC32 bits por parte de Microchip, con velocidades de computo superiores a 1,5 DMIPS /MHz y modulo HOST USB.

*PIC Peripheral Interface Controller*

*DMIPS/ MHz Dhrystone MIPS (Million Instructions por Segundo)*

# ARQUITECTURA DE LOS MICROCONTROLADORES.



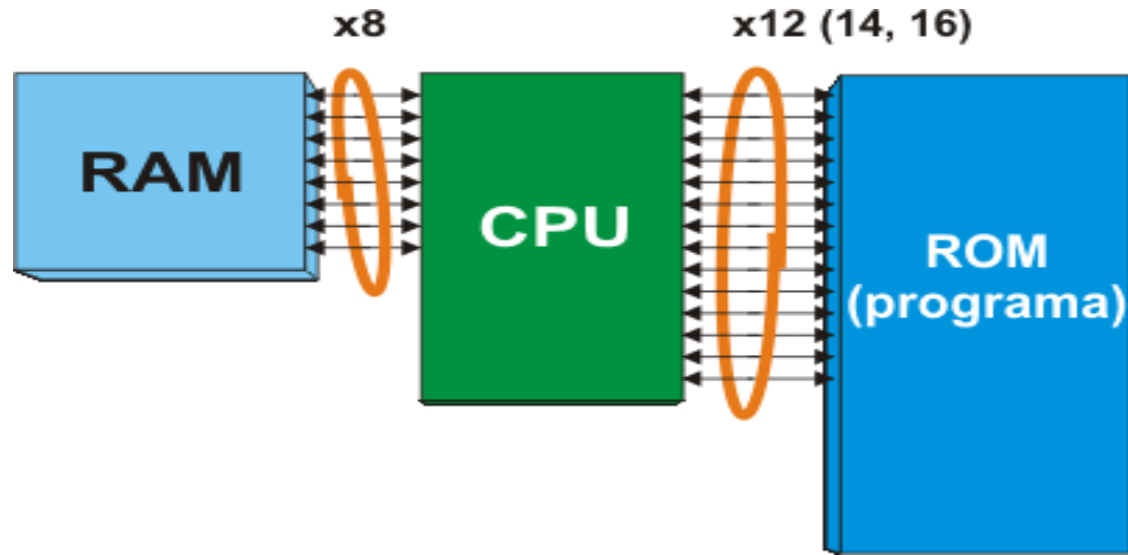
Arquitectura Von Neumann, usada por los primeros microcontroladores.

# ARQUITECTURA DE LOS MICROCONTROLADORES.

Desarrollo de una operación incremento *incf SUM*.

1. Leer la parte de la instrucción de programa que especifica la operación a realizar.
2. Leer de la instrucción el dato lo debe realizar la operación.
3. Realización del incremento y re escritura del contenido del registro del que se tomo el dato original.

# ARQUITECTURA DE LOS MICROCONTROLADORES.



*Arquitectura Harvard, usada por la mayoría de microcontroladores modernos.*

# ARQUITECTURA DE LOS MICROCONTROLADORES.

Todos los datos en el programa son de un byte. La longitud del bus de datos (12,14,16 bits) usa para la memoria de programa la instrucción y el dato se pueden leer simultáneamente.

La separación de las memorias hace posible que la CPU ejecute dos intrusiones simultáneamente. Esto se logra porque mientras se realiza la lectura o escritura de la RAM, la siguiente instrucción se lee por medio de otro bus.

Conocimiento de la memoria ocupada por un programa, porque una instrucción equivale a una posición de memoria.

## CICLO DE INSTRUCCIÓN (*fetch-and-execute*).

Periodo que tarda la unidad central de procesamiento en ejecutar una instrucción de lenguaje maquina.

En este periodo se comprende una secuencia de acciones preestablecidas que debe realizar la CPU para ejecutar una instrucción de un programa.

Todas las instrucciones no requieren el mismo numero de ciclo de instrucciones para su ejecución.

Un ciclo de instrucción requiere de uno o mas ciclo maquina.

# CICLO DE INSTRUCCIÓN (*fetch-and-execute*).

## FASE FETCH.

Buscar la instrucción en la memoria principal: Se direcciona la memoria de programa con el dato del contador de programa. La instrucción se guarda en el registro de instrucción actual (*CIR*).

# CICLO DE INSTRUCCIÓN (*fetch-and-execute*).

## FASE FETCH.

Decodificar la instrucción: Se mueve la instrucción de CIR al registro IR. Este registro mantiene la instrucción mientras se ejecuta y permite que se disecione otra interrupción. La unidad de control decodifica la operación a realizar y mueve los datos de memoria la unidad de procesamiento.



## CICLO DE INSTRUCCIÓN (*fetch-and-execute*).

### FASE EXECUTE.

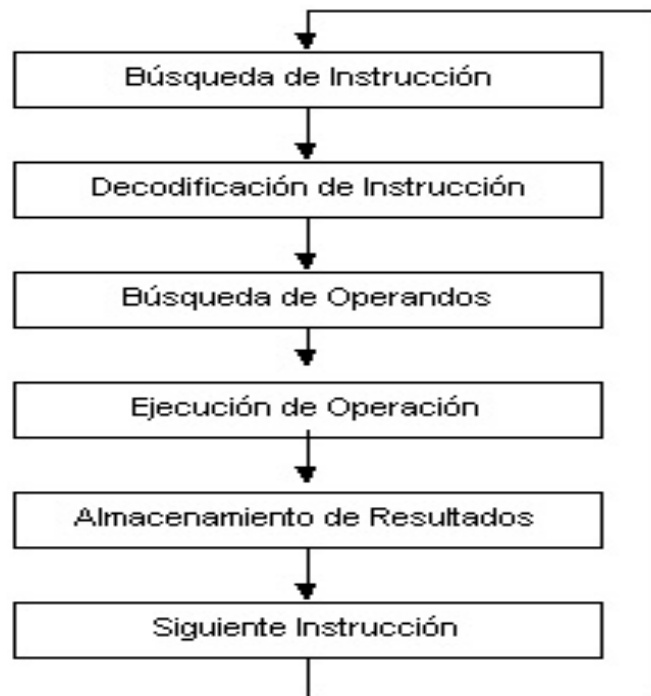
Ejecución de la instrucción: La unidad de control envía las señales de control a las unidades de procesamiento para que estas desarrollen la operación.

## CICLO DE INSTRUCCIÓN (*fetch-and-execute*).

### FASE EXECUTE.

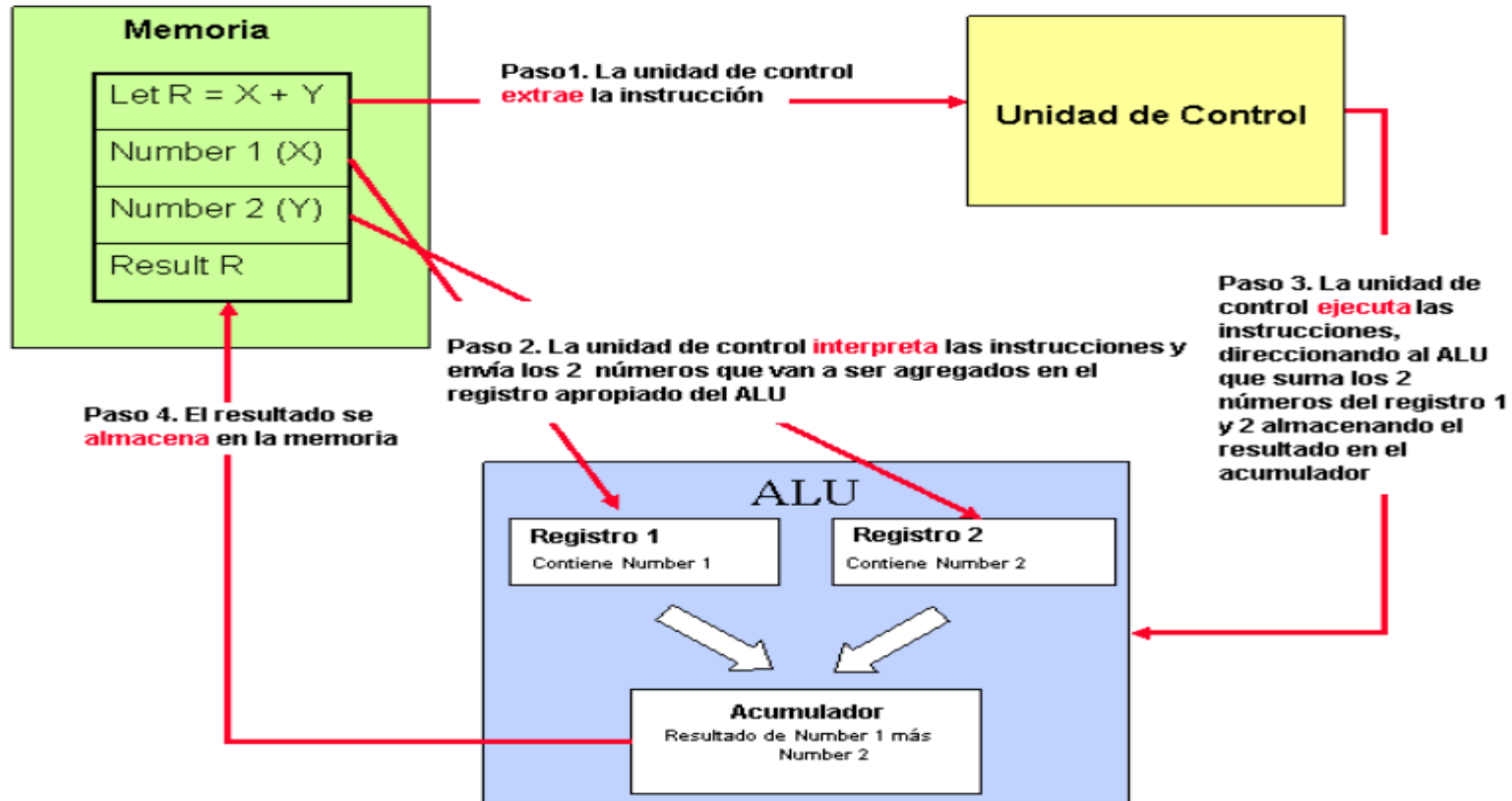
Almacenar o guardar resultados: El o los resultados son almacenados en memoria o enviados a los diferentes dispositivos. El contador de programa se incrementa o se modifica de acuerdo a los resultados obtenidos de la operación.

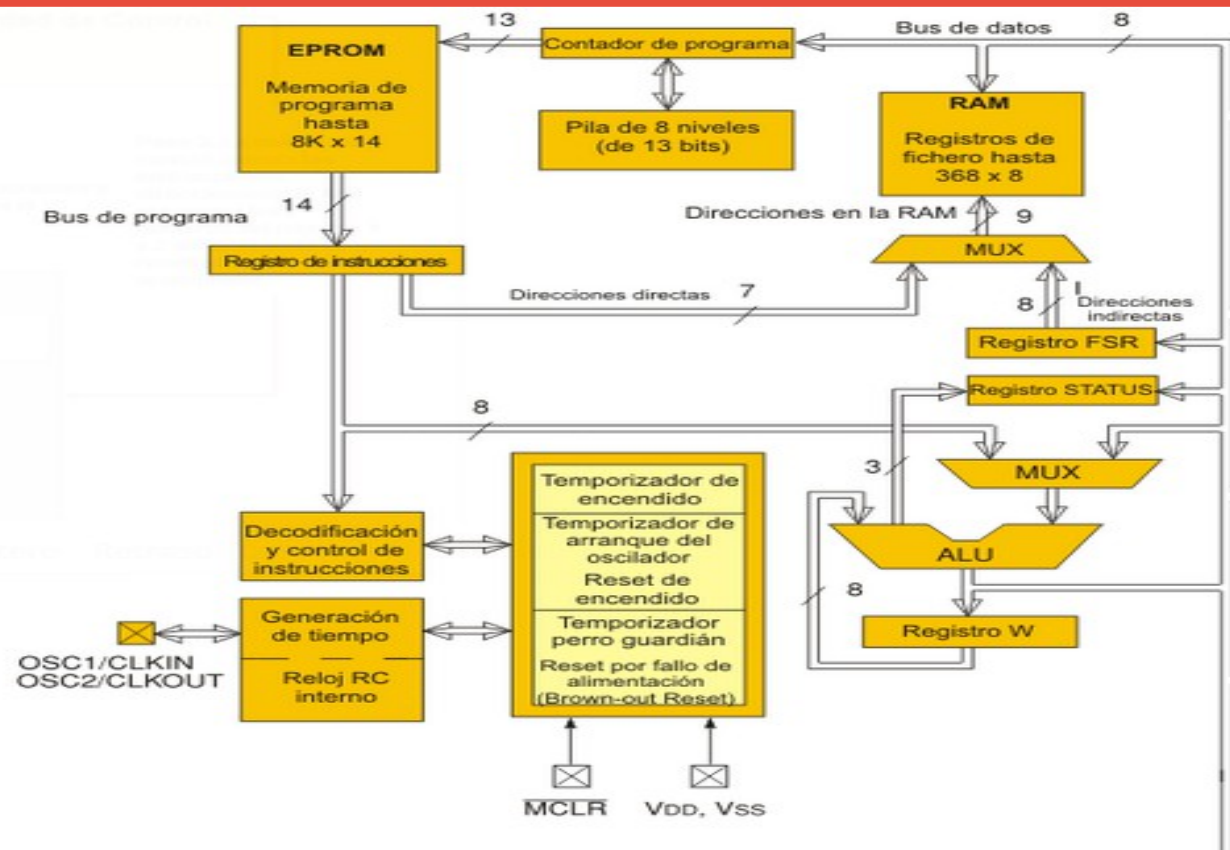
## CICLO DE INSTRUCCIÓN (*fetch-and-execute*).





# CICLO DE INSTRUCCIÓN (*fetch-and-execute*).





## REPERTORIO DE INSTRUCCIONES ISA.

Especificación detallada de las instrucciones u operaciones que una unidad central de procesamiento (*CPU*) puede entender y ejecutar.

Conjunto de comandos específicos implementados en un diseño de específico de CPU.

Son el conjuntos de especificaciones accesibles para el programador. Describe tipo de datos propios de cada *CPU*, interrupciones, distribución de la memoria, etc.

# REPERTORIO DE INSTRUCCIONES.

CISC (Complex Instruction Set Computer)

RISC (Reduced Instruction Set Computer)

SISC (Simple Instruction Set Computing).



## CISC (Complex Instruction Set Computer)

Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requieren múltiples ciclos para su ejecución y buscan ortogonalidad.

Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

En esta arquitectura se dificulta el paralelismo entre instrucciones, por las diferencias de ciclos de instrucciones entre esta.

## RISC (Reduced Instruction Set Computer)

El repertorio de instrucciones máquina es reducido y las instrucciones son simples y generalmente se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del processor.

Instrucciones de tamaño fijo y presentadas en un reducido número de formatos.

Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

Gran numero de registros de propósito general.

## SISC (Simple Instruction Set Computing)

Set de instrucciones destinado a aplicaciones muy concretas. El juego de instrucciones, es reducido, específico. Las instrucciones se adaptan a las necesidades de la aplicación prevista.

