Arquitectura de computadoras

Presenta: M. en C.: Baltazar Jiménez

¿Qué es una computadora?

- Es una máquina electrónica capaz de procesar datos a gran velocidad a partir de un grupo de instrucciones denominado **programa.**

¿Qué es un programa?

Es un conjunto de instrucciones codificadas que se almacenan en la memoria interna de la computadora, junto con todos los datos que requiere el programa.

Arquitectura de computadoras

- Se refiere a los atributos de un sistema que *son visibles a un programador* y a la estructura operacional fundamental de un sistema.

Arquitectura de computadoras

- Conjunto de instrucciones.
- Número de bits usados para representar los tipos de datos.
- Mecanismos de E/S.
- Técnicas para direccionamiento de memoria.

Organización

Se refiere a las unidades funcionales y a sus interconexiones, que dan lugar a especificaciones arquitectónicas.

Entre los atributos de organización se incluyen aquellos detalles de hardware *transparentes hacia el programador*:

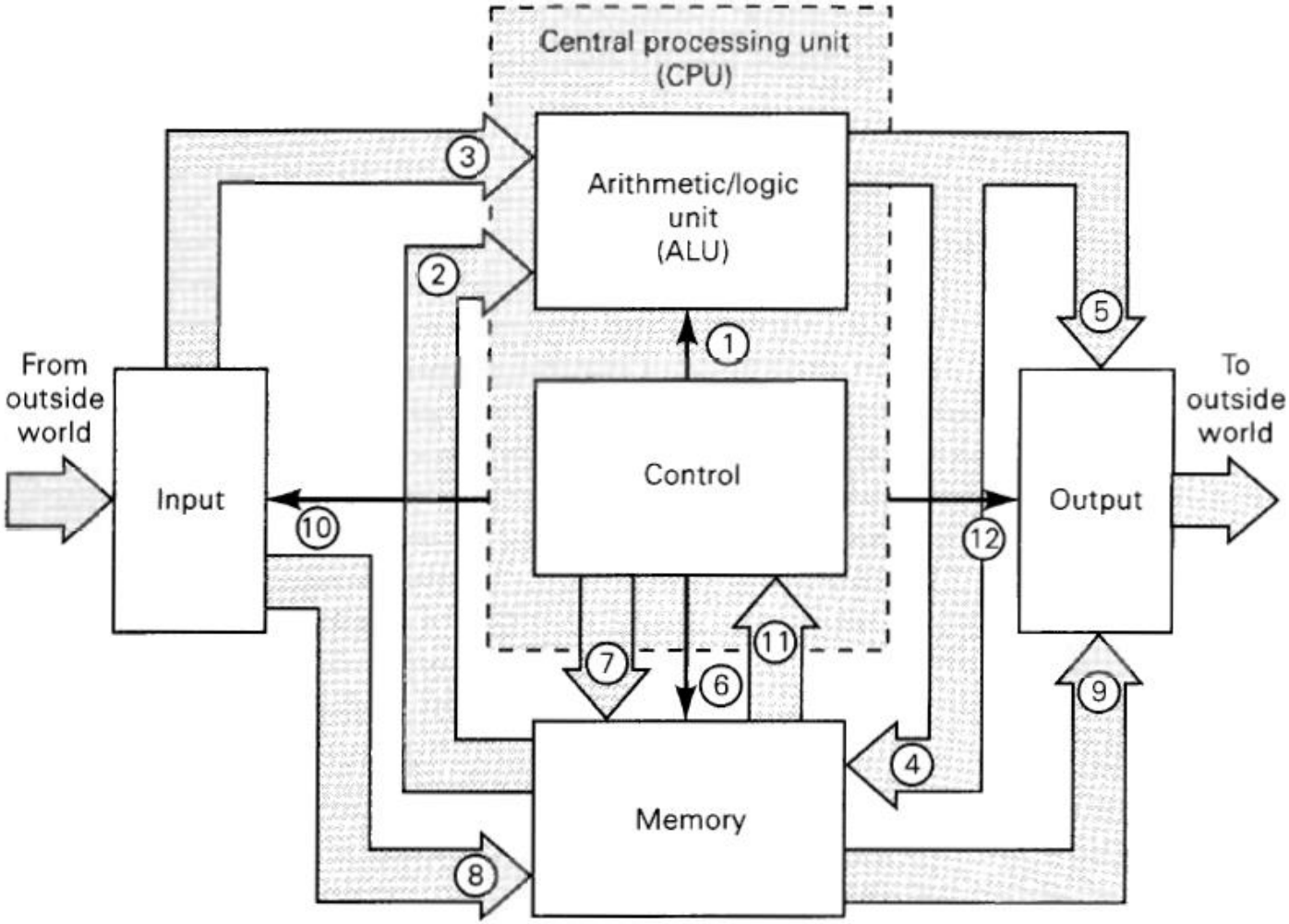
Organización

- Señales de control.
- Interfaces entre la computadora y los periféricos.
- La tecnología de memoria usada.

Organización básica de un sistema de cómputo

- Unidad Aritmético-lógica
- Unidad de memoria
- Unidad de control
- Unidad de entrada
- Unidad de salida

Organización básica de un sistema de cómputo



Basic computer organization.

Unidad aritmético-lógica ALU

Es el área de la computadora en **donde se llevan a cabo las operaciones aritméticas lógicas de los datos**

Unidad de memoria

Almacena grupos de dígitos binarios (palabras) que pueden representar instrucciones (programa) que realizará la computadora y los datos que serán procesados a través del programa.

Unidad de entrada

Consiste en todos los dispositivos que se usan para recibir información y datos que son externos a la computadora y ponerlos en la unidad de memoria.

Unidad de salida

- Consiste en todos los dispositivos que se usan para transferir datos e información desde la computadora hacia el mundo exterior.

Periféricos

- Los dispositivos que constituyen las unidades de entrada y salida

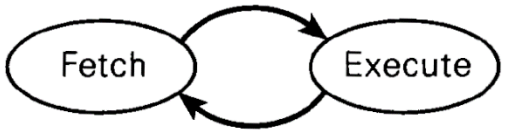
Interfaz

- Específicamente: La transmisión de información digital entre una computadora y sus periféricos.

Unidad de control

- Dirige la operación de todas las otras unidades proporcionando señales de sincronización y control.
- Busca una instrucción en la memoria, la decodifica y envía las señales adecuadas al resto de las unidades para ejecutar la operación específica.

FIGURE A-3 A computer continually fetches and executes instructions.



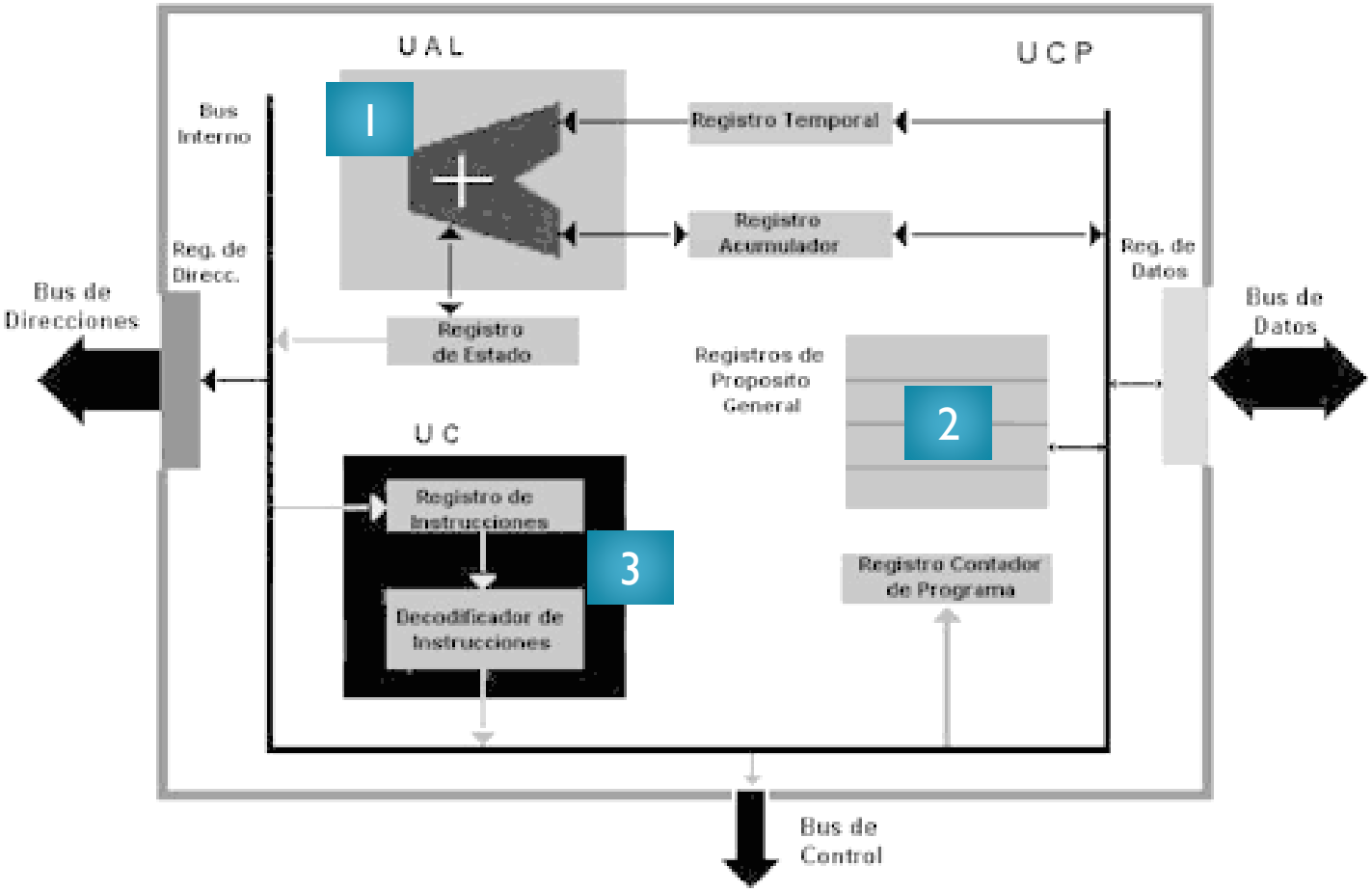
Unidad Central de Procesamiento

Controla el funcionamiento de la computadora decodificando las instrucciones contenidas en un programa de cómputo.

La CPU por lo general se implementa en un solo chip: el microprocesador.

1. Unidad aritmético-lógica
2. Unidad de control
3. Registros de memoria (no necesariamente la memoria externa)

1. Unidad aritmético-lógica
2. Registros de memoria
3. Unidad de control



Microcontrolador o microcomputadora

- El µC es un computador completo, aunque de limitadas prestaciones, que esta contenido en el chip de un circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria.

Cuestionario

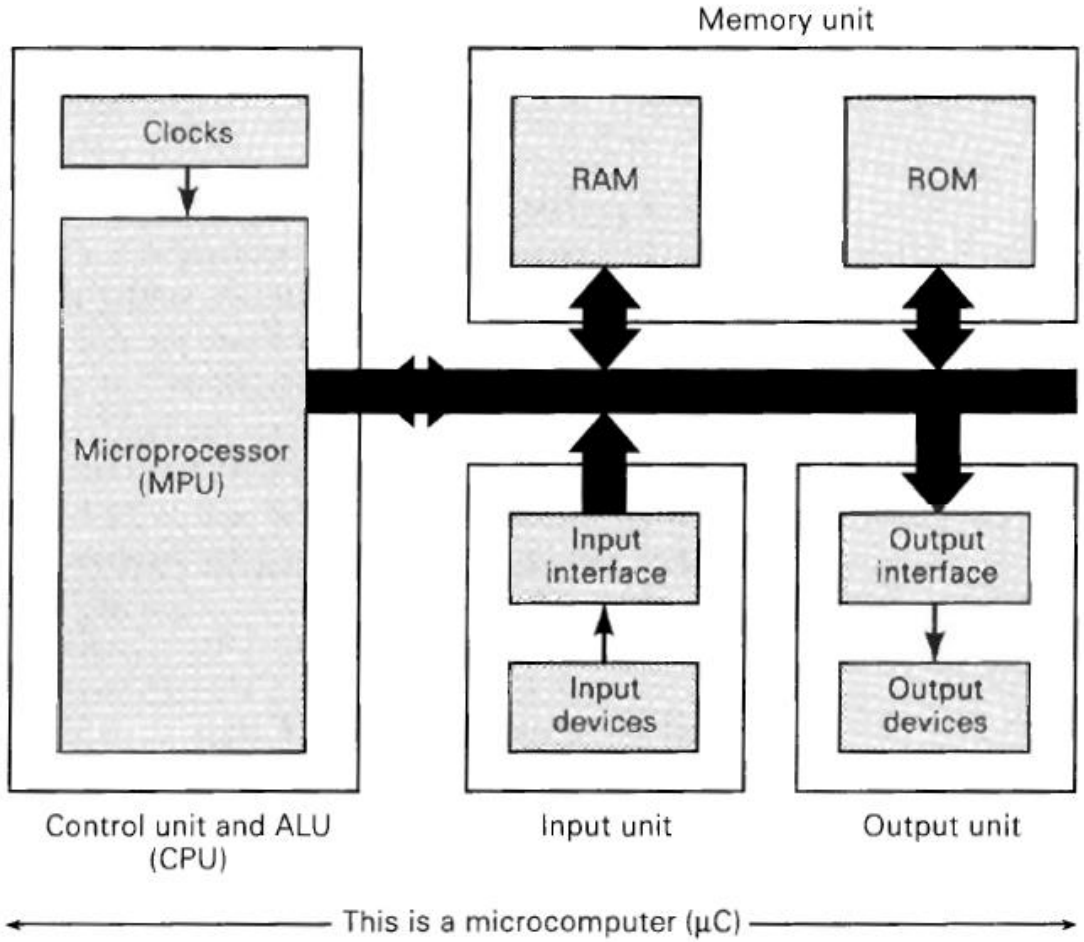
- ¿Qué es una computadora?
- Nombre las cinco unidades básicas de una computadora y describa sus funciones.
- ¿Qué es una CPU?
- ¿Cuál es el significado de Interfaz en un sistema de cómputo?
- ¿Qué operaciones básicas ocurren repetidamente en una computadora?
- ¿Qué es un microcontrolador?

Elementos básicos de un Microcontrolador o microcomputadora

- Un µC contiene varios elementos, Memoria RAM, ROM. periféricos. El más importante es el µP.

- El µP es un solo CI que contiene toda la circuitería de las unidades de control y ALU (CPU).
  - µP = UC + ALU

Elementos básicos de un Microcontrolador



Basic elements of a microcomputer.

Unidad de memoria

RAM:

- Consta de uno o más chips LSI configurados para proporcionar la capacidad de memoria diseñada.
- Se usa para almacenar programas y datos, los cuales cambiarán con frecuencia durante el curso de la operación.
- Se usa como almacenamiento para resultados intermedios y finales de operaciones realizadas durante la ejecución de un programa.

Unidad de memoria

ROM:

- Consta de uno o más chips ROM para almacenar instrucciones y datos que no cambian y que no se deben perder cuando se interrumpa la energía.
- Ejemplo 1, almacena el programa de arranque que el µC ejecuta al momento de encenderse.
- Ejemplo 2, almacena una tabla ASCII necesaria para dar salida a información a una VDT *(video/visual Data terminal) o a una impresora.*

Secciones de entrada y salida

- Contienen los circuitos de interfaz necesarios para permitir que los periféricos se comuniquen apropiadamente con el resto de la computadora.
- En algunos casos estos circuitos de interfaz son chips *LSI* diseñados por el fabricante de la *MPU* para conectar la *MPU* a una variedad de dispositivos de E/S.
- En otros casos los circuitos de interfaz pueden ser tan simples como un registro búfer.

Scale of integration

Scale Of Integration

The number of components fitted into a standard size IC represents its integration scale, in other words it's a density of components. It is classified as follows:

1. **SSI – Small Scale Integration**

   It have less than 100 components (about 10 gates).

2. **MSI – Medium Scale Integration**

   It contains less than 500 components or have more than 10 but less than 100 gates.

3. **LSI – Large Scale Integration**

   Here number of components is between 500 and 300000 or have more than 100 gates.

4. **VLSI – Very Large Scale Integration**

   It contains more than 300000 components per chip

5. **VVLSI - Very Very Large Scale Integration**

   It contains more than 1500000 components per chip.

Microprocesador

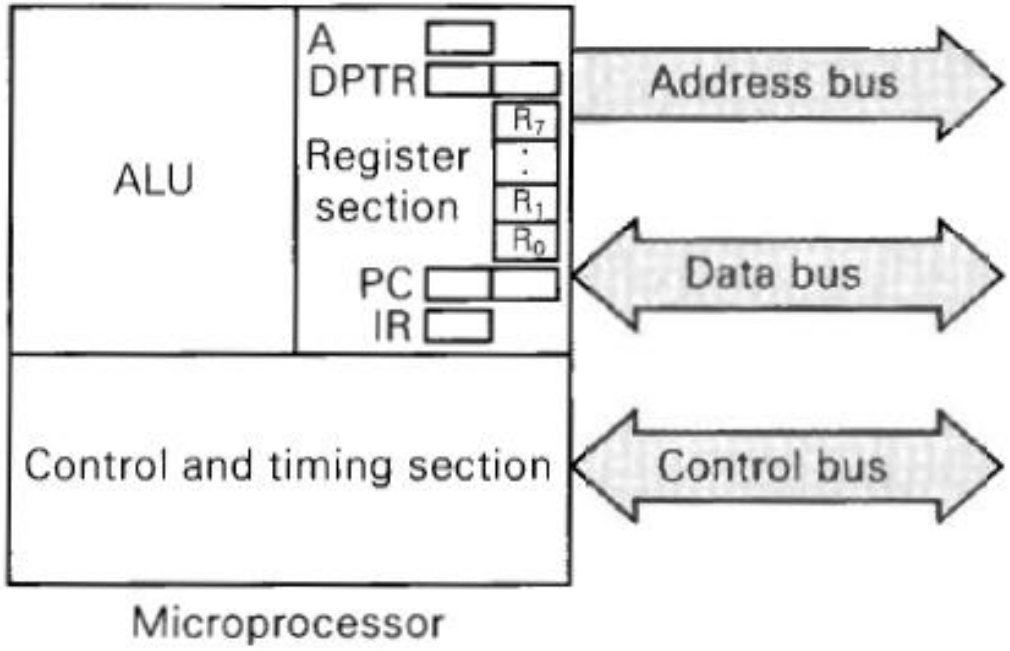
- Es el corazón del µC.
- Proporciona señales de sincronización y control.
- Busca instrucciones y datos en la memoria.
- Transfiere datos hacia y desde la memoria y a los dispositivos de E/S.
- Decodifica instrucciones.
- Realiza operaciones aritméticas y lógicas.
- Responde a señales de control como RESET e INTERRUPT.

Microprocesador

- Contiene toda la circuitería lógica para llevar a cabo las funciones anteriores.
- Su lógica interna por lo general no es accesible de forma externa.
- *Se puede controlar lo que sucede en su interior mediante el programa de instrucciones que se puso en su memoria.*
- ***Cuando se requiere cambiar su operación simplemente se cambian los programas almacenados en la RAM (software) o ROM (firmware)***

Principales áreas funcionales de un Microprocesador

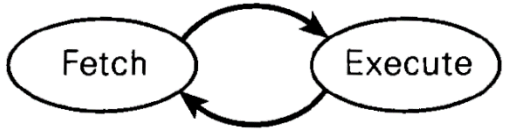
- SecciónALU
- Sección de Control y sincronización
- Sección de registro



Sección de control y sincronización

- Busca y decodifica/interpreta códigos de instrucciones de la memoria de programa y luego genera las señales de control necesarias que requieren otras secciones de la MPU para llevar a cabo la ejecución de las instrucciones.
- Genera señales de sincronización y control como R/W o reloj que necesitan los dispositivos RAM, ROM y E/S.

FIGURE A-3 A computer continually fetches and executes instructions.



Sección de registro

- Contiene varios registros dentro de la MPU que realizan una función especial.
- **El más importante es el Contador de programa: PC**. Este mantiene un registro de las direcciones de los códigos de instrucciones a medida que son buscados en la memoria.

Sección de registro

Otros registros importantes:

- IR: Registro de instrucciones. Almacena códigos de instrucción a medida que se decodifican.
- A: Retención de datos que son operados por la ALU. Ejemplo, el registro «acumulador».
- DPTR: Apuntador de datos. Almacenar direcciones de datos que son buscados en la memoria.
- R0..R7: Almacenamiento general y conteo.

Cuestionario

- Nombre las tres secciones más importantes de una MPU
- Describe cada elemento de la sección de registros.
- ¿De qué se encarga la sección de control y sincronización?

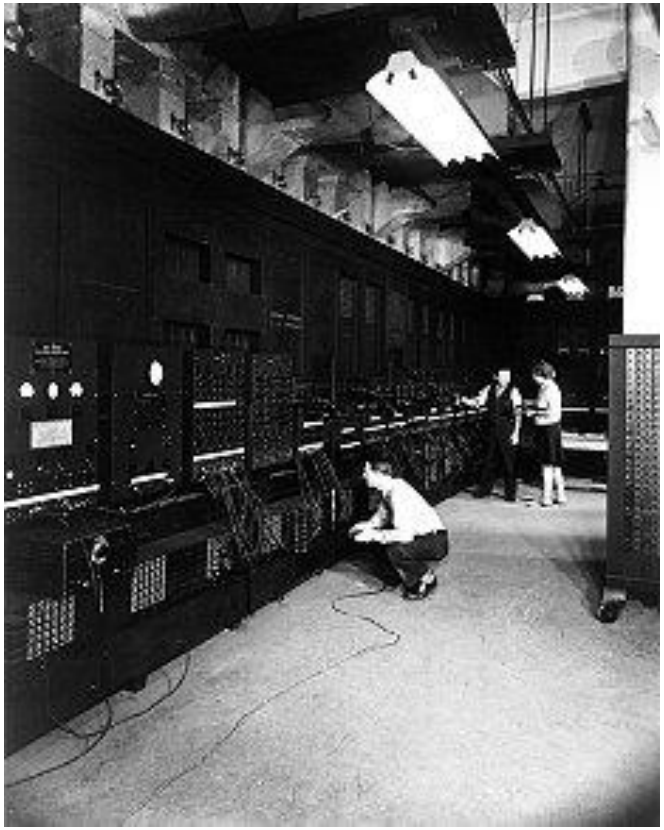
Arquitectura de Von Neumann

- En la década de los 40's surgían las computadoras con tecnología de tubos al vacío procesando hasta 5000 sumas / segundo.

- La primera computadora de propósito general fue la "ENIAC" (Electronic Numerical Integrator And Computer)
Basada en la máquina mecánica de Turing pero con componentes digitales.

- Tediosas de modificar.

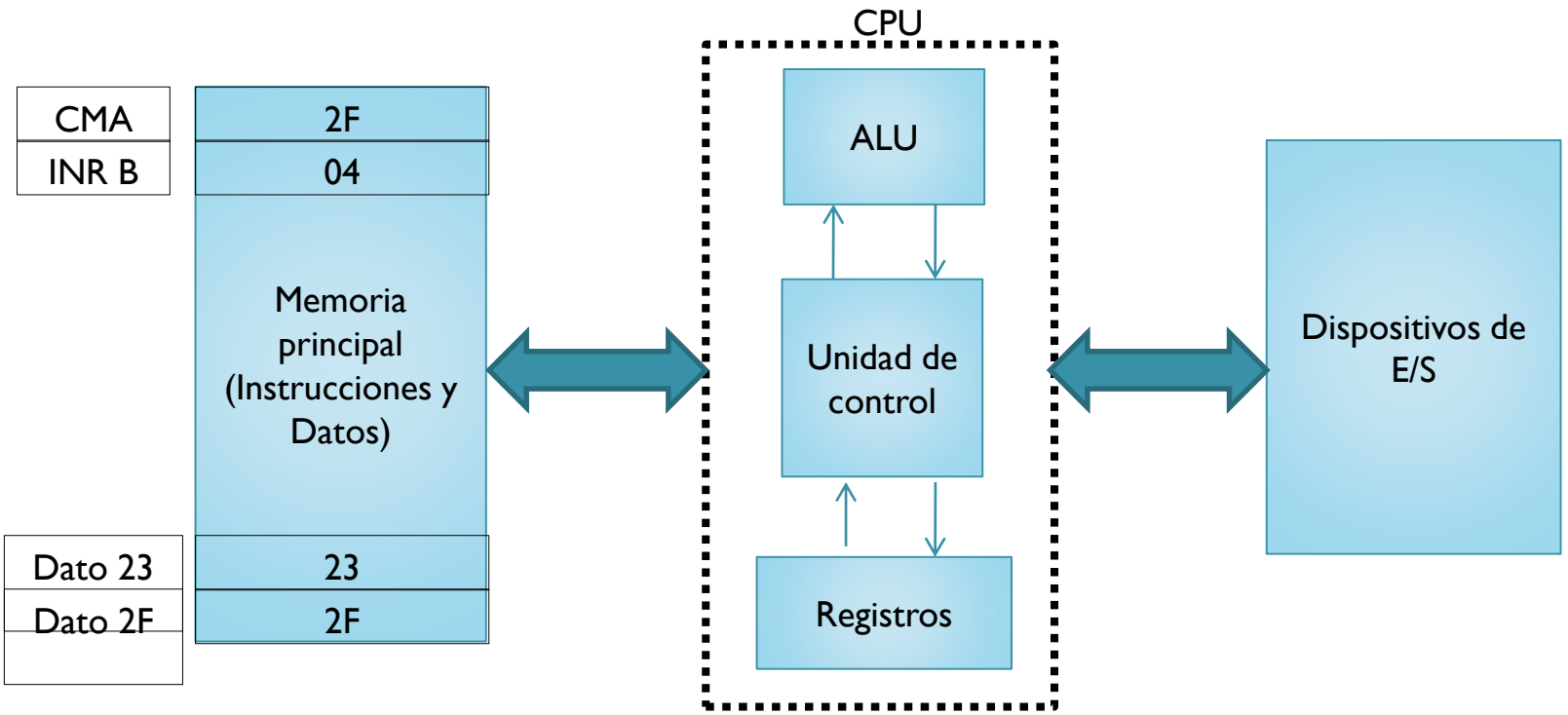
- Neumann propuso el proyecto de una computadora de uso general. (IAS- machine) del *Institute for Advanced Study*



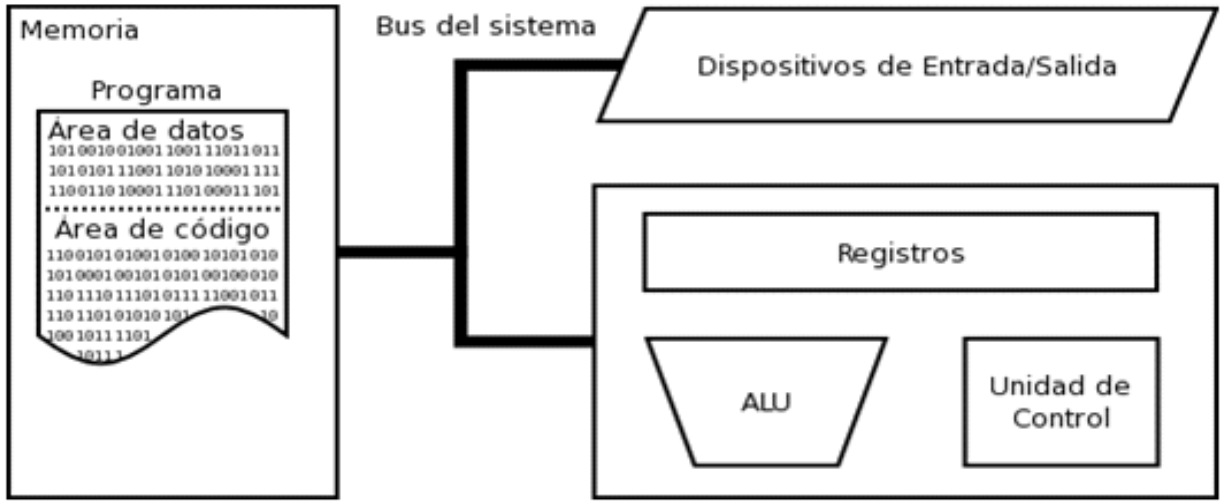
ENIAC being set up

Designing the correct configuration for each new problem, and then connecting the wires and setting the switches, took many days.

Arquitectura de Von Neumann



Arquitectura de Von Neumann



En los últimos años, **la velocidad de los microprocesadores** ha ido creciendo mucho **en comparación a la de las memorias** por lo que una diferencia significativa conlleva a un **problema de limitación de memoria.**

Una solución es la **memoria caché.**

Memoria Caché

Definition of: ***cache.*** *Reserved areas of memory in every computer that are used to speed up instruction execution, data retrieval and data updating. Pronounced "cash," they serve as staging areas, and their contents are constantly changing.*

There are two kinds: **memory caches and disk caches.**

Memoria Caché

Memory Caches

A memory cache, also called a "CPU cache," is a memory bank that bridges main memory and the processor. Comprising faster static RAM (SRAM) chips than the dynamic RAM (DRAM) used for main memory.

Memoria Caché

Disk Caches

A disk cache is a section of main memory or memory in the disk controller that bridges the disk and CPU. When the disk is read, a larger block of data is copied into the cache than is immediately required. If subsequent reads find the data already stored in the cache, there is no need to retrieve it from the disk, which is slower to access.

Arquitectura Harvard

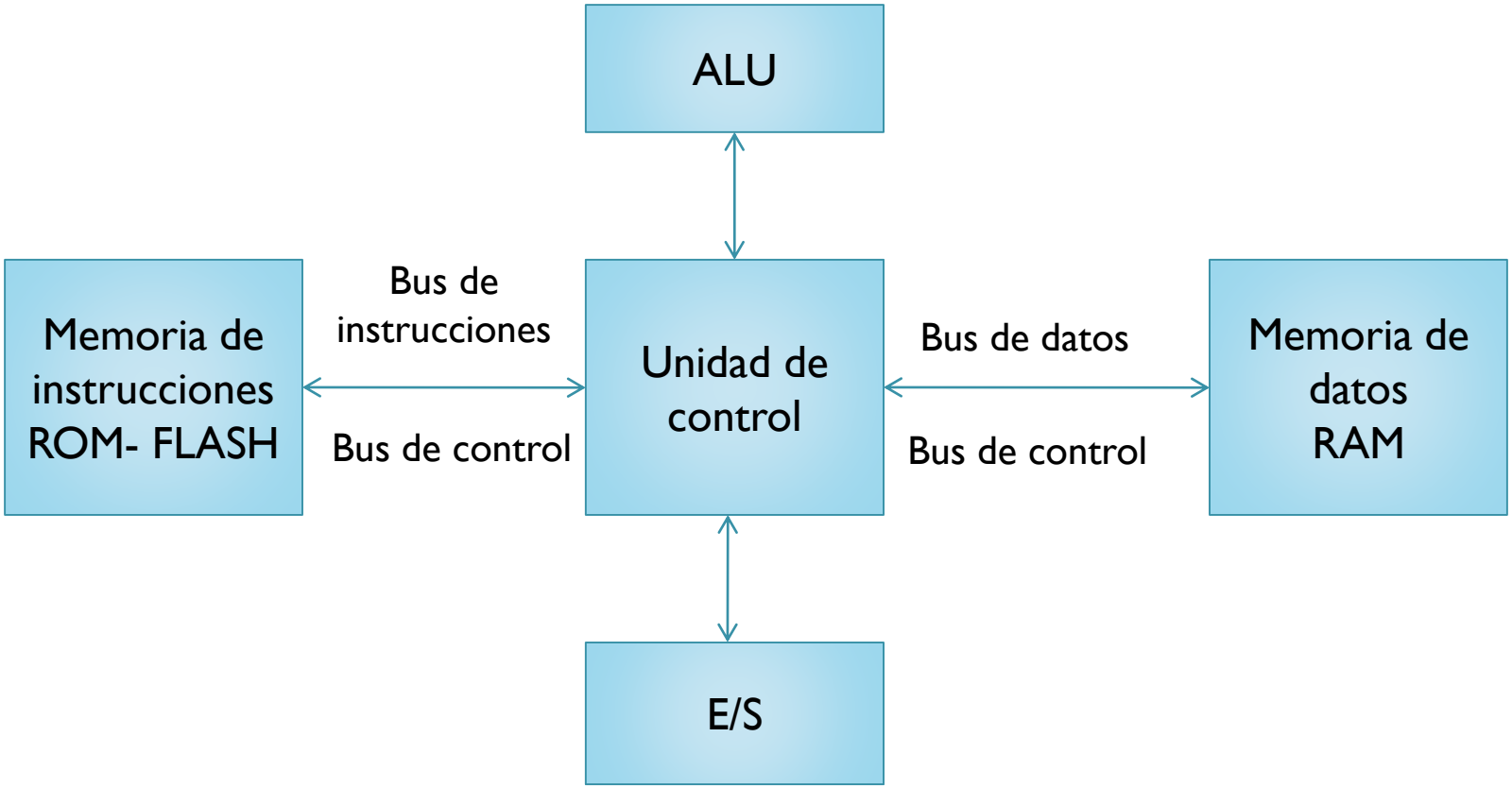
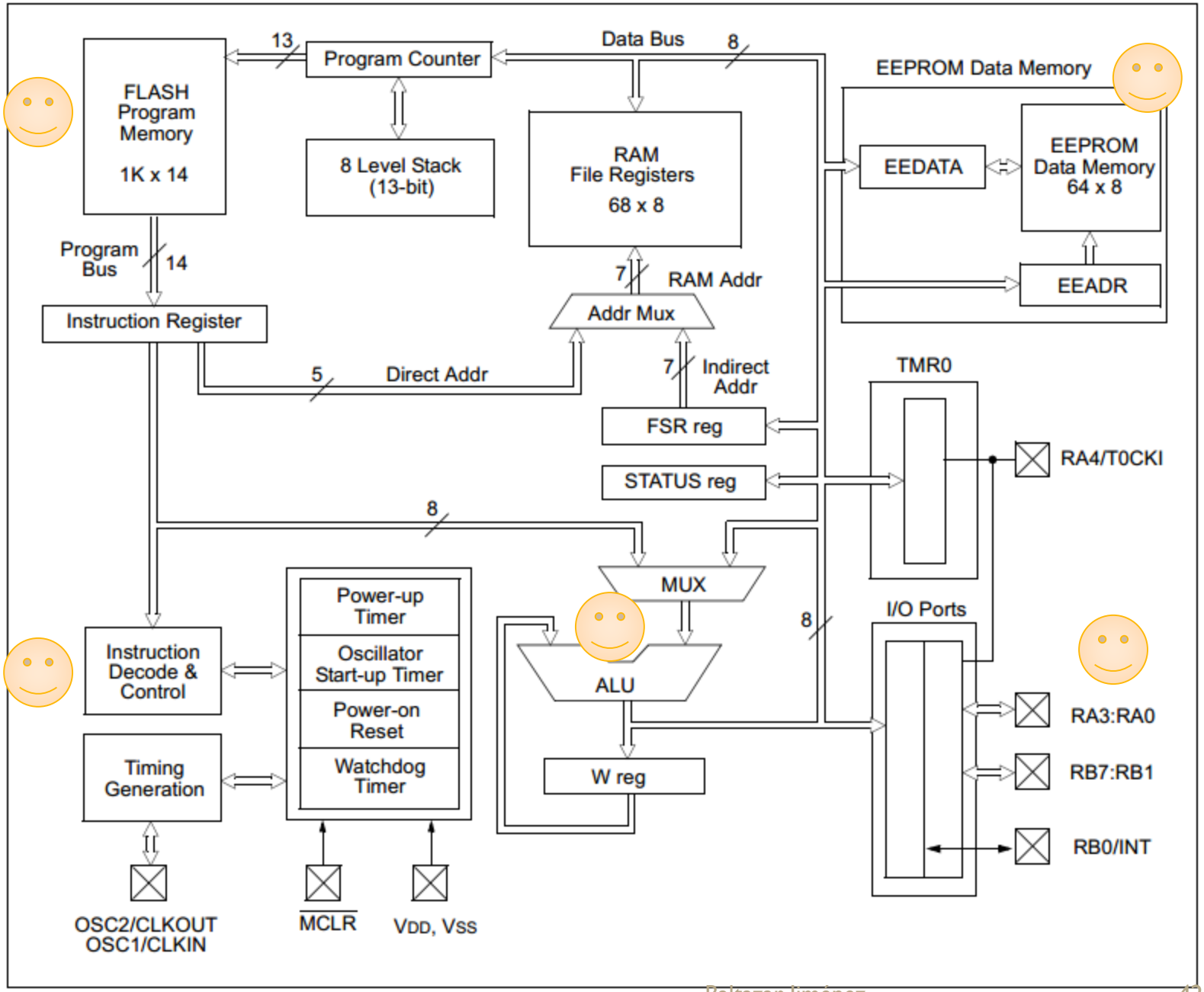


FIGURE 1-1: PIC16F84A BLOCK DIAGRAM



Arquitecturas que dependen del tipo de instrucciones

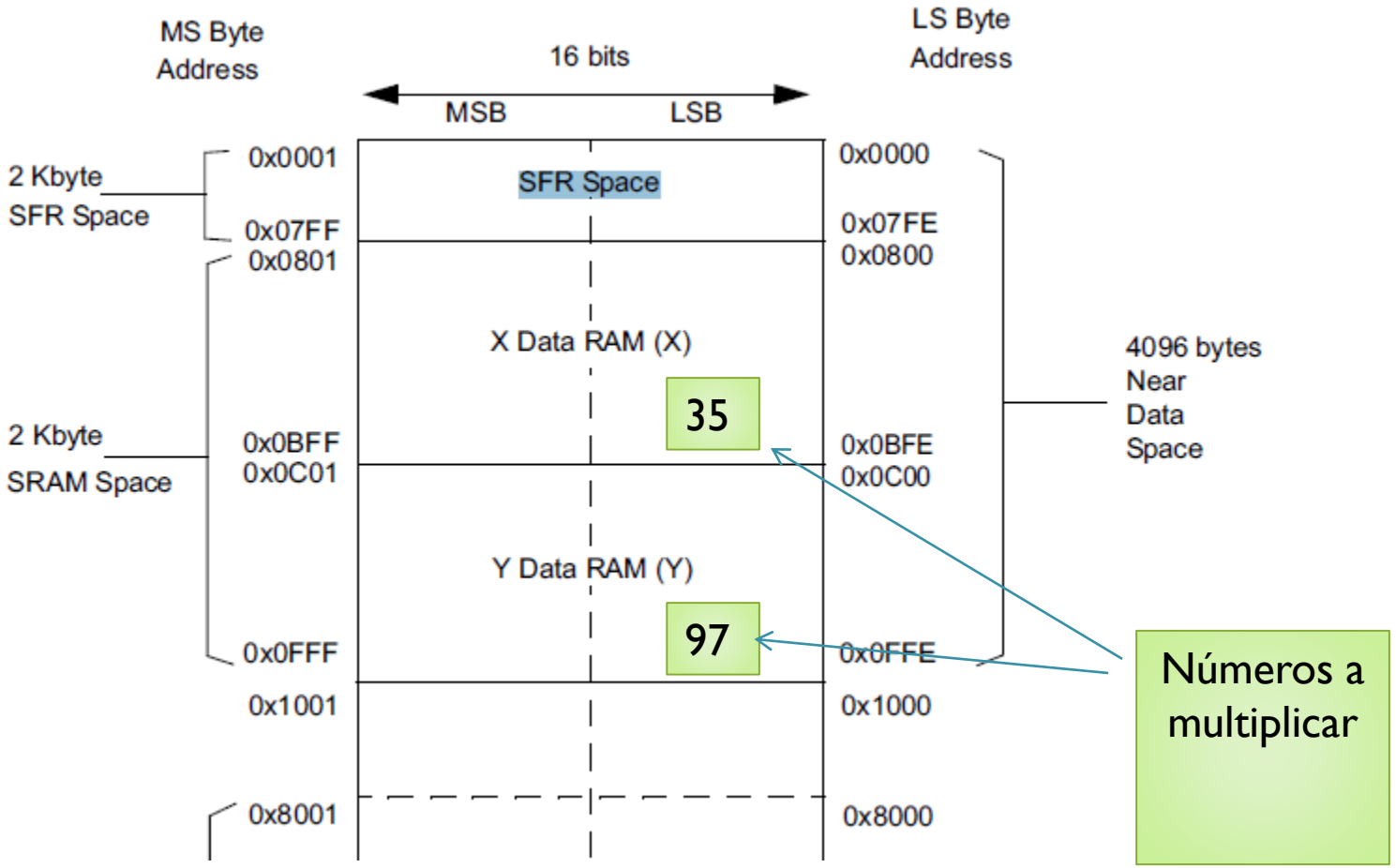
CISC

- El objetivo principal de la arquitectura *CISC* es completar una tarea en el menor número de líneas de código ensamblador posibles.

- Para ello es necesaria la construcción de un microprocesador capaz de comprender y ejecutar una serie de operaciones complejas.

  - Ejemplo: Implementar la función "multiplicación" en un microcontrolador con arquitectura *CISC*.

Mapa de memoria de un microcontrolador (DSPIC 30F)



Ejemplo CISC

La instrucción con una arquitectura *CISC* para multiplicar los números anteriores y dejar el resultado en el número de la parte superior sería:

- Mult ([0x0BFE],[0X0FFE]) una sola instrucción

Que es similar a un lenguaje de programación de alto nivel: **A = A * B**

Por lo tanto una ventaja de *CISC* es el poco trabajo para traducir de un lenguaje de alto nivel a lenguaje ensamblador

Arquitectura RISC

- Usan instrucciones sencillas que se puedan ejecutar rápidamente.
- Suelen ser arquitecturas basadas en registros de propósito general.

- Ejemplo: Implementar la función "multiplicación" en un microcontrolador con arquitectura *RISC*.

Ejemplo RISC

Con una arquitectura tipo RISC, las instrucciones para hacer la misma multiplicación serían:

```
MOV    [0x0BFE],          W0
MOV    [0X0FFE],          W1
MUL    W0,                W1,          W3:W2
MOV    [W2],              W0
```

- Instrucciones sencillas que no requieren tanto hardware para interpretar una instrucción compleja
- No hay operadores que combinen la carga y el almacenamiento al mismo tiempo

A-3 SECRET AGENT 89

Secret Agent 89 is trying to find out the number of the airport landing strip where a known terrorist will be landing. His contact tells him that this information is located in a series of post office boxes. To ensure that no one else gets the information, it is spread through 10 different boxes. His contact gives him 10 keys along with the following instructions:

1. The information in each box is written in code.
2. Open box 1 first and execute the instruction located there.
3. Continue through the rest of the boxes in sequence unless instructed to do otherwise.
4. One of the boxes contains information that will misdirect anyone but Agent 89.

Agent 89 takes the 10 keys and proceeds to the post office, code book in hand. Figure A-1 shows the contents of the 10 post office boxes after having been decoded. Assume that you are Agent 89; begin at box 1 and go through the sequence of operations to find the number of the landing strip. Of course, it should not be as much work for you as it was for Agent 89 because you don't have to decode the messages. The answer is given in the next paragraph.

① Add the number stored in box ⑨ to your secret agent code number.	② Divide the previous result by the number stored in box ⑩.
③ Subtract the number stored in box ⑧.	④ If the previous result is not equal to 30, go to box ⑦. Otherwise continue to next box.
⑤ Subtract 13 from the previous result.	⑥ Return to headquarters for more instructions.
⑦ The landing will take place on strip #3.	⑧ 20
⑨ 11	⑩ 2

FIGURE A-1 Ten post office boxes with coded message for Agent 89.

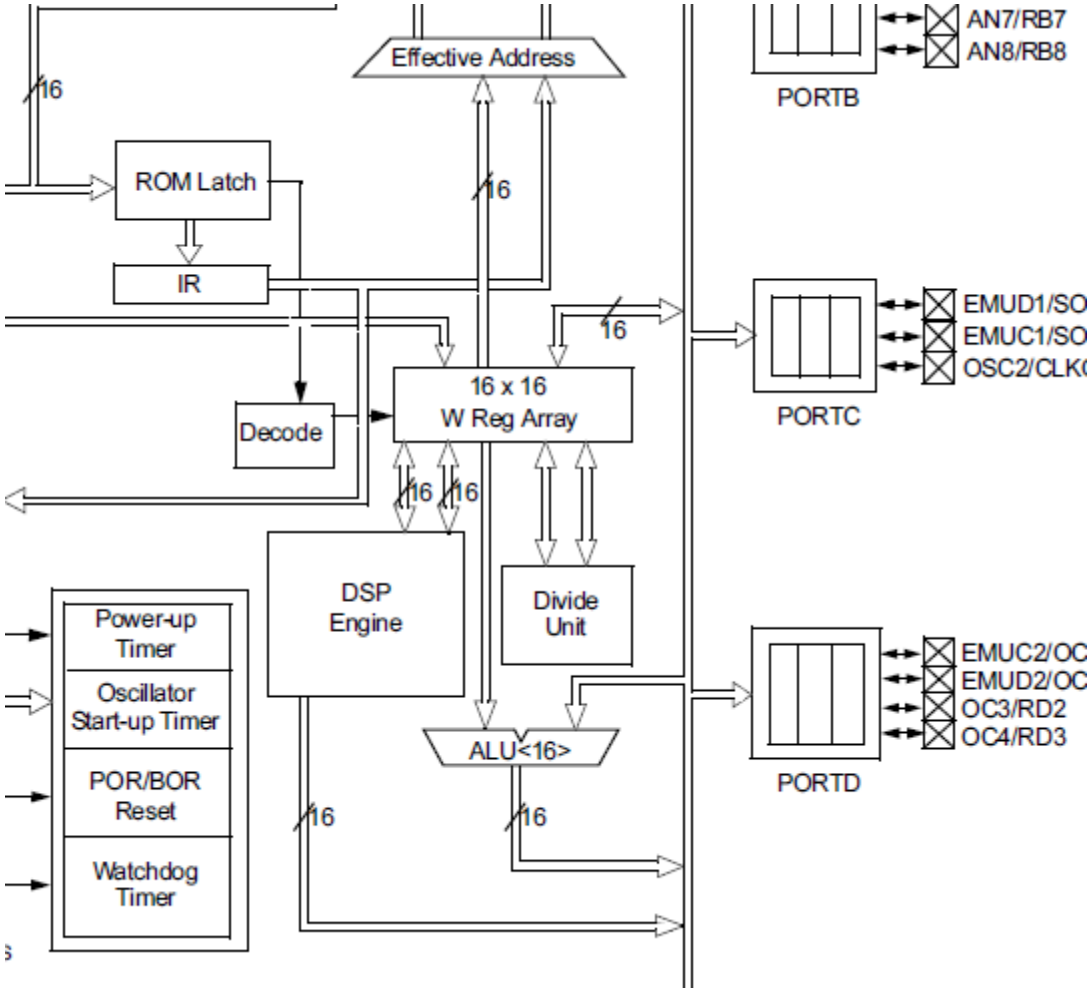
Palabras de computadora

- Un bit es la unidad más pequeña de información en una computadora.
- La unidad principal de información en una computadora es el grupo de bits al que se le denomina **palabra.**
- El número de bits que componen una palabra se llama **tamaño de palabra de computadora.**

Palabras de computadora

- Ejemplo:
  - Una computadora de 16 bits es aquella en la cual las instrucciones y datos se almacenan en memoria como unidades de 16 bits. El tamaño de palabra también indica el tamaño del bus de datos que transporta los datos entre la CPU y la memoria y entre la CPU y los dispositivos de E/S

Dspic30F4011: uC de 16 bits



Tamaño de palabra

- Los tamaños mayores de palabra significan más líneas que conforman el bus de datos y por lo tanto más interconexiones entre la CPU y la memoria y los dispositivos de E/S

Tipos de palabra

- Una palabra almacenada en la memoria de una computadora puede contener **Instrucciones o datos.**

- **Los datos** pueden ser información numérica o caracteres que procesará un programa que la CPU está ejecutando.

- **Las palabras de instrucción** contienen la información necesaria para que la computadora ejecute sus diversas operaciones. Su formato y códigos pueden variar entre computadoras

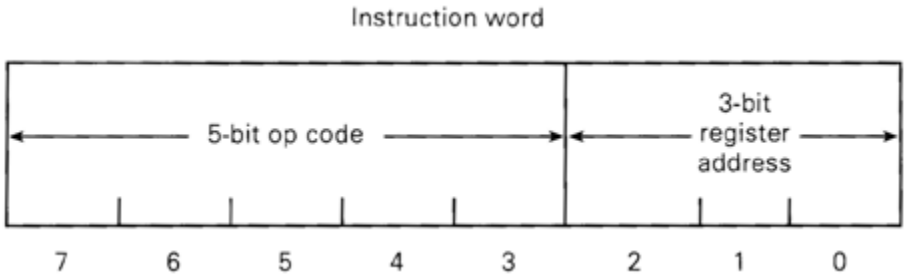
Palabras de instrucción

- Conllevan unidades básicas de información:
  - La operación que se llevará a cabo.
  - La dirección del operando (el dato).

Palabras de instrucción

Figure A-6 shows an example of a *single-address instruction word* for the 8051. The eight bits of the instruction word are divided into two parts. The first part of the word (bits 7 through 3) contains the five-bit **operation code** (**op code,** for short). The op code represents the operation that the computer is being instructed to perform, such as addition, subtraction, or moving data. The second part (bits 2 through 0) is the **operand address,** which represents the location in memory where the operand is stored.

FIGURE A-6 Typical single-address instruction word.



Instruction words

The important point to understand from this form of an instruction is that the first five bits tell the micro (microprocessor) what to do. The last three bits tell the micro where to get the data. As an example, the 8051 instruction to "MOVE to the accumulator the data in register 3" is encoded as follows:

11101	011
op code	register address

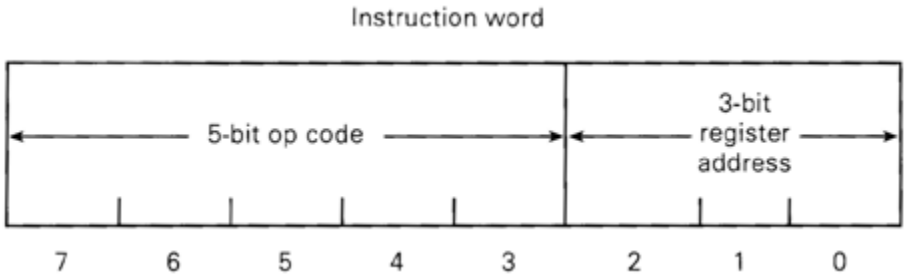


FIGURE A-6 Typical single-address instruction word.

Instruction words

The important point to understand from this form of an instruction is that the first five bits tell the micro (microprocessor) what to do. The last three bits tell the micro where to get the data. As an example, the 8051 instruction to "MOVE to the accumulator the data in register 3" is encoded as follows:

11101	011
op code	register address

The eight-bit data word contained in R3 is moved (actually copied) into the accumulator. The original contents of the accumulator are lost.

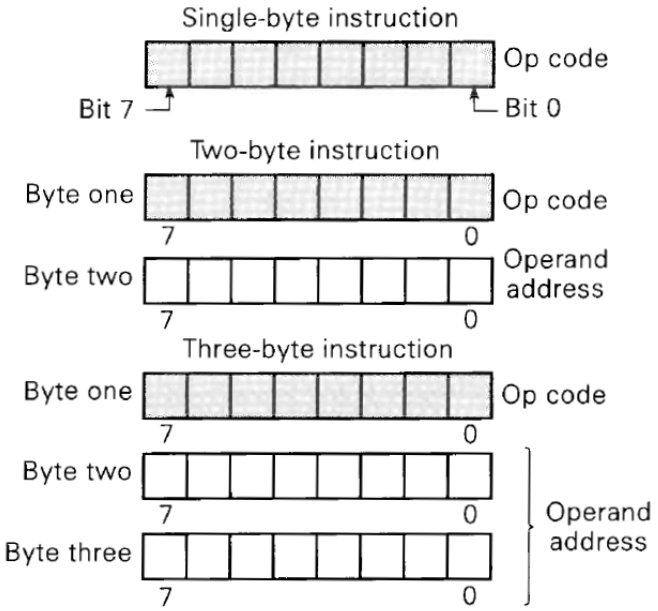
Multibyte Instructions

We have seen an instruction-word format that contains op code and operand address information in a *single* word. In other words, a complete instruction such as that in Figure A-6 is stored in a *single* memory location. If all instructions were to be encoded this way, we would need a larger word size to contain all of the possible instructions and the operand addresses. Computers with larger word sizes make use of this format whenever possible. The 8051 is limited to an eight-bit word size. In order to provide more instructions and more flexibility in accessing data, most computers with small word sizes use more than one word to describe each instruction. Typically there are three basic instruction formats: single-byte, two-byte, and three-byte. These are illustrated in Figure A-7.

Multibyte Instructions

In a two-byte instruction, the first byte always contains the op code. The purpose of the second byte (operand) is to specify the data value that is to be used in

FIGURE A-7 Instruction formats used in eight-bit microcomputers.



Multibyte Instructions

the operation. There are various methods of specifying this data. These methods are called **addressing modes.** We will discuss only two popular addressing modes in this text. The first is similar to the method described previously, in which the operand represents an address that directs the computer to the place where the data value is stored. This is called **direct addressing.** The other addressing mode includes the actual data value as part of the instruction. In this mode the operand is the data value itself instead of the address of the data value. Since the data value immediately follows the op code in the program memory, this is called the **immediate addressing** mode.

Multibyte Instructions

The three-byte instruction is necessary when the operand must be a 16-bit number. For these multibyte instructions, the two or three bytes making up the complete instruction must be stored in successive memory locations. This is illustrated in Table A-1 for a 3-byte instruction. The left-hand column lists the address locations in memory where each byte (word) is stored. These addresses are given in hexadecimal. The second column gives the binary word as it is actually stored in memory; the third column is the hex equivalent of this word. Examine this table carefully before reading further, and try to figure out what it represents.

For example, consider the instruction that causes the computer to jump back to the first instruction and start the program over again. This instruction is shown in Table A-1 as it would appear in memory. The first byte is the op code (02), which tells the computer to jump. This code also informs the control unit that this is a three-byte instruction and that in order to know where to jump, it must fetch the next two bytes. Together these bytes make up the 16-bit address of the next instruction to be fetched. Recall that the program counter always holds the address of the next instruction. When the program counter has advanced to 2377, the computer will fetch the op code 02. It then fetches the next two bytes (20 and 50) and loads them back into the program counter. The new address in the program counter means that the next instruction will be fetched from 2050 instead of 237A.

TABLE A-1 Program instructions stored in memory.

Memory Address	Binary	Hex	Description
2050	XXXXXXXX	XX	First instruction in a program
2051			
2052			
.			
.			
.			
2377	00000010	02	Op code for Jump (LJMP)
2378	00100000	20	High byte of where to jump
2379	01010000	50	Low byte of where to jump
237A			
.			
.			
.			

Assembly language

TABLE A-2 Some 8051 instructions.

Mnemonic	Op Code		Description of Operation
	Binary	Hex	
LJMP address	0000 0010	02	Long Jump. Jump from the current memory location in the program to the address specified in the two-byte operand.
MOV A, direct	1110 0101	E5	Move from an address to Acc. The data value contained in the direct address is moved to Acc.
MOV direct, A	1111 0101	F5	Move from Acc to an address. The data value in Acc is moved to the direct memory address specified in the instruction.
DEC A	0001 0100	14	Decrement the Accumulator. Subtract 1 from the value in Acc.
JNZ address	0111 0000	70	Jump if Acc is Not Zero. If Acc $\neq$ 0, the next instruction is to be fetched from the operand address. Otherwise, the next instruction in sequence is fetched.
JZ address	0110 0000	60	Jump if Acc is Zero. If Acc = 0, the next instruction is to be fetched from the operand address. Otherwise, the next instruction in sequence is fetched.
LCALL address	0001 0010	12	Long Call to subroutine. Causes a short program called a *subroutine* to execute. When it is done, the program returns to the instruction following LCALL.
RET	0010 0010	22	Return from subroutine. Sends the computer back to wherever it was called from.

What does it does?

TABLE A-3 Sample machine-language program.

Memory Address (Hex)	Memory Contents (Hex)	Assembly Language	Description
0000	02	LJMP 0100H	;JUMP to start of program
0001	01		
0002	00		
0100	60	JZ 010AH	;Should we cook the food?
0101	08		
0102	F5	MOV P1, A	;Display cook time on port 1
0103	90		
0104	12	LCALL 1__SEC__DELAY	;Waste one sec
0105	28		
0106	55		
0107	14	DEC A	;Subtract one sec from time
0108	70	JNZ 0102H	;Is the food done?
0109	F8		
010A	********This is where the rest of the program continues.********		

Cuestionario (pág 806, Tocci)

- ¿Qué es una palabra de computadora?
- ¿Cuáles son las diferencias entre una instrucción y un dato?
- Describe el funcionamiento de una *"single addres instruction"*. Menciona alguna limitante
- *¿Qué significa "código de operación"?*
- *¿Cómo funciona una "Multibyte Instruction"?*
- *¿Qué es un modo de direccionamiento?*
- *¿En qué consiste el direccionamiento directo?*
- *¿En qué consiste el direccionamiento inmediato?*

Bibliografía

[1]Andrew S. Tanenbaum, *Organización de computadoras, un enfoque estructurado*, Cuarta ed. México, 2000.

[2]Microchip. (2013, Diciembre) Pic Microcontrollers. [Online].

http://www.microchip.com/pagehandler/en-us/products/picmicrocontrollers

[3]Ronald J. Tocci and Neal S. Widmer. Digital Systems. 8th Ed.

[4] Alfonso Gutierrez Aldana. Microcontroladores PIC16 fundamentos y aplicaciones. 1ra edición.