

Universidad Tecnológica Nacional
Facultad Regional Tucumán
Ingeniería en Sistemas de Información
ARQUITECTURA DE COMPUTADORES

TRABAJO PRÁCTICO N° 4

Arquitectura. Modelos Von Neumann e IAS.

Capítulos 2 y 3

Niveles de máquina en la jerarquía de un sistema

Superior

Nivel de usuario: programas de aplicación.

Lenguajes de alto nivel.

Lenguajes de bajo nivel (assembler y código máquina)

Control microprogramado / cableado

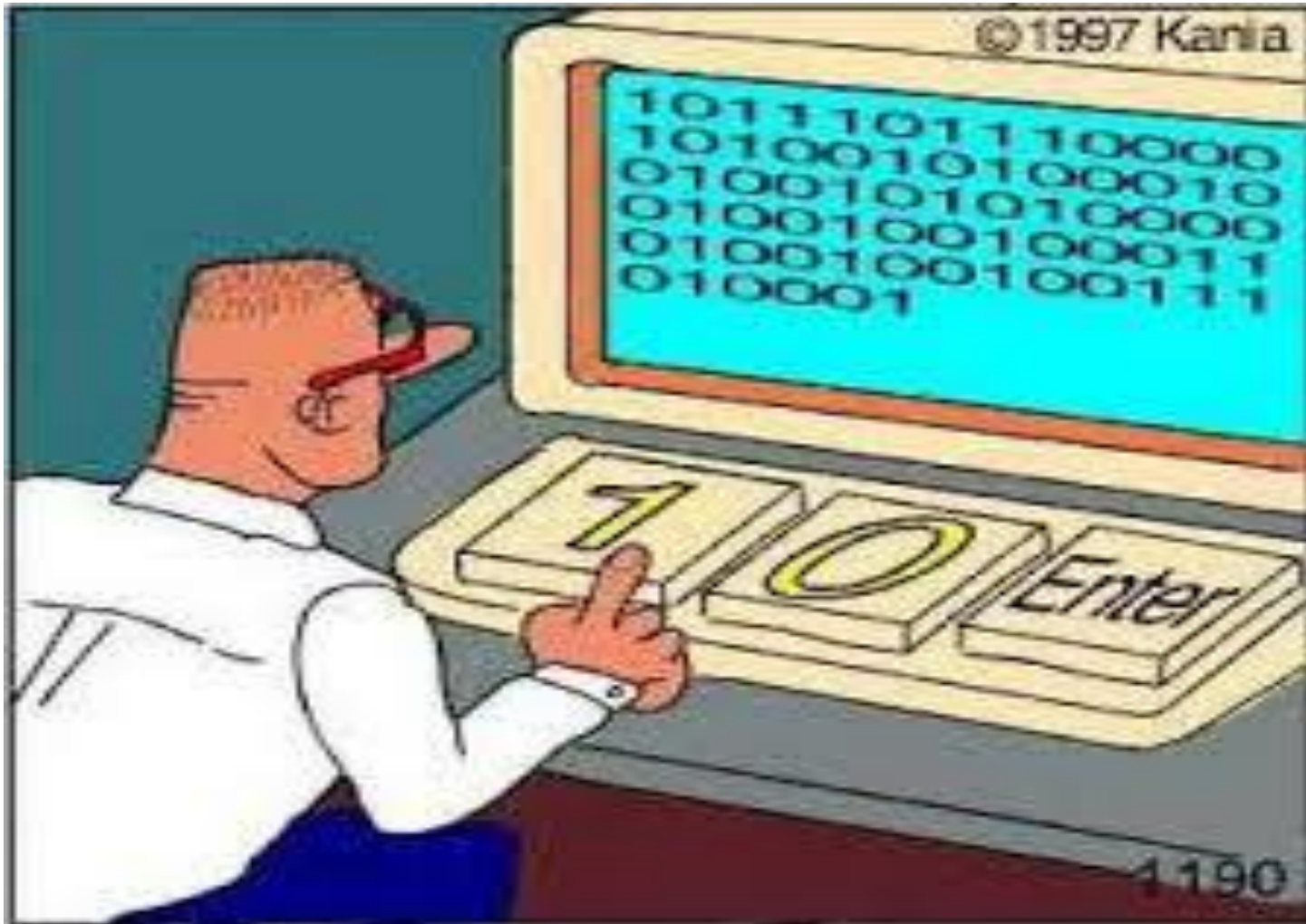
Unidades funcionales (ALU, memoria, etc.)

Circuitos lógicos

Transistores y cables

Inferior

La computadora entiende solo 0 y 1, pero ¿cómo es el proceso?





escribe
un

Programa en Lenguaje de Alto Nivel

C++

Momento de
Compilado

Programa en Lenguaje Simbólico (assembler)

Momento de
Ensamblado

Programa en Lenguaje de Máquina (código máquina)

Momento de Enlace

Programa Ejecutable

Almacenamiento

Disco Duro



Momento de la
Carga

Memoria



Momento de la
Ejecución



Salidas del
Programa



Los resultados
obtenidos en la
CPU vuelven a
Memoria para su
almacenamiento

**DEPENDENCIA
DE LA CPU**

0 y 1 es lo
único que
entiende la
computadora

Permite
relacionarse
con otros
programas
en código
máquina.

El Sistema Operativo
carga el programa, en
lenguaje máquina, desde
el disco a memoria

Desde memoria se
carga la instrucción a
la CPU, una por vez,
junto con los datos
necesarios para
ejecutar la instrucción

Algunas Definiciones

- Un programa compilador traduce un programa en lenguaje de alto nivel, que es independiente de la arquitectura, a uno en lenguaje ensamblador, el cual es dependiente de la arquitectura.
- Un programa ensamblador traduce programas en lenguaje assembler a códigos binarios ejecutables.

Algunas Definiciones

- **Instrucción**: orden a ejecutar por la CPU.
- **Lenguaje ensamblador, assembler o simbólico**: se escribe con letras, símbolos, dígitos.
- **Lenguaje de máquina o Código objeto**: es lo que entiende la máquina, 0 y 1.

Cada instrucción escrita en assembler le corresponde un único código máquina.

Algunas Definiciones

Las personas se comunican con Palabras, y usan la voz como el canal para transmitirlos.



Una **Palabra**, en computación, es una secuencia de bits.
El tamaño depende de la arquitectura del procesador.

La CPU y la Memoria se comunican a través de palabras,
las cuales viajan por el Bus de Sistemas.

Algunas Definiciones

Cuando nos dan una dirección nos indican el nombre de la calle y su numeración.



- Si trabajamos con **un dígito decimal** existen 10 posibilidades de combinarlos ($10^1 = 10$), esto es :

0 9

El rango de direcciones sería de 0 a $10^1 - 1$.

- Si trabajamos con **dos dígitos decimales** existen 100 posibilidades de combinarlos ($10^2 = 100$), esto es :

00 99

El rango de direcciones sería de 0 a $10^2 - 1$.

- Si trabajamos con **tres dígitos decimales** existen 1000 posibilidades de combinarlos ($10^3 = 1000$), esto es :

000 999

El rango de direcciones sería de 0 a $10^3 - 1$.

El rango va de 0 a la base^(cantidad de dígitos) — 1.

Cuantos más dígitos más direcciones tendremos.

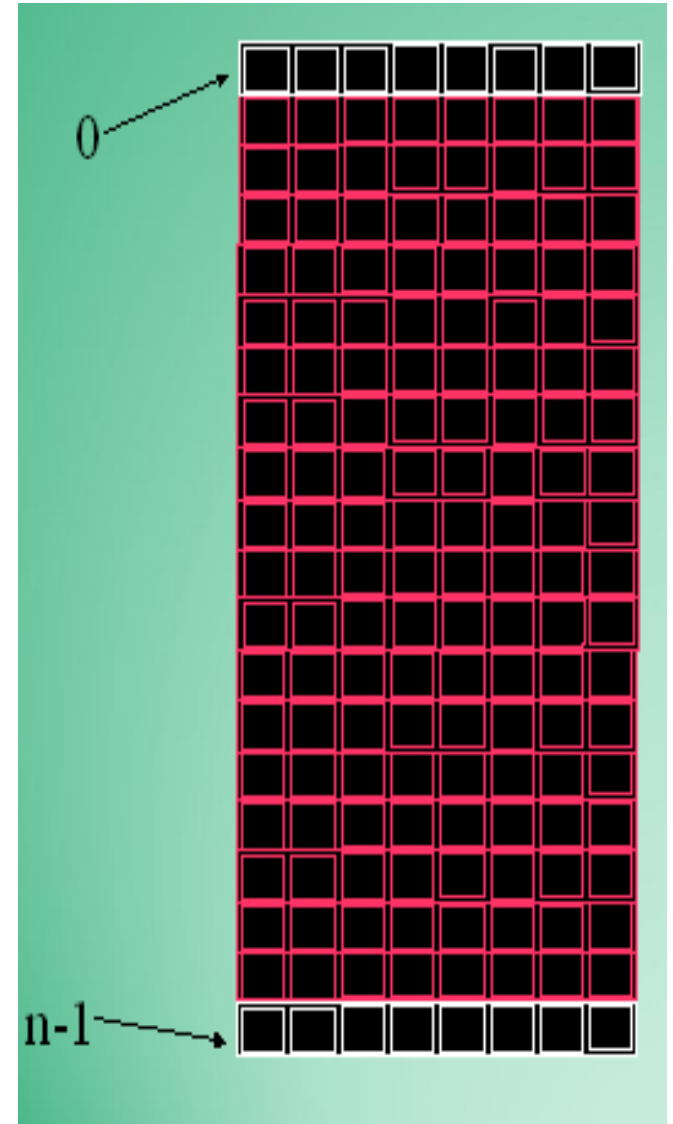
Algunas definiciones

- **Dirección** (en informática):

Es un puntero a una posición de memoria, la cual contiene un dato o una instrucción.

Es el número que identifica en forma unívoca cada palabra.

Las direcciones comienzan en cero, y la última es una unidad menos que el tamaño de memoria.



El gráfico representa una Memoria de tamaño 2^n , donde cada palabra tiene un tamaño de 8 bits.

Un sistema de computación típico



Un sistema de computación típico

Funcionamiento de una computadora:

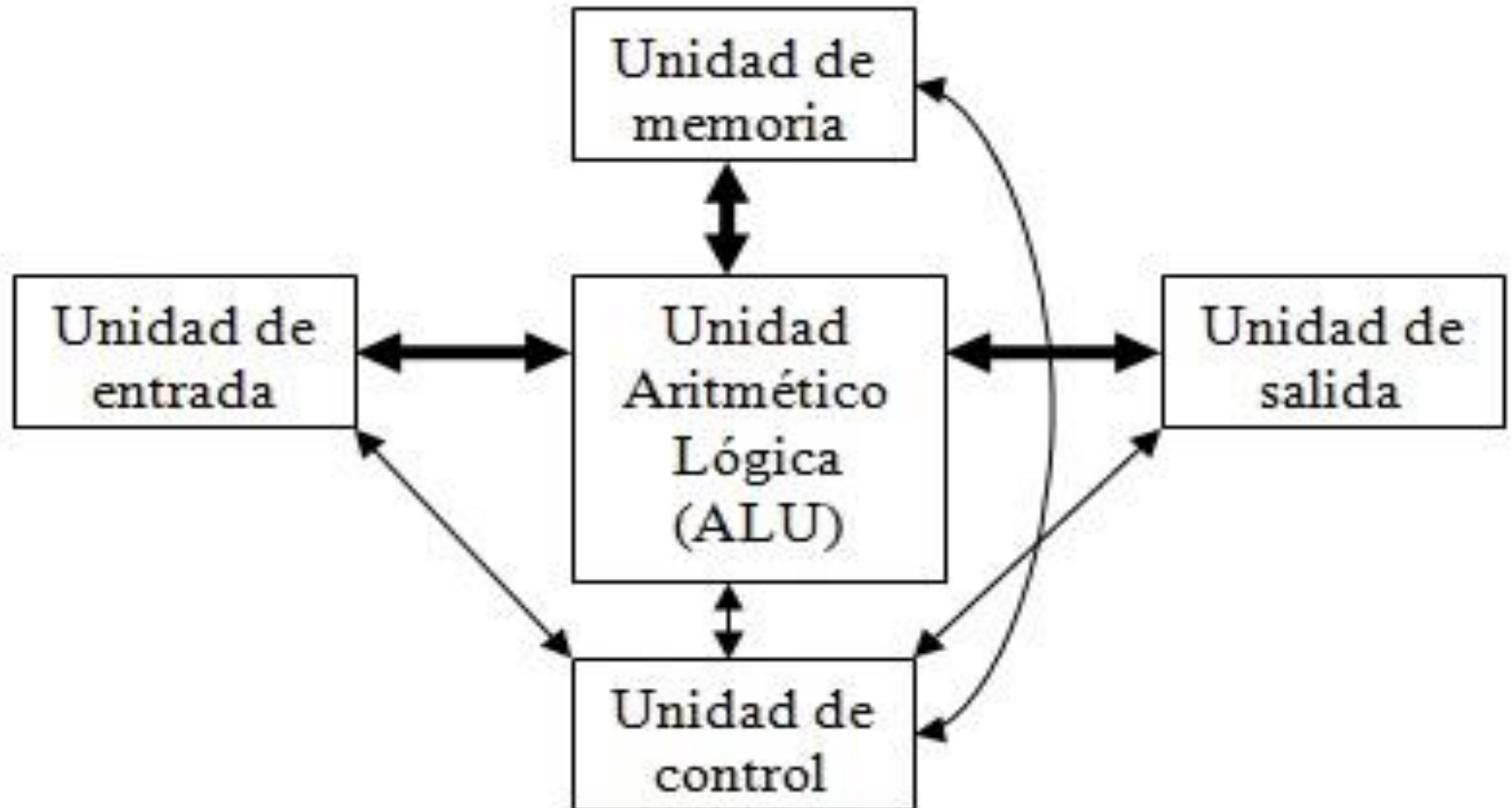
- Extracción de instrucciones de la memoria.
- Interpretación de las mismas.
- Extracción de memoria de los datos implicados en la operación.
- Envío a una unidad encargada de realizar las operaciones y cálculo del resultado.

El Modelo de Von Neumann

**¿Recuerda los componentes del
Modelo de Von Neumann?**

El Modelo de Von Neumann

Consta de cinco componentes principales.



El Modelo de Von Neumann

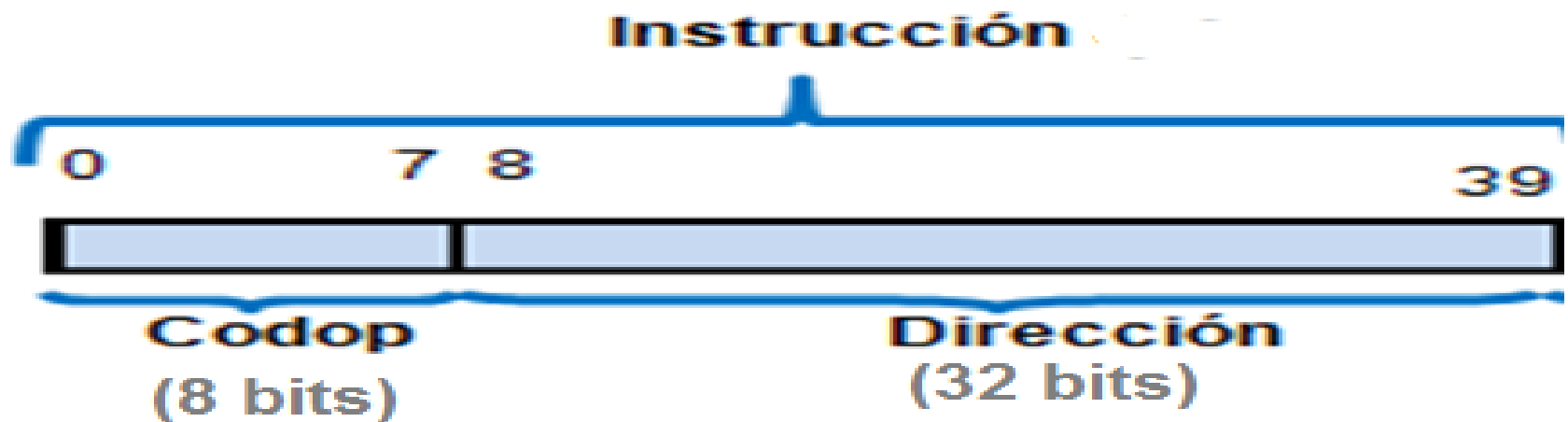
UNIDAD DE MEMORIA

- Las palabras de la memoria RAM se identifican con un número llamado *dirección*.
- Una *palabra* almacena un *dato* o una *instrucción*.
- Una computadora de programa almacenado mantiene las instrucciones y datos del programa en una memoria de acceso aleatorio (RAM) de lectura-escritura.

El Modelo de Von Neumann

UNIDAD DE MEMORIA

- El tamaño de una palabra en este modelo es de 40 bits.
- Hay dos tipos de palabras: palabra dato y palabra instrucción.



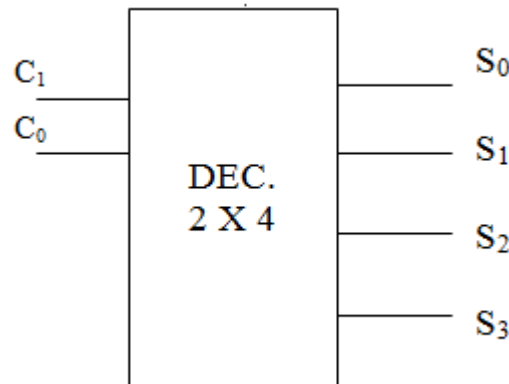
Conociendo el formato de la palabra instrucción se puede deducir:

- Tamaño de la Memoria.
- Rango de Direcciones.

Decodificadores

(tema a ser visto en otro trabajo práctico)

- Un decodificador es un circuito lógico que tiene n entradas y 2^n salidas, se dice que es un **Dec $n \times 2^n$**
- Para una combinación dada de las n entradas, solamente una salida tomará el valor 1, y todas las demás tomarán el valor 0.
- Los **decodificadores** se usan en las **memorias** para **seleccionar solo una palabra** de la misma.



**¿Qué decodificador
usará la Memoria de Von
Neumann?**

El Modelo de Von Neumann

ALU y UC

- Las computadoras están formadas por una gran cantidad de **REGISTROS**.
- Los registros son medios de almacenamiento de información binaria.
- Pueden tener *diferentes tamaños* dentro de una máquina, de acuerdo a su uso y al diseño de la Arquitectura.

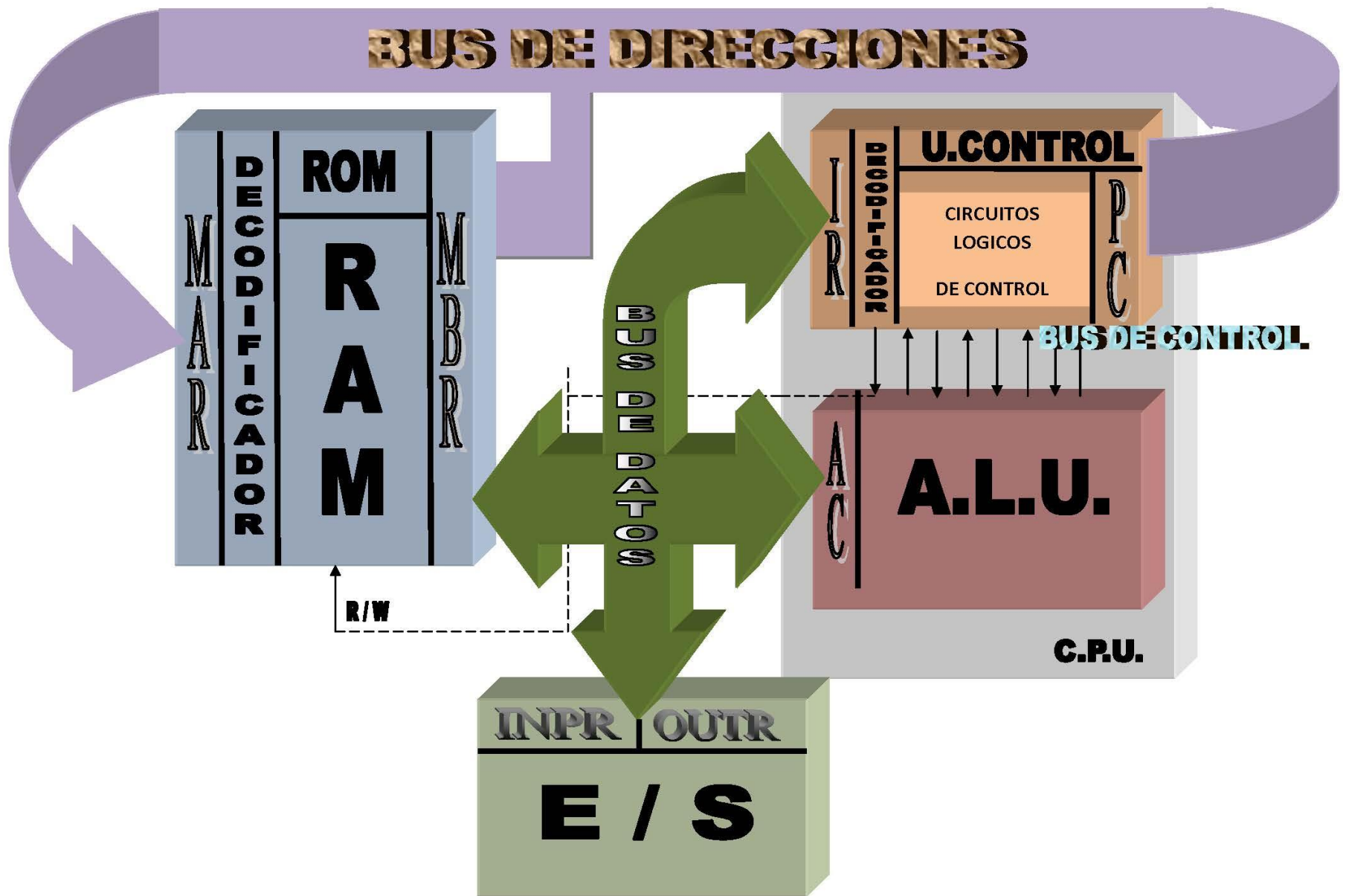
El Modelo de Von Neumann

ALU y UC

Los principales registros de ésta computadora son:

- **MAR** (Registro de Dirección de Memoria, Memory Address Register)
- **MBR** (Registro Temporal de Memoria, Memory Buffer Register)
- **PC** (Contador de Programa , Program Counter)
- **IR** (Registro de Instrucción, Instruction Register)
- **AC** (Acumulador, Accumulator)
- **INPR** (Registro de Entrada, Input Register)
- **OUTR** (Registro de Salida, Output Register)

¿Cuáles son los tamaños en bits de estos registros?



- **MAQUINA DE VON NEUMANN**

El Modelo de Von Neumann

Principios:

1. Los datos y las instrucciones se almacenan en una memoria de lectura-escritura.
2. Los contenidos de esta memoria se direccionan indicando su posición (dirección), sin considera el tipo de dato contenido en la misma.
3. La ejecución se produce siguiendo una secuencia de instrucción tras instrucción (a no ser que dicha secuencia se modifique explícitamente).

El Modelo de Von Neumann

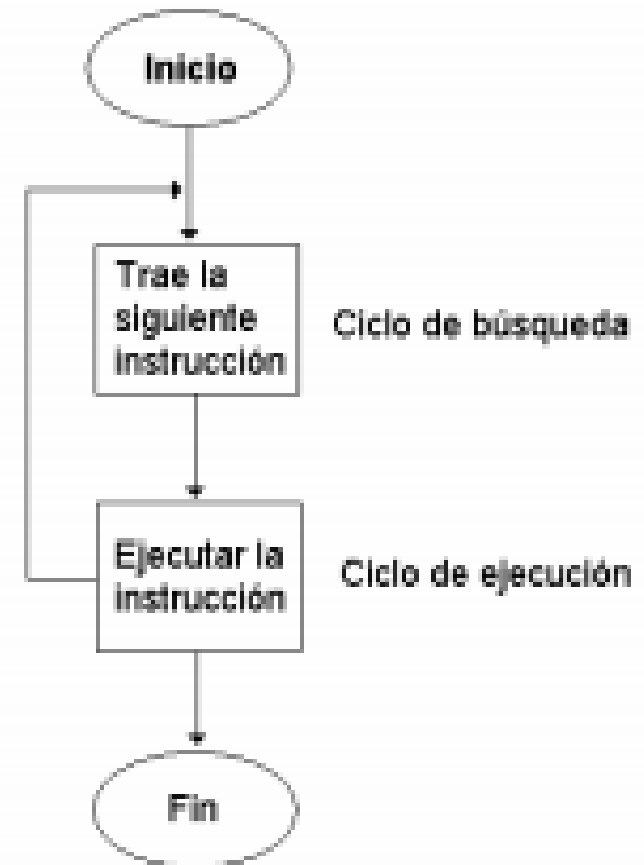
- La función de una computadora es la ejecución de programas.
- La CPU es quien se encarga de ejecutar dichas instrucciones a través de un ciclo denominado ciclo instrucciones.

El Modelo de Von Neumann

¿Qué entiende por ciclo de instrucciones?

Desde una visión muy simple, el llevar a cabo las instrucciones, consiste de dos etapas:

- A) Leer (fetch) instrucción de memoria
 - B) Ejecutar (execute) instrucción
- El programa se ejecuta a través de una iteración constante de estos dos etapas



Vista simplificada del ciclo de instrucción

El Modelo de Von Neumann

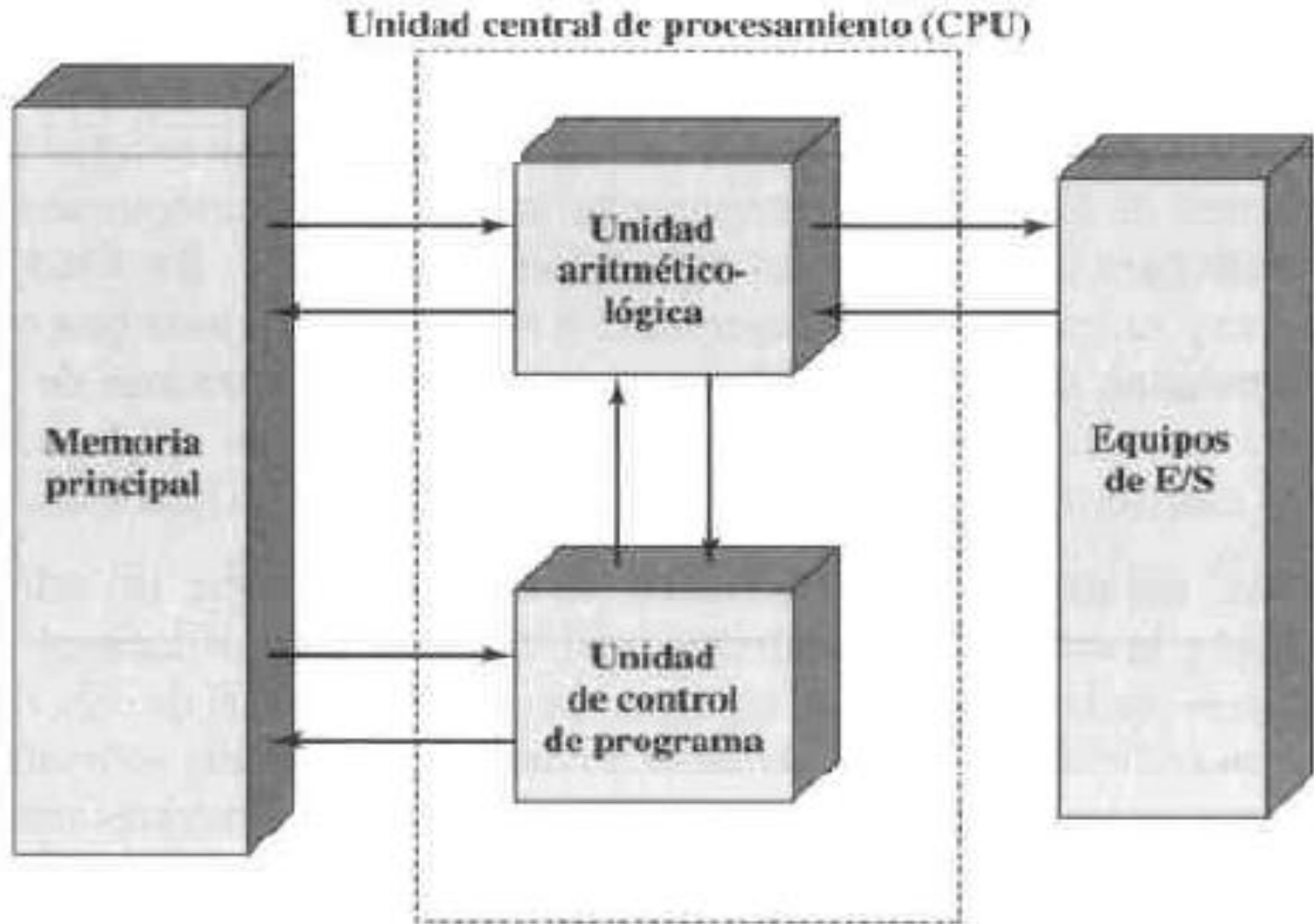
**¿Cómo se realizaría el
Ciclo de Instrucción
en el modelo de
Von Neumann?**



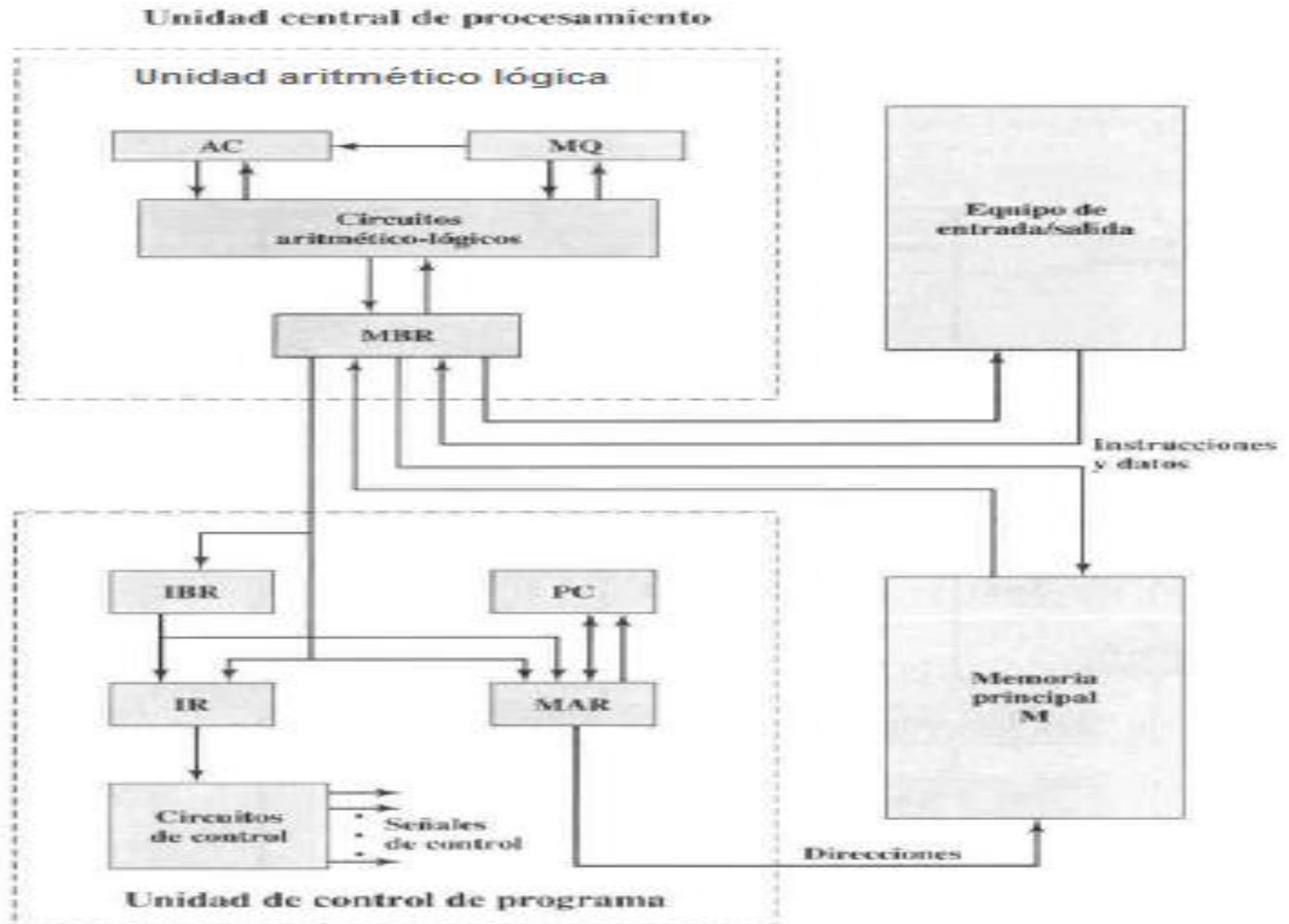
El Computador IAS

**¿Cuáles son los componentes del
computador IAS?**

Computadora IAS



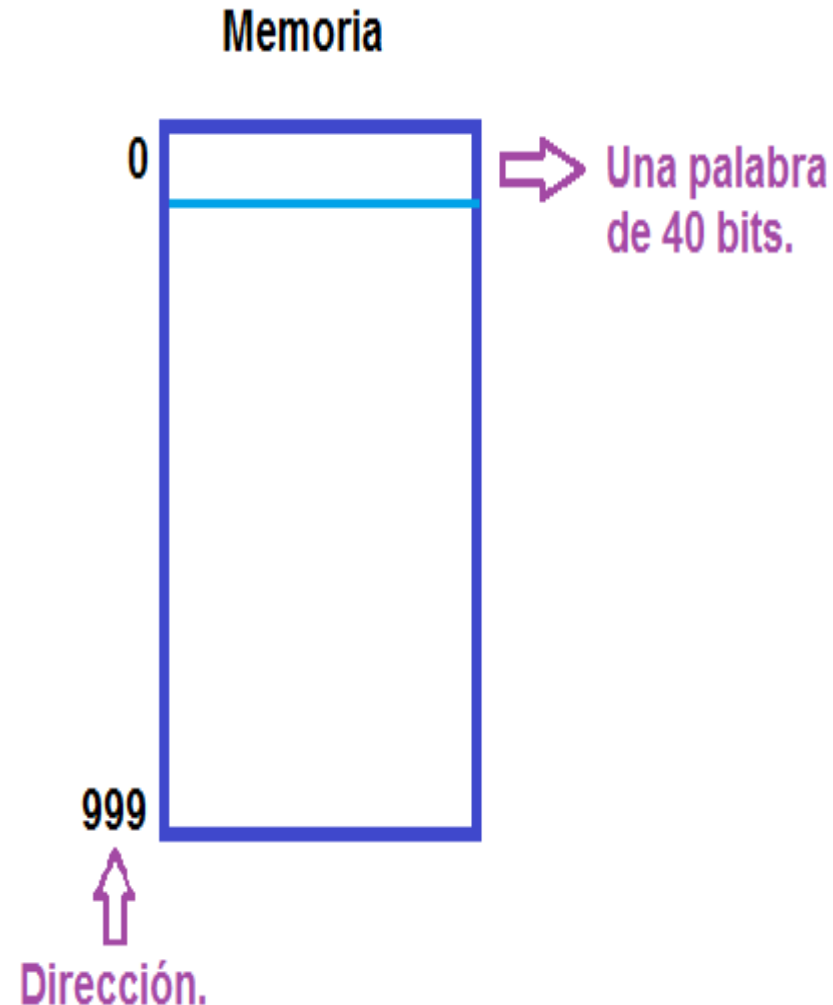
● Estructura ampliada del IAS



Computadora IAS

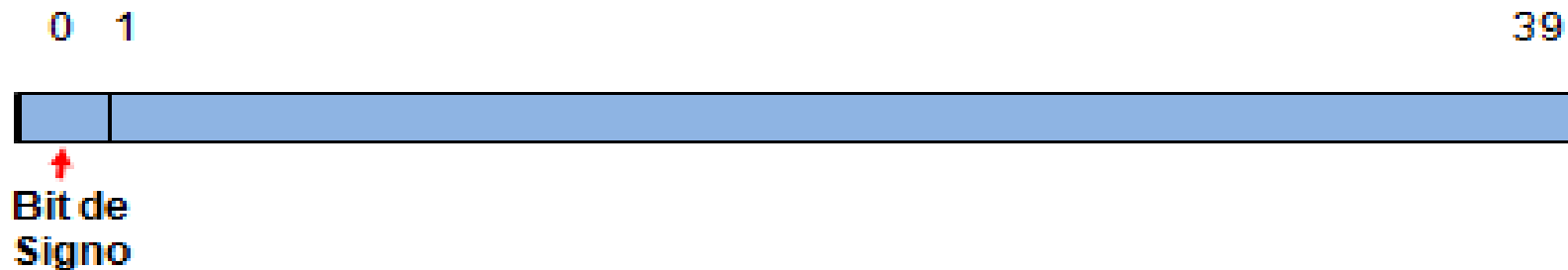
- **MEMORIA** de IAS:

Consiste en 1000 posiciones de almacenamiento, **PALABRAS**, de 40 bits cada una (5 bytes). Tanto los datos como las instrucciones se almacenan ahí.



Formato de la palabra en IAS

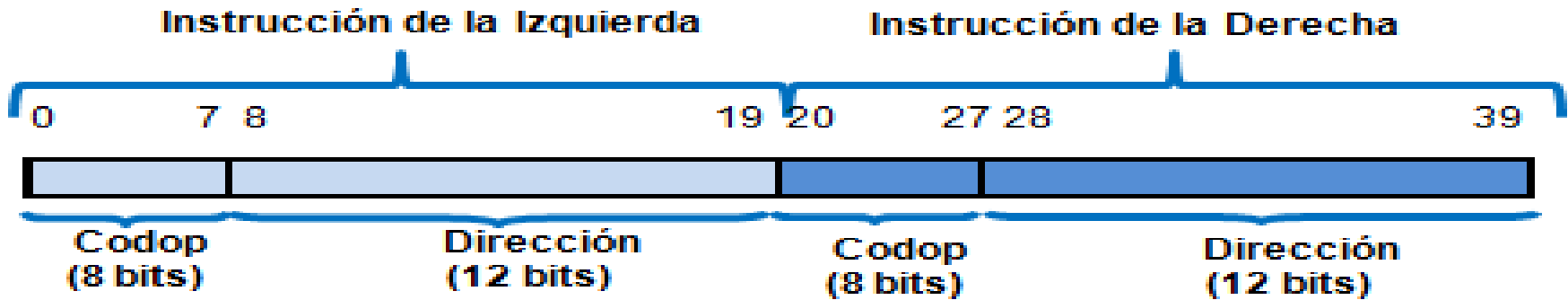
Palabra Número



Tiene un bit de signo y 39 bits de magnitud.

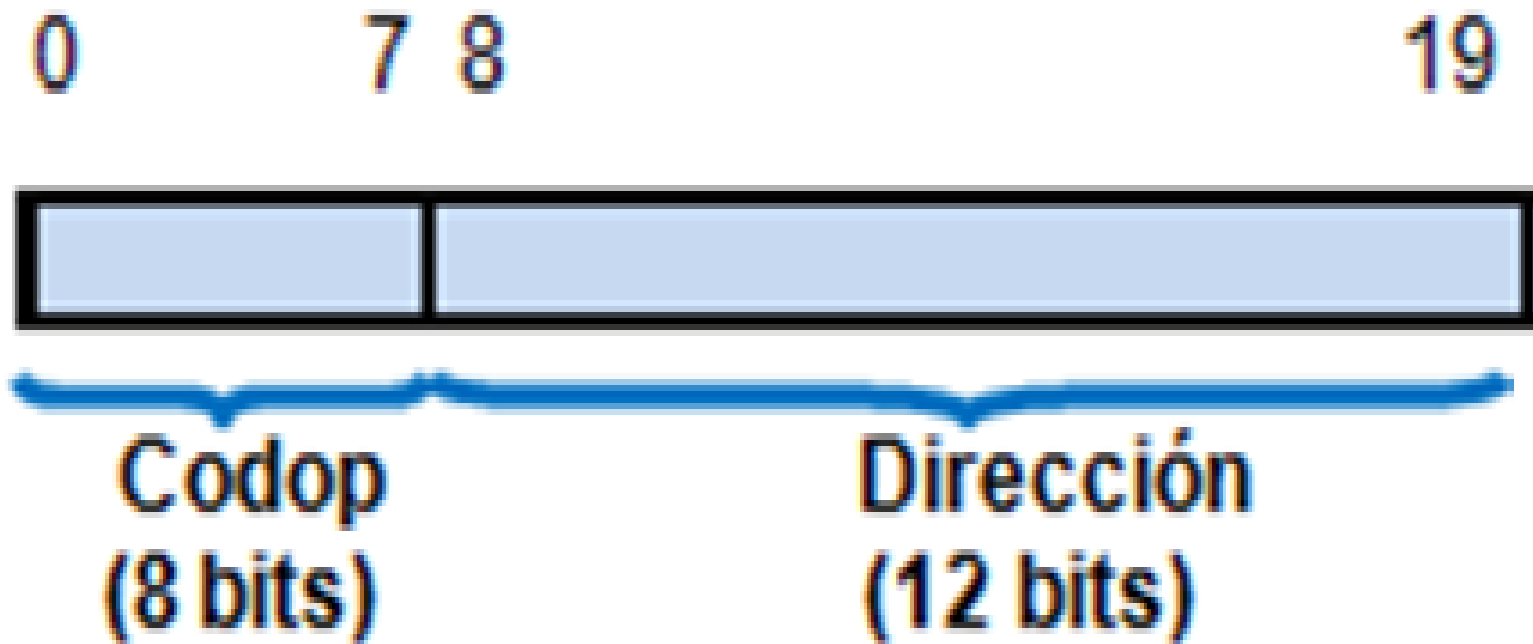
Formato de la palabra en IAS

Palabra Instrucción



Tiene 40 bits, con lo cual al extraer una palabra de memoria se sacan 2 instrucciones al mismo tiempo, cada una de 20 bits.

Formato de la instrucción en IAS



Formato de la instrucción en IAS

CODOP (Código de Operación): tiene información de la instrucción que se quiere ejecutar.

Tiene en total 8 bits, que indican que pueden haber $2^8 = 256$ operación diferentes.

DIRECCIÓN: nos indica una posición de memoria, de 0 a 999, en la cuál se encuentra el operando.

Tiene 12 bits, es decir que se cuenta con $2^{12} = 4 \text{ K} = 4096$ bytes o 4096 posiciones de memoria posibles de ser usadas en la instrucción.

$1000 \text{ palabras} \times 40 \text{ bits} = 40000 \text{ bits} / 8 = 5000 \text{ bytes}$
aproximadamente 4096 bytes.

Para escribir $999_{(10)}$ en binario usamos 12 bits
 $001111100111_{(2)}$

Computadora IAS

- **UNIDAD DE CONTROL (UC):**

Capta instrucciones de la memoria y las ejecuta una a una.

- **UNIDAD ARITMÉTICO LÓGICA (ALU):**

Realiza operaciones con los datos binarios obtenidos.

Registros de la Computadora IAS

Utiliza los mismos registros que el modelo de Von Neumann, y además incorpora los Registros:

- IBR (Instruction Buffer Register)
- MQ (Multiplier Quotient)

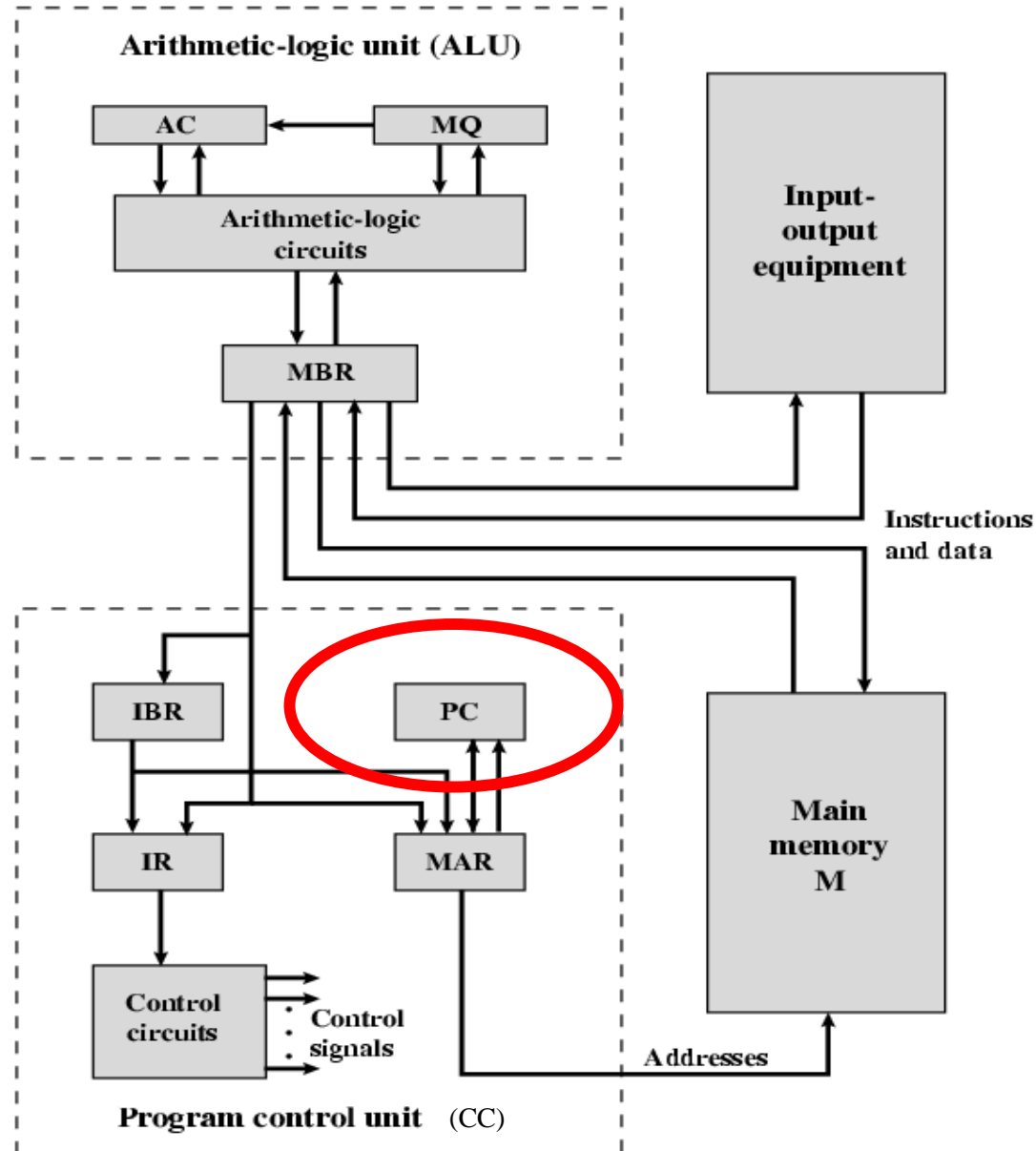
Registros de la Computadora IAS

Contador de Programa PC (Program Counter):

Contiene la dirección de la próxima pareja de instrucciones que van a ser captadas de la memoria.

**Tamaño
en bits?**

12 bits



Registros de la Computadora IAS

Registro de Dirección
de Memoria

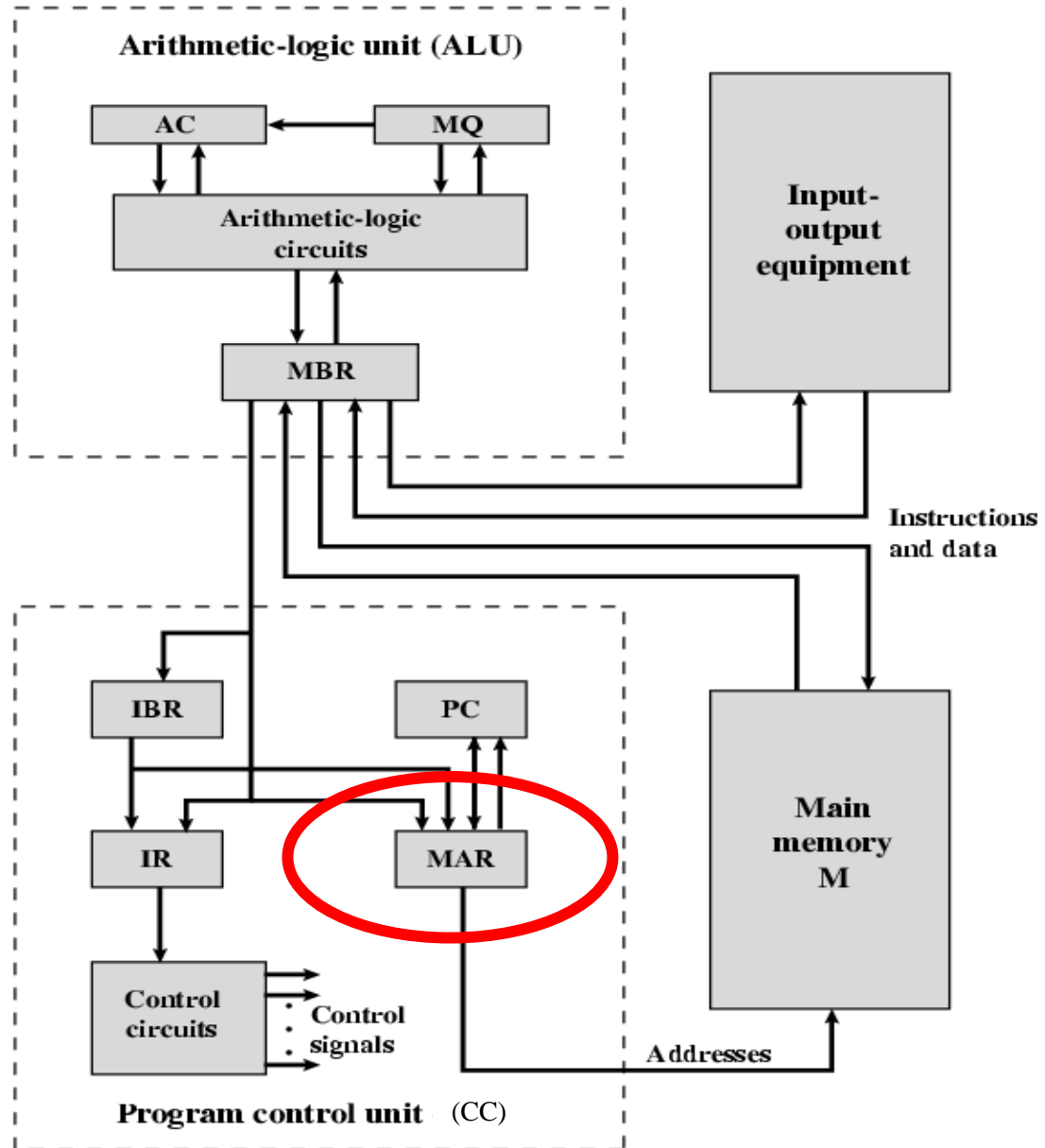
MAR

(Memory Address
Register):

Especifica la dirección
en memoria de la
palabra que va a ser
escrita o leída en MBR.

Tamaño
en bits?

12 bits



Registros de la Computadora IAS

Registro Temporal de Memoria

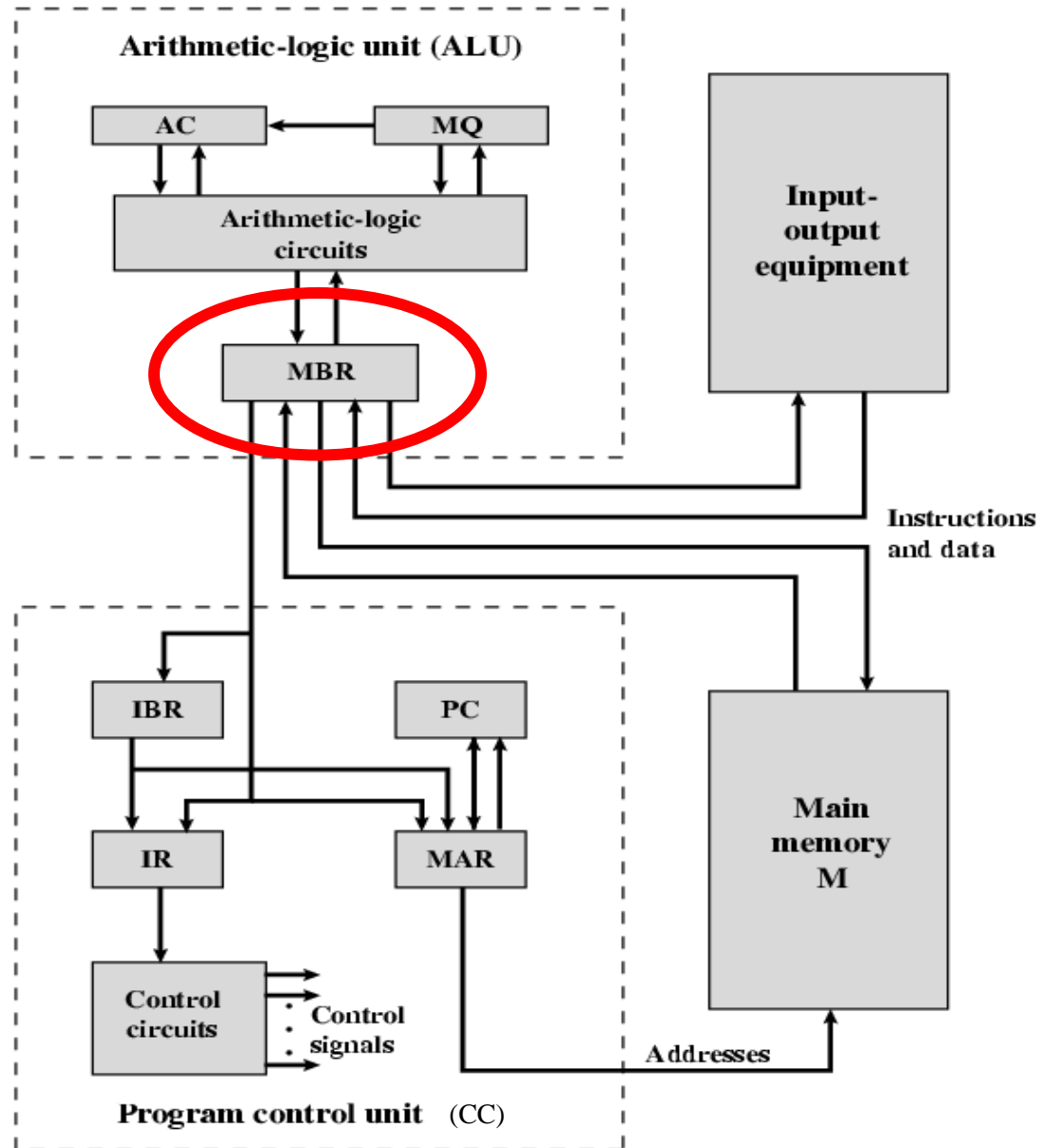
MBR

(Memory Buffer Register):

Contiene una palabra que debe ser almacenada en memoria, o que viene desde la memoria.

Tamaño
en bits?

40 bits



Registros de la Computadora IAS

Registro Temporal de Instrucción

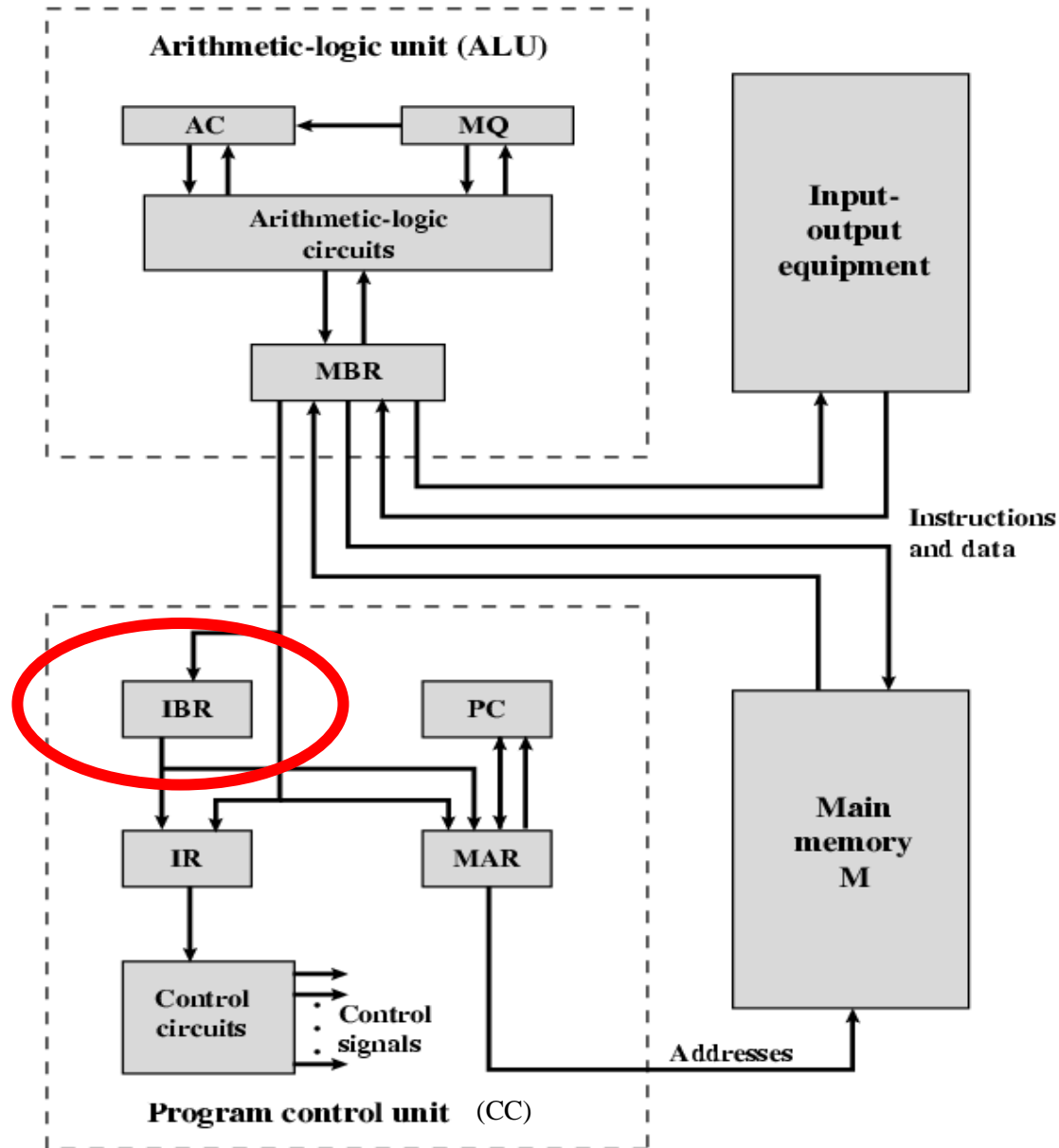
IBR

(Instruction Buffer Register):

Almacena temporalmente la instrucción contenida en la parte *derecha* de una palabra (**instrucción en espera**)

Tamaño en bits?

20 bits



Registros de la Computadora IAS

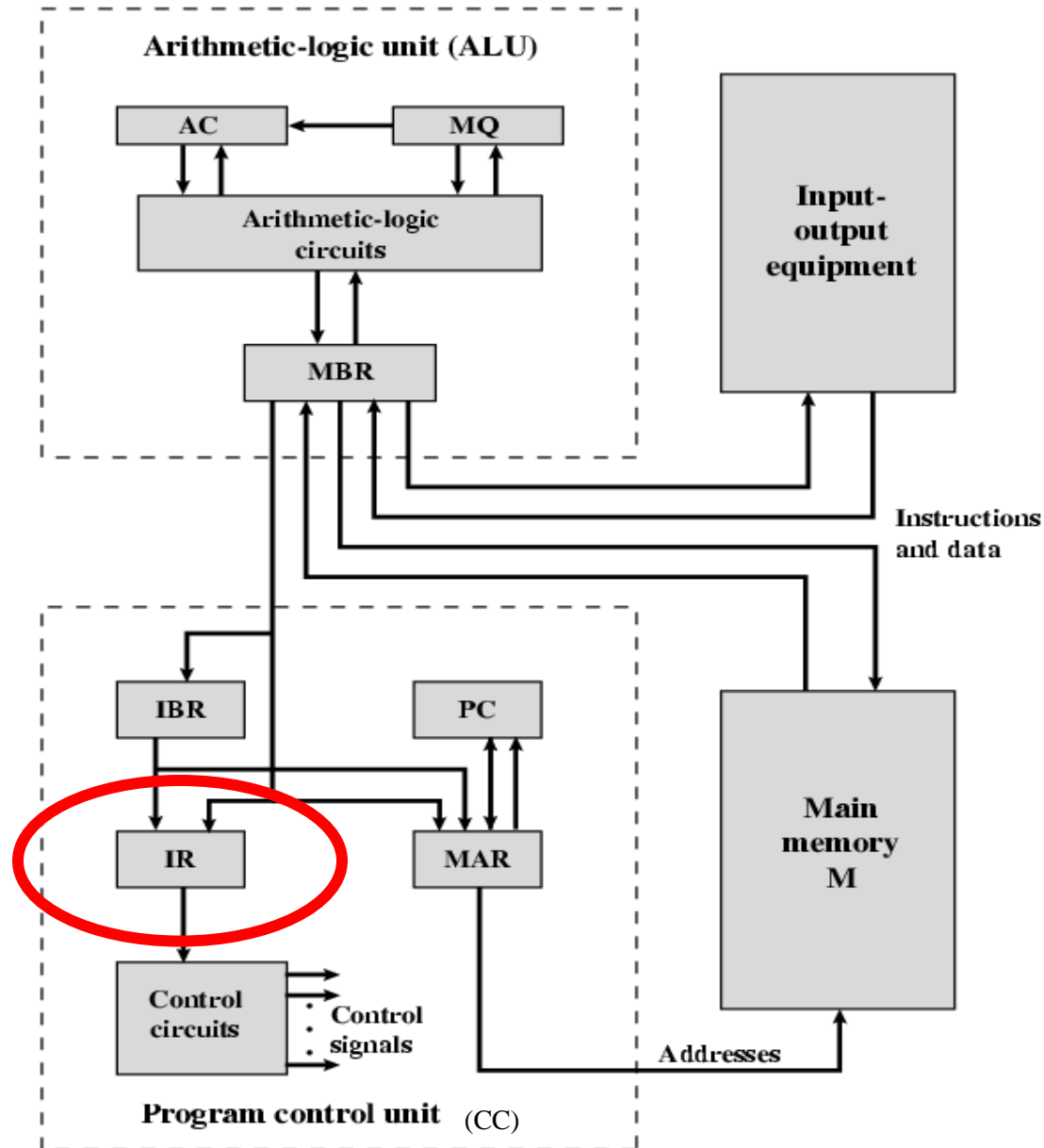
Registro de Instrucción IR

(Instruction Register):

Contiene los bits del código de operación de la instrucción que se va a ejecutar.

**Tamaño
en bits?**

8 bits



Registros de la Computadora IAS

Acumulador

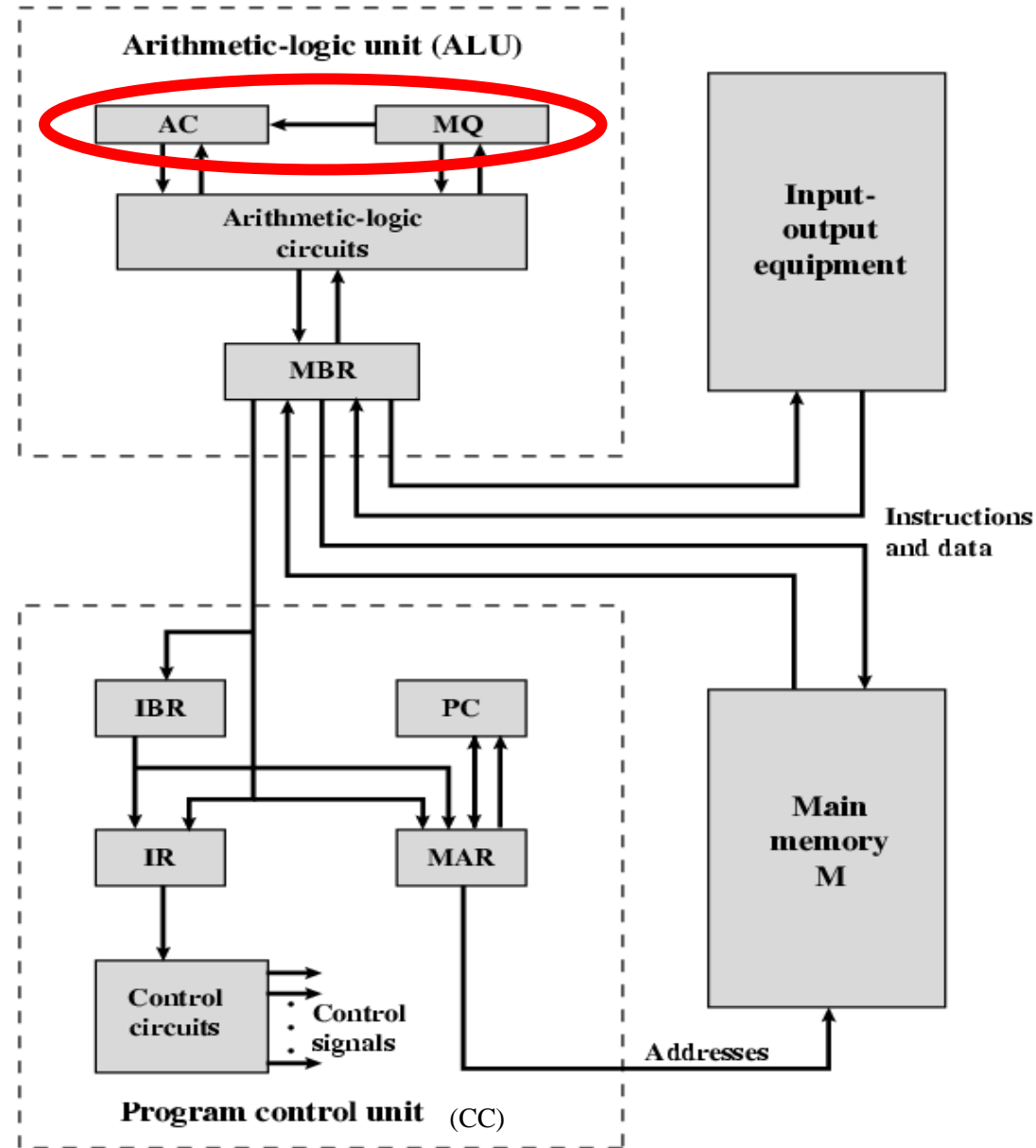
AC y Multiplicador

Cociente MQ:

Almacenan operandos y resultados de operaciones de la ALU temporalmente.

Tamaños en bits? **40 bits c/u**

Por ejemplo, el resultado de multiplicar dos operandos de 40 bits es uno de 80 bits, los 40 bits más significativos se almacenan en AC y los menos significativos en MQ.



Multiplicación en binario

(tema a ser visto en otro trabajo práctico)

La tabla de multiplicar para números binarios es la siguiente:

x	0	1
0	0	0
1	0	1

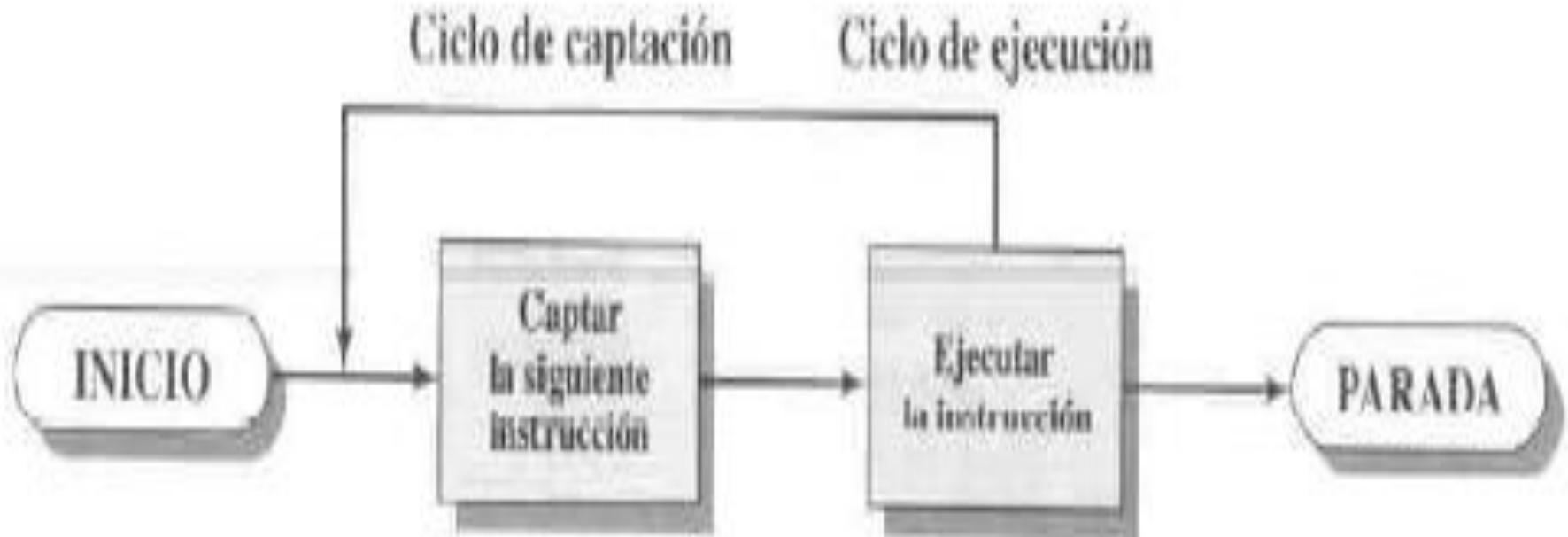
$$\begin{array}{r} 6 \\ \times 5 \\ \hline 30 \end{array}$$

$$\begin{array}{r} 0110 \\ \times 0101 \\ \hline 0110 \\ 0000 \\ 0110 \\ 0000 \\ \hline 0011110 \end{array}$$

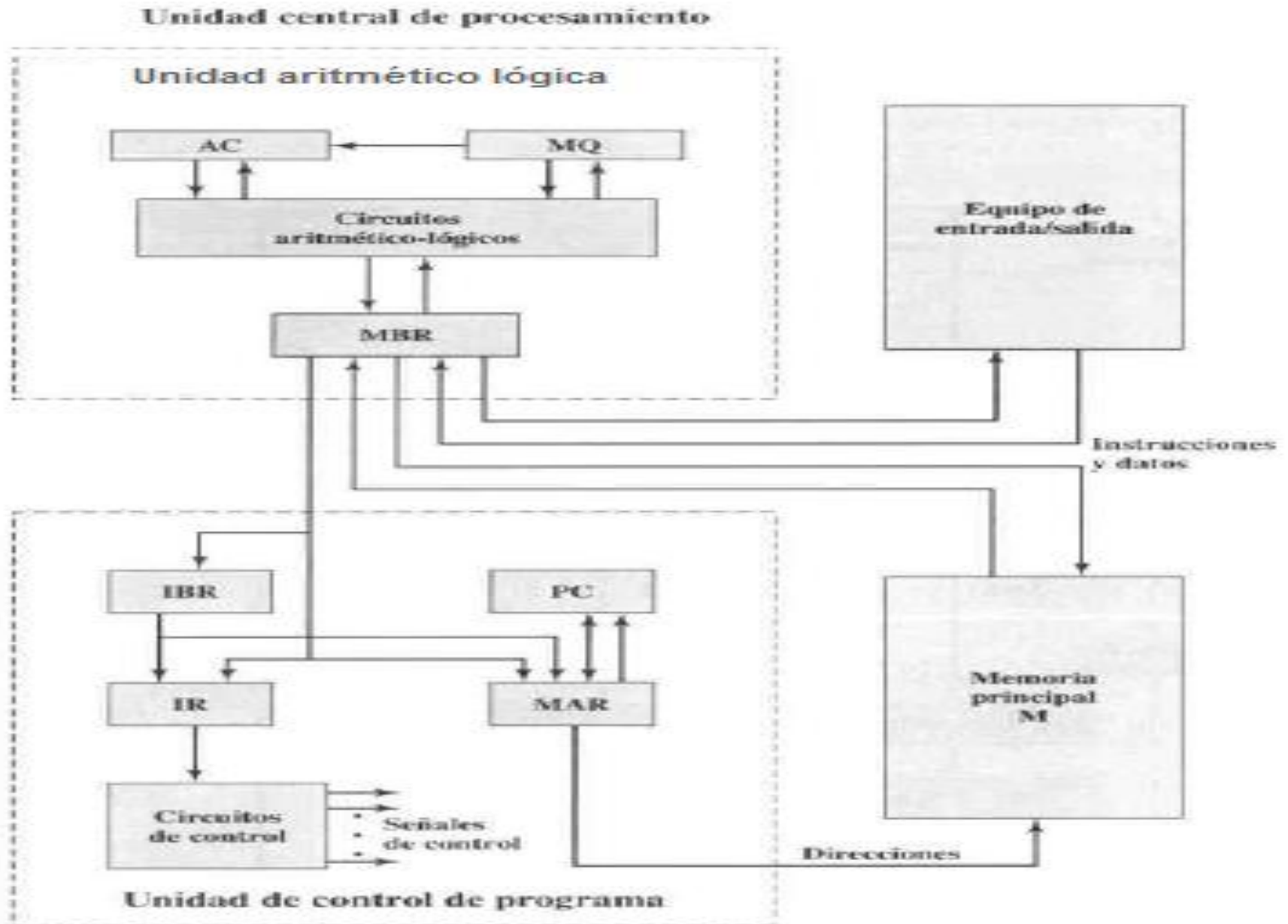
El Ciclo de Instrucción en IAS

Un ciclo de instrucción es el procesamiento que requiere una instrucción.

La ejecución de un programa consiste en la repetición de éste ciclo.



• Estructura ampliada del IAS



El Ciclo de Instrucción en IAS

**¿Cómo se realizaría el
Ciclo de Instrucción
en el modelo IAS?**

• El Ciclo de Instrucción en IAS

Cada ciclo de instrucción consta de dos subciclos:

- **CICLO DE CAPTACIÓN (búsqueda):**

¿Por dónde se comienza?

¿Cómo continuamos?

Sigan el camino de las líneas (buses)

IAS permite obtener 2 instrucciones de la memoria en el mismo ciclo de búsqueda.

No se ejecutan las dos a la vez.

• El Ciclo de Instrucción en IAS

• CICLO DE EJECUCIÓN:

¿Quién tiene la información sobre la operación a realizar (CODOP)?

¿Esa información a dónde va?

¿Qué le falta a la ALU para trabajar?

¿Ese dato dónde está?

¿Quién tiene la dirección del dato?

Vamos con la dirección por el Bus de Direcciones a la Memoria y sacamos la Palabra Dato. ¿Dónde la colocamos?

• El Ciclo de Instrucción en IAS

• CICLO DE EJECUCIÓN (continuación):

La ALU ya sabe que operación debe hacer con el dato recibido y realiza la ejecución.

El resultado ¿dónde lo almacena?

Si el resultado fuera más grande, ¿dónde almacena los otros bits?

Luego se repite lo mismo con la instrucción almacenada en el IBR.

¿Cómo se termina?

¿En cuánto se incrementa? ¿Para qué se lo hace?

Instrucciones en IAS

Las Instrucciones en IAS

El computador IAS tiene un total de 21 instrucciones, que se clasifican en cinco grupos.

Las Instrucciones en IAS

INSTRUCCIONES DE TRANSFERENCIA DE DATOS:

- Se transfieren datos entre la memoria y los registros de la ALU o entre dos registros de la ALU.

Codop	Instrucción	Descripción
00001010	LOAD MQ	Transferir el contenido del registro MQ al AC
00001001	LOAD MQ, M(X)	Transferir el contenido de la posición de memoria X al MQ
00100001	STOR M(X)	Transferir el contenido del AC a la posición de memoria X
00000001	LOAD M(X)	Transferir M(X) al AC
00000010	LOAD -M(X)	Transferir -M(X) a AC
00000011	LOAD M(X)	Transferir el valor absoluto de M(X) al AC
00000100	LOAD - M(X)	Transferir - M(X) a AC

Las Instrucciones en IAS

INSTRUCCIONES DE SALTO INCONDICIONAL:

- Normalmente, la unidad de control ejecuta instrucciones secuencialmente en la memoria. Las instrucciones de salto pueden cambiar ese orden (ej. Operaciones repetitivas).

Codop	Instrucción	Descripción
00001101	JUMP M(X, 0:19)	Captar la instrucción indicada por la mitad izquierda de M(X)
00001110	JUMP M(X, 20:39)	Captar la instrucción indicada por la mitad derecha de M(X)

Las Instrucciones en IAS

INSTRUCCIONES DE SALTO CONDICIONAL:

- El salto depende de una condición, esto permite puntos de decisión.

Codop	Instrucción	Descripción
00001111	JUMP +M(X, 0:19)	Si $AC \geq 0$ saltar a la instrucción indicada por la mitad izquierda de M(X)
00001000	JUMP +M(X, 20:39)	Si $AC \geq 0$ saltar a la instrucción indicada por la mitad derecha de M(X)

Las Instrucciones en IAS

INSTRUCCIONES DE ARITMÉTICA:

- Son las operaciones realizadas por la ALU.

Codop	Instrucción	Descripción
00000101	ADD M(X)	$AC \leftarrow AC + M(X)$
00000111	ADD M(X)	$AC \leftarrow AC + M(X) $
00000110	SUB M(X)	$AC \leftarrow AC - M(X)$
00001000	SUB M(X)	$AC \leftarrow AC - M(X) $
00001011	MUL M(X)	$[AC][MQ] \leftarrow AC \times M(X)$
00001100	DIV M(X)	$[AC][MQ] \leftarrow AC \div M(X)$
00010100	LSH	$AC \leftarrow AC \times 2$
00010101	RSH	$AC \leftarrow AC \div 2$

Las Instrucciones en IAS

INSTRUCCIONES DE MODIFICACIÓN DE DIRECCIONES:

- Permite que la ALU haga operaciones con las direcciones y las inserte en instrucciones almacenadas en memoria. Esto permite una considerable flexibilidad de direccionamiento en un programa.

Codop	Instrucción	Descripción
00010010	STOR M(X, 8:19)	Reemplazar el campo de dirección de la izquierda de M(X) por los 12 bits de la derecha de AC.
00010011	STOR M(X, 28:39)	Reemplazar el campo de dirección de la derecha de M(X) por los 12 bits de la derecha de AC.



Las Instrucciones en IAS

(resumen)

CODOP	INSTRUCCIÓN EN ASSEMBLER	DESCRIPCION
00100001	STOR M(X)	Transferir el contenido del AC a la posición de memoria X
00000001	LOAD M(X)	Transferir M(X) al AC
00000101	ADD M(X)	$AC \leftarrow AC + M(X)$
00000110	SUB M(X)	$AC \leftarrow AC - M(X)$
00001011	MUL M(X)	$[AC][MQ] \leftarrow AC \times M(X)$
00001100	DIV M(X)	$[AC][MQ] \leftarrow AC \div M(X)$
00010100	LSH	$AC \leftarrow AC \times 2$
00010101	RSH	$AC \leftarrow AC \div 2$

● Ejercicios con Instrucciones.

Ejercicio:

Escribir un programa que lea un dato de la dirección 600 de memoria y lo almacene en la dirección 601.

LOAD M(600)
STOR M(601)

% transfiere el contenido de 600 al AC

% transfiere AC a la posición de memoria 601

● Ejercicios con Instrucciones.

Ejercicio:

Escribir un programa que sume el número almacenado en la posición 867 más el número almacenado en la posición 562. El resultado de la suma se debe almacenar en la posición 778.

LOAD M(867)
ADD M(562)
STOR M(778)

% transfiere el contenido de la posición 867 al AC

% $AC \leftarrow AC + M(562)$

% transfiere el contenido de AC a la posición de M(778)

● Ejercicios con Instrucciones.

¿Cómo se almacena el programa en 0 y 1 (Código Máquina)?

Lenguaje Assembler	Código Máquina																			
	CODOP (8 bits)								Dirección (12 bits)											
									2048	1024	512	256	128	64	32	16	8	4	2	1
LOAD M(867)	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0	0	0	1	1
ADD M(562)	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	0	1	0
STOR M(778)	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0

IAS Simulator V 1.0.4

Link:

[https://www.ic.unicamp.br/~edson/disciplinas/mc404/
2017-2s/abef/IAS-sim/](https://www.ic.unicamp.br/~edson/disciplinas/mc404/2017-2s/abef/IAS-sim/)

IAS Simulator V 1.0.4

En el **mapa de memoria** hay que cargar las **direcciones y el programa** (instrucciones y datos) en **hexadecimal**.

Luego presionar el botón "**Load Me. Map into IAS Memory**", con esto aparecerá el programa cargado en la parte central inferior de la página "**MAIN MEMORY**".

También se puede adjuntar un archivo .txt y luego presionar el botón "**Load Map from File**".

**Ejemplos con
máquinas
hipotéticas.**

Ejercicios con Instrucciones.

Ejercicio (màquina hipotética):

Dada una **hipotética máquina** que maneja una Memoria de 128 MB, cuyo Procesador posee un conjunto de 3 instrucciones (sumar, leer de la memoria, escribir en la memoria). Los códigos numéricos asignados a cada operación son desde el número 0 (cero) en adelante, desde la primera operación que es la suma. Esta máquina trabaja con un formato de instrucción que puede manejar una dirección por vez. Se pide:

a) ¿Cuántos bits se necesitan para representar las tres instrucciones?

0 = 00 → Sumar

1 = 01 → Leer de memoria

2 = 10 → Escribir en memoria

**$2^2 = 4$, abarca las
3 instrucciones.
→ Dos bits para
el CODOP.**

Ejercicios con Instrucciones.

Ejercicio (continuación):

b) Dibujar el formato de instrucción correspondiente, indicando el nombre de cada campo, su función y longitud.

Memoria de 128 MB

$2^x = 128 \text{ MB} \rightarrow 2^{27} \rightarrow 27 \text{ bits para la dirección.}$



CODOP (Campo Código de Operación): almacena la operación a realizar.

DIRECCIÓN: indica la dirección de memoria de la cuál se extraerá un dato o se guardará un dato en ella.

Ejercicios con Instrucciones.

Ejercicio (continuación):

c) Con el formato del punto anterior, realizar el código máquina de la instrucción de lectura de un dato de la posición 900 de memoria.

01	000000000000000000001110000100
----	--------------------------------

Ejercicios con Instrucciones.

Ejercicio (continuación):

d) Escribir la instrucción del punto anterior en lenguaje assembler, de acuerdo a la arquitectura IAS.

LOAD M(900)

Ciclo de Instrucción

Ejemplo (máquina hipotética)

Suponiendo una *máquina hipotética* con las siguientes características:

- AC: único registro de datos, almacenamiento temporal.
- PC: registro con la dirección de la instrucción.
- IR: registro instrucción.
- Instrucciones y datos de 16 bits.



Formato de enteros.



Ciclo de Instrucción

Ejemplo (máquina hipotética)



Formato de instrucción.

$2^{12} = 4096$ (4K) palabras de memoria que se pueden direccionar.

$2^4 = 16$ codops diferentes. A continuación lista parcial:

Codop	Descripción
0001	Carga AC desde memoria
0010	Almacenar AC en memoria
0101	Sumar al AC un dato de memoria

Ciclo de Instrucción

Ejemplo (máquina hipotética)

Nota: se utiliza notación hexadecimal, cada dígito representa cuatro bits.

PASO 1:

El PC contiene el valor 300.

Se carga el contenido de esa dirección (una instrucción) en el IR.

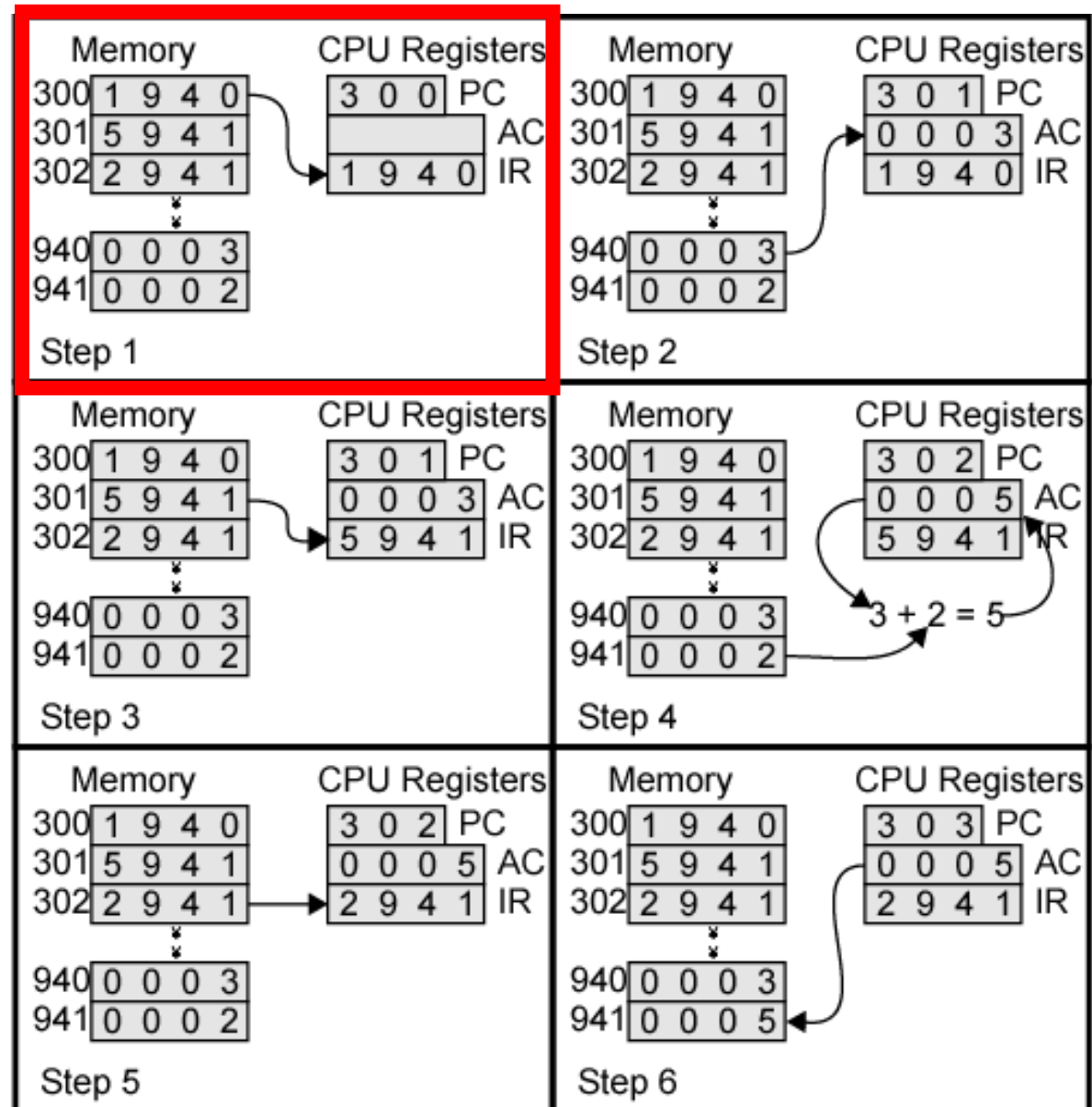


Figura en Cap. 3,
Pag. 64, Libro
Stalling.

Ciclo de Instrucción

● Ejemplo (máquina hipotética)

PASO 2:

¿Qué operación se tiene que ejecutar?

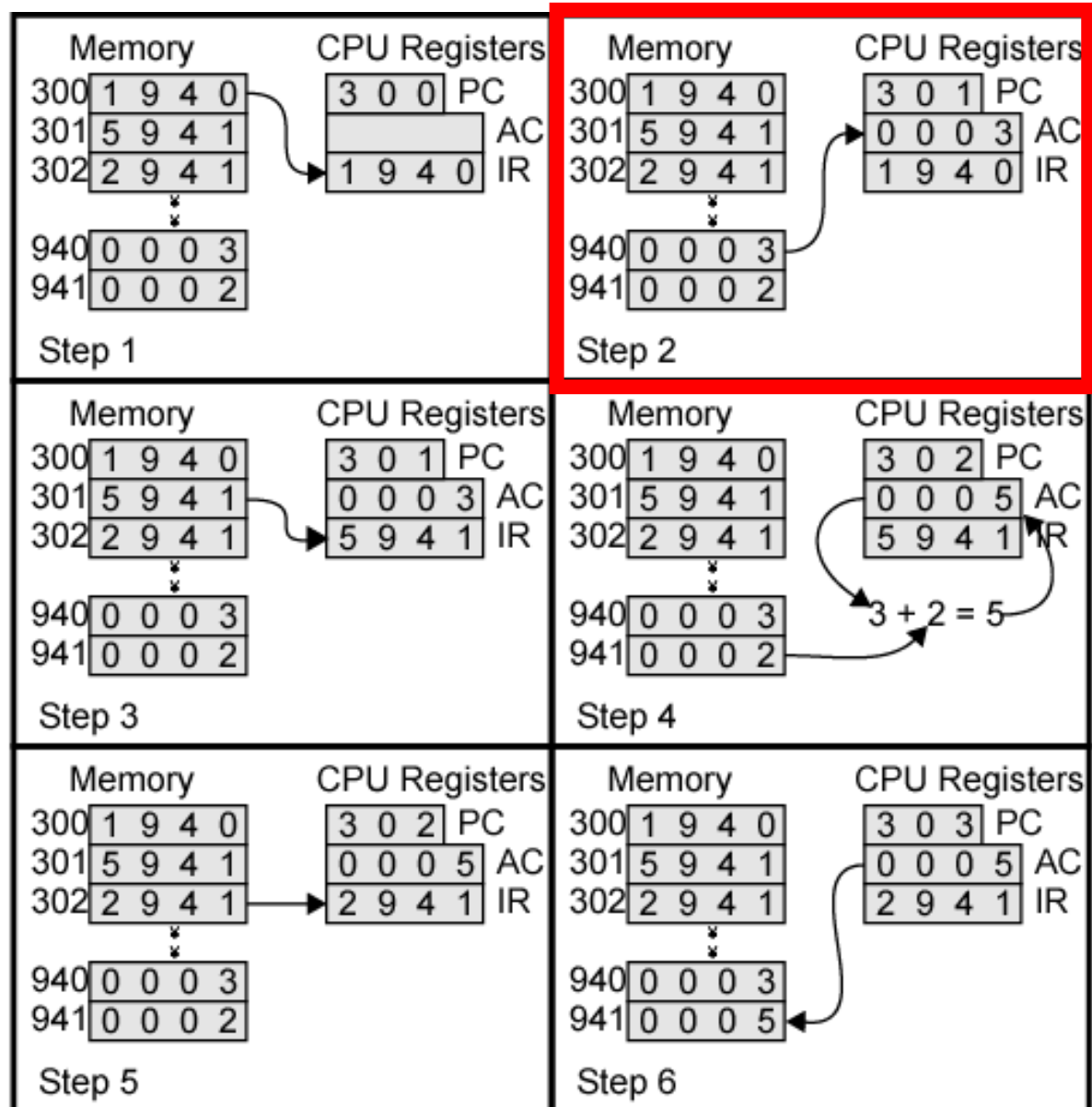
El primer dígito del IR indica la operación:

$$1_{(16)} = 0001_{(2)}$$

El AC se va a cargar con un dato de la memoria.

Los restantes 12 bits especifican la dirección (es decir "940").

El registro de PC se incrementa en uno (toma el valor 301).

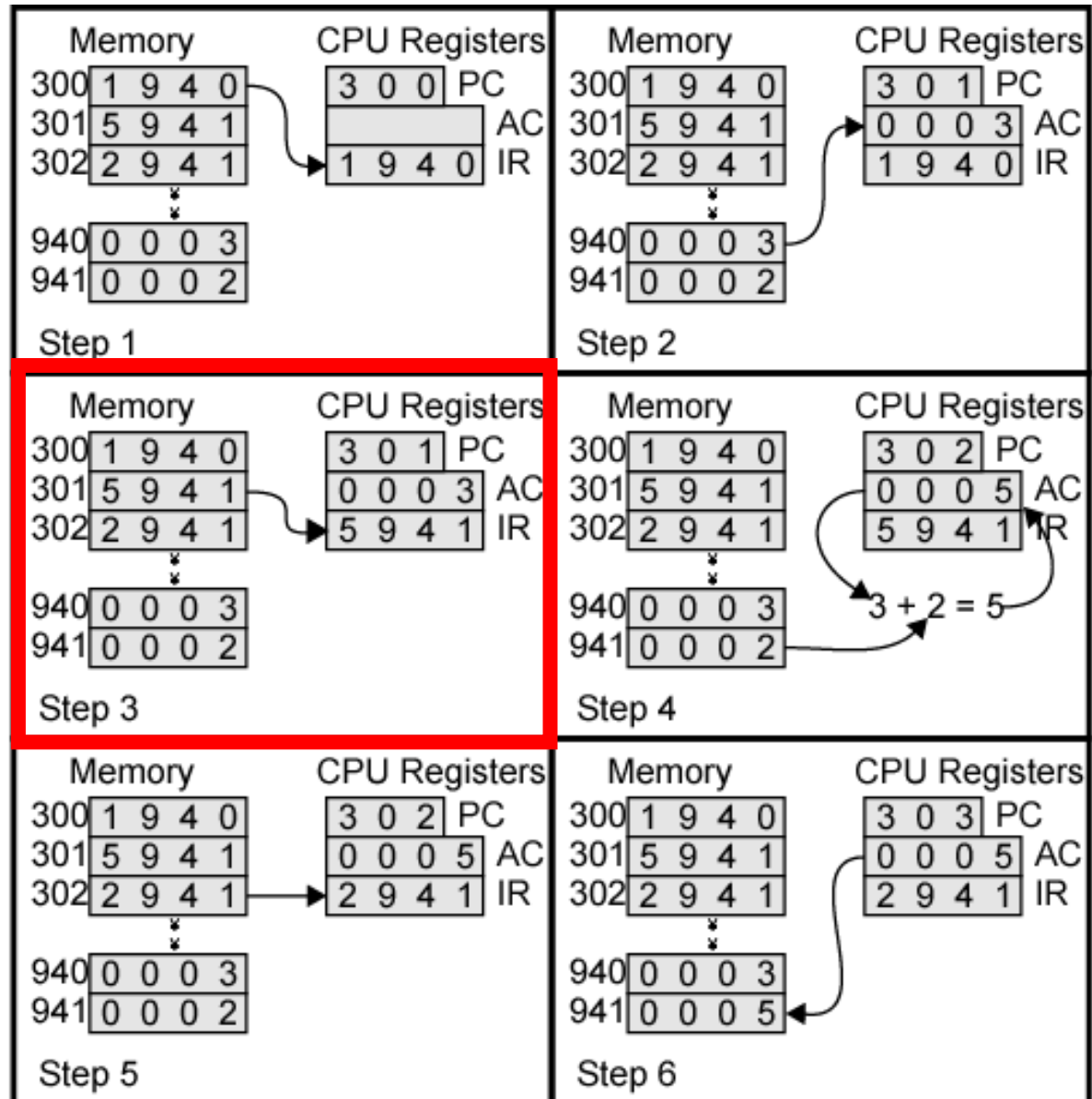


Ciclo de Instrucción

● Ejemplo (máquina hipotética)

PASO 3:

Se capta la siguiente instrucción que se carga en IR (5941).



Ciclo de Instrucción

● Ejemplo (máquina hipotética)

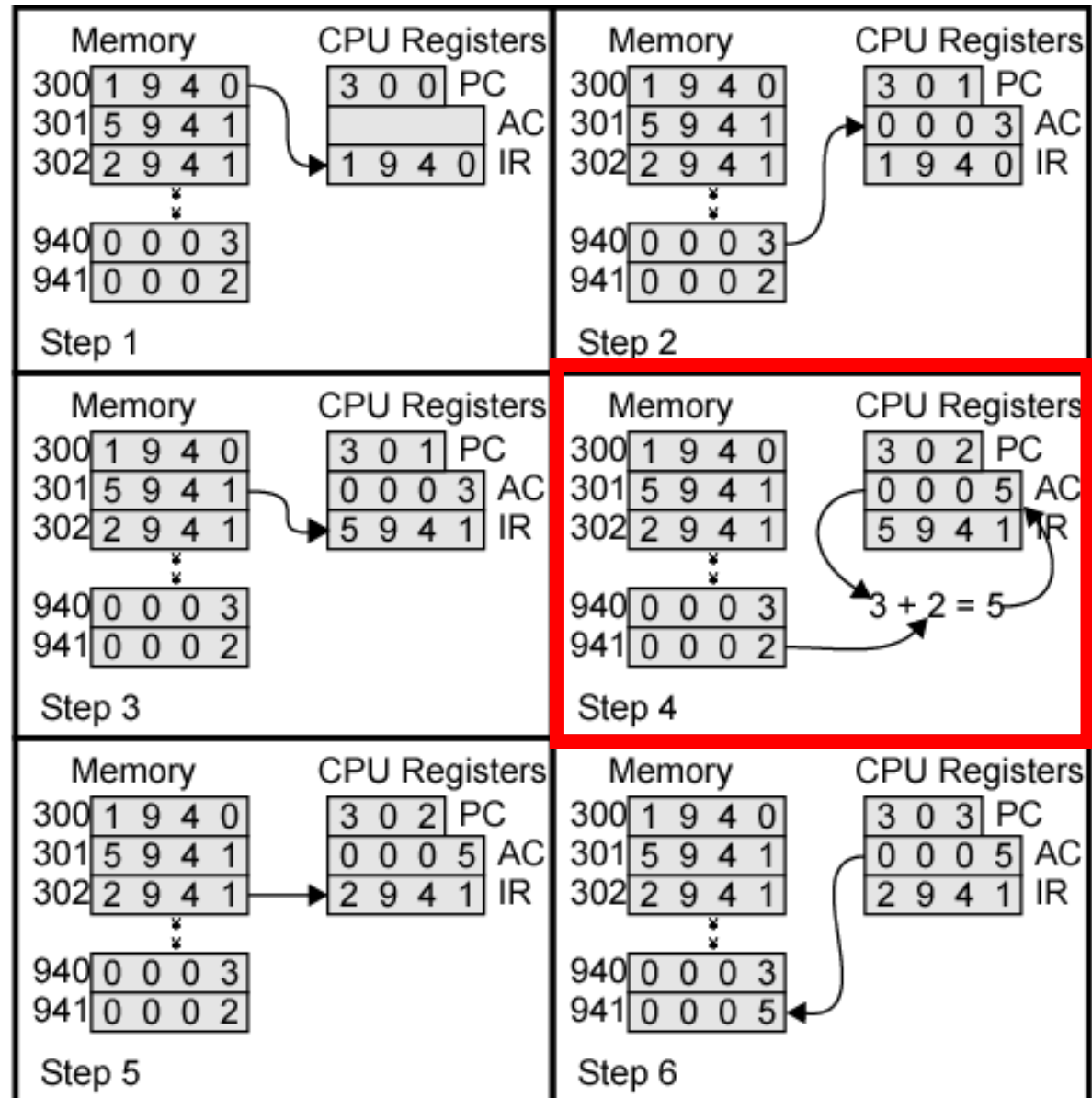
PASO 4:

El primer dígito del IR, $5_{(16)} = 0101_{(2)}$, indica que la instrucción debe sumar el contenido del AC con el de una posición de memoria.

Los siguientes 12 bits indican la dirección de memoria ("941").

El contenido de AC (0003) y el de la posición 941 (0002) se suman y el resultado se almacena en AC (0005).

El PC se incrementa en 1 (pasa a 302).

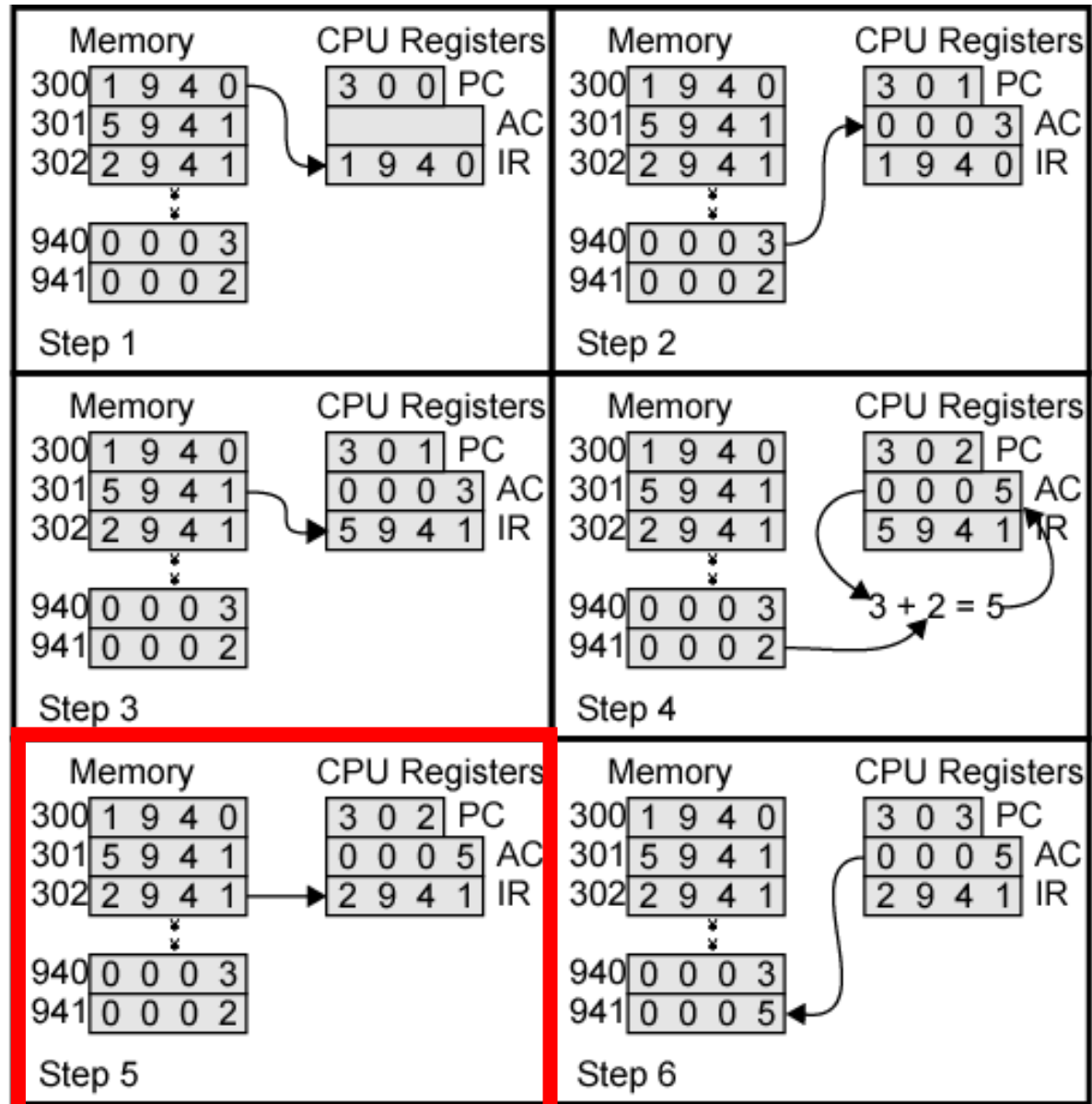


Ciclo de Instrucción

● Ejemplo (máquina hipotética)

PASO 5:

Se capta la siguiente instrucción (2941) desde la nueva posición de memoria, cargándola en IR.



Ciclo de Instrucción

● Ejemplo (máquina hipotética)

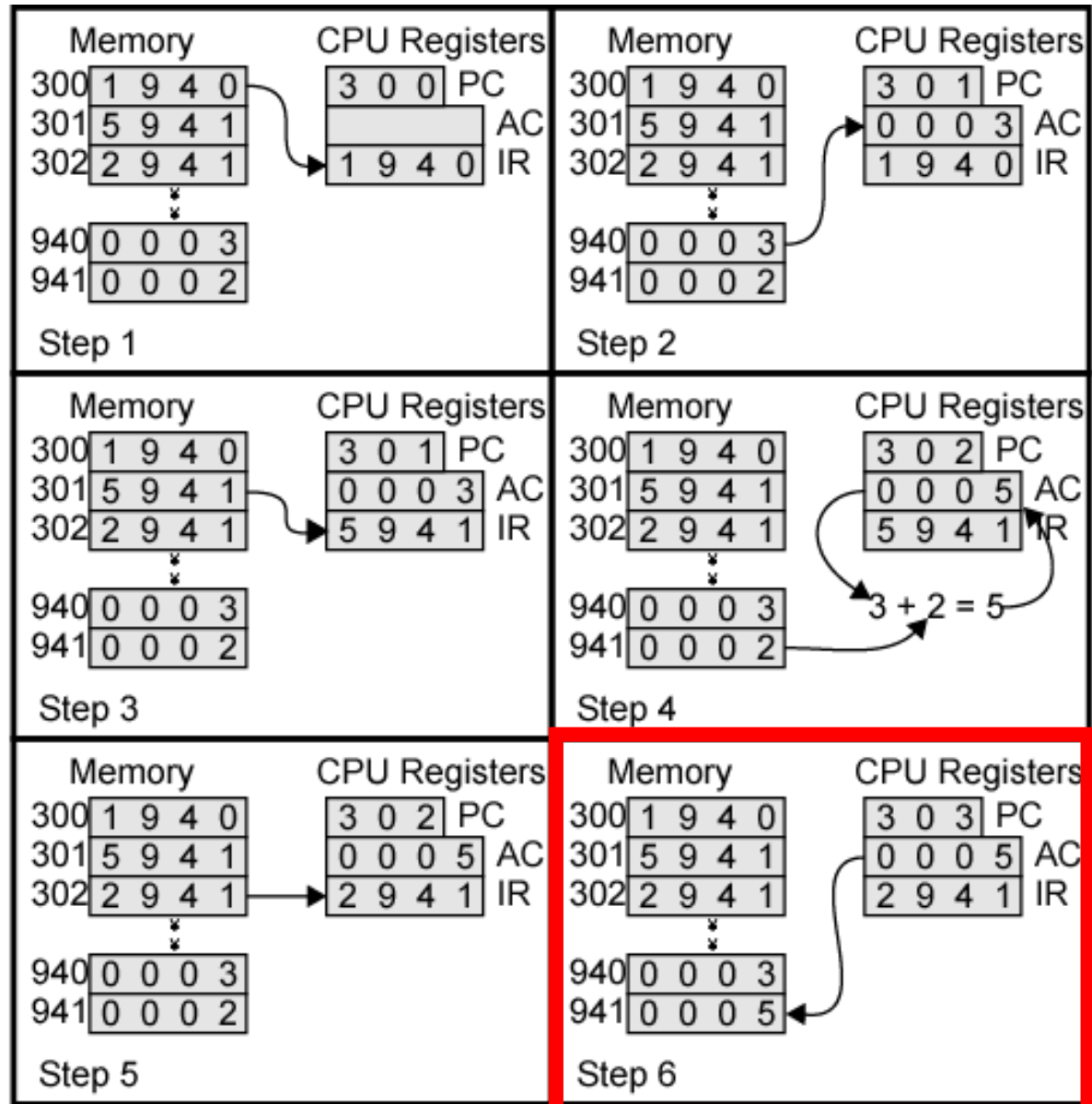
PASO 6:

El primer dígito del IR, $2_{(16)} = 0010_{(2)}$, indica que el AC se debe almacenar en una posición de memoria.

Los siguientes 12 bits indican la dirección de la memoria ("941").

El contenido de AC se almacena en la posición 941.

El registro PC se incrementa en 1 (toma el valor 303).



Arquitectura CISC, explicación básica.

CISC

CISC: Computadoras con un conjunto de instrucciones complejo (complex instruction set computer).

- La microprogramación es una característica importante y esencial de casi todas las arquitecturas CISC.
- Significa que cada instrucción es interpretada por un microprograma localizado en una memoria de control dentro del procesador.
- Para esto se requieren de varios ciclos de reloj (al menos uno por microinstrucción).

CISC

Instrucciones multiciclo

Carga/almacenamiento incorporadas en otras instrucciones

Arquitectura memoria-memoria

Instrucciones largas,
Código con menos líneas

Utiliza memoria de microprograma

Se enfatiza la versatilidad del repertorio de instrucciones

Reduce la dificultad de implementar compiladores