

El Procesador IV:

Modo de ejecución de las instrucciones

Componentes básicos de un computador

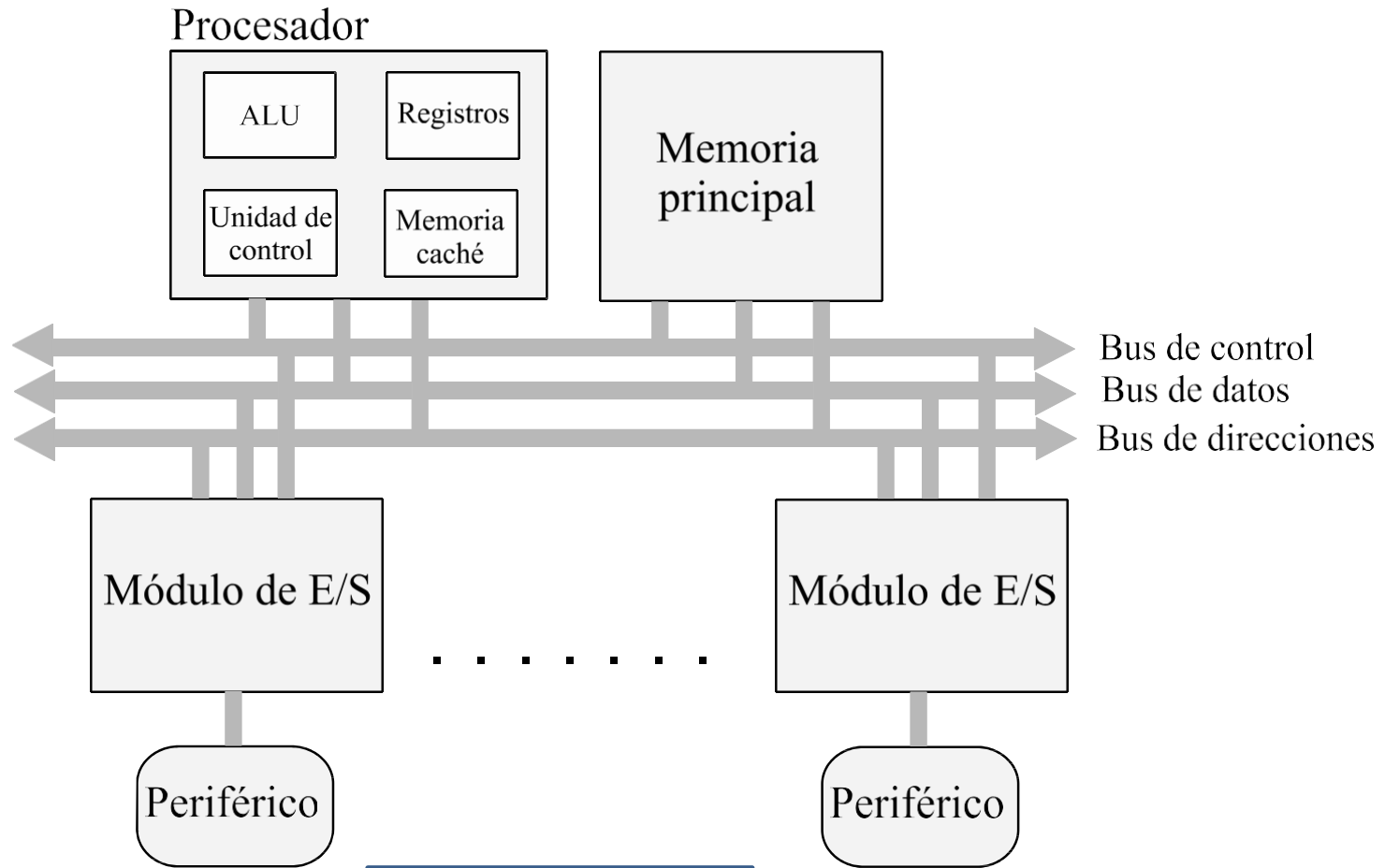
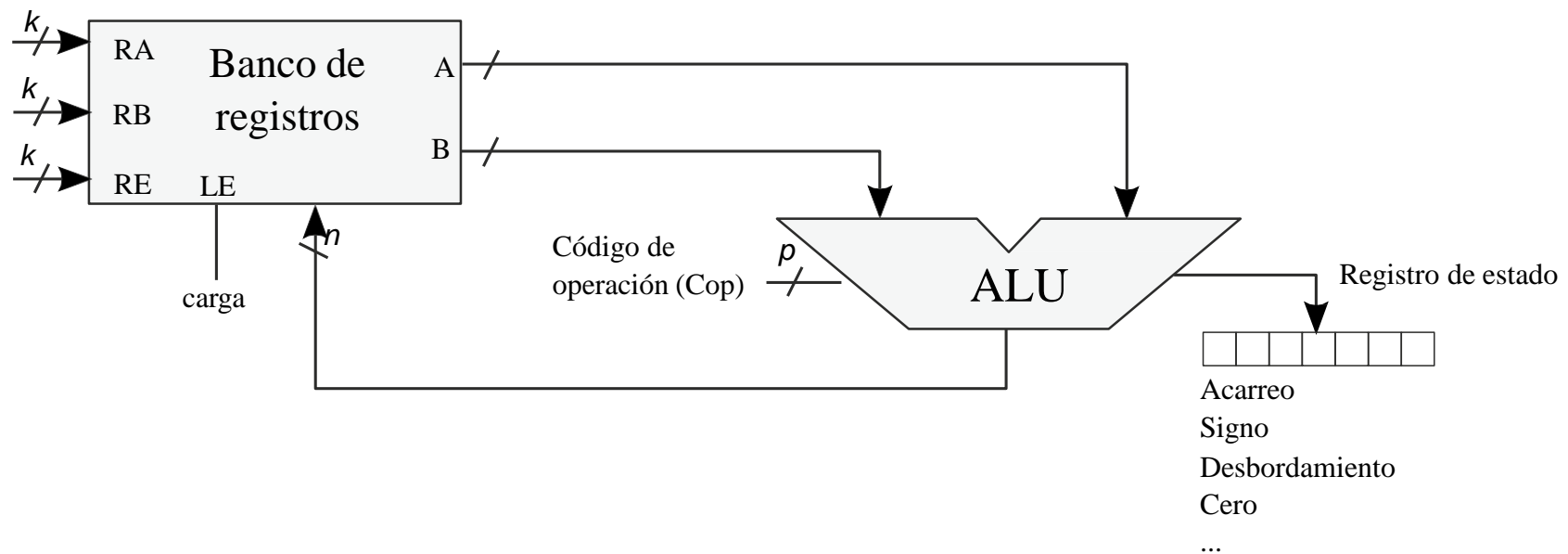


Figura 1



Datapath

Figura 2

Buffer triestado y tabla de funcionamiento

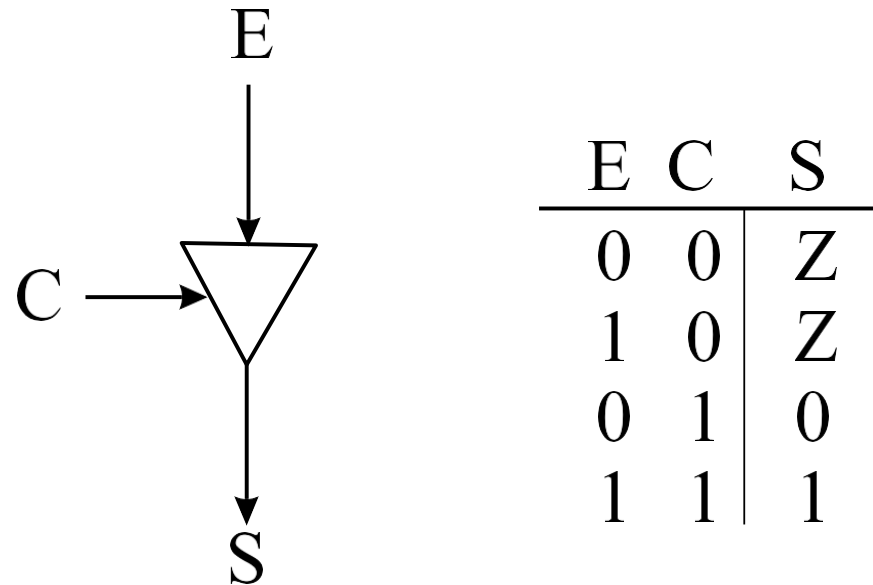
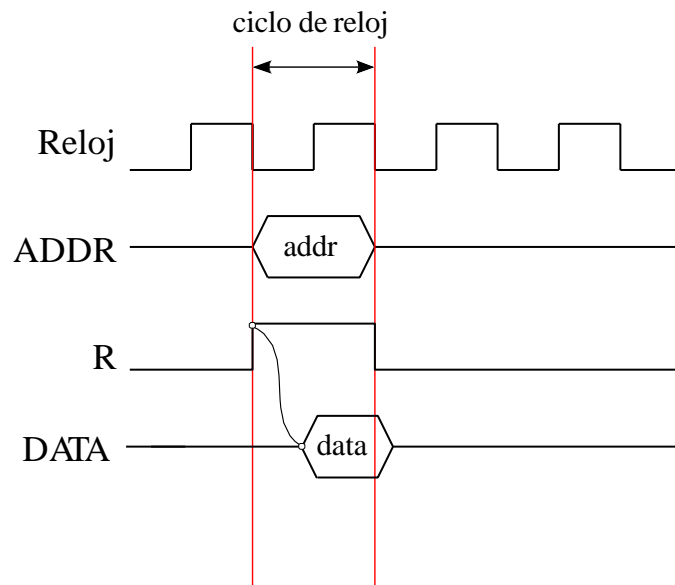
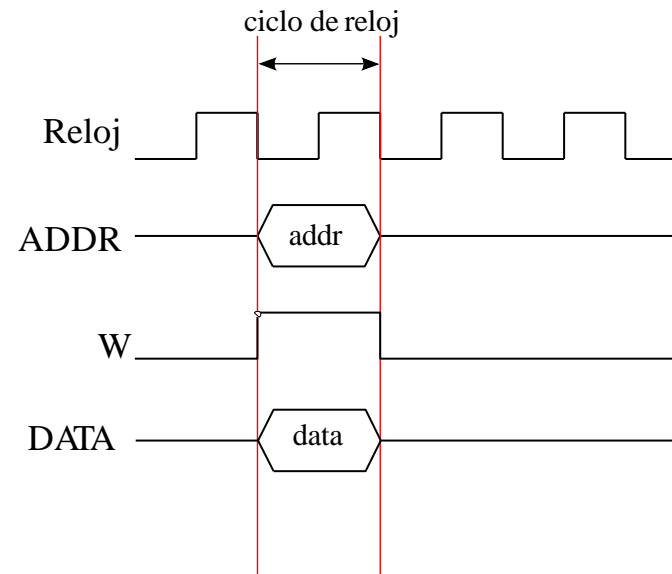


Figura 3

Cronograma de acceso a la memoria del procesador



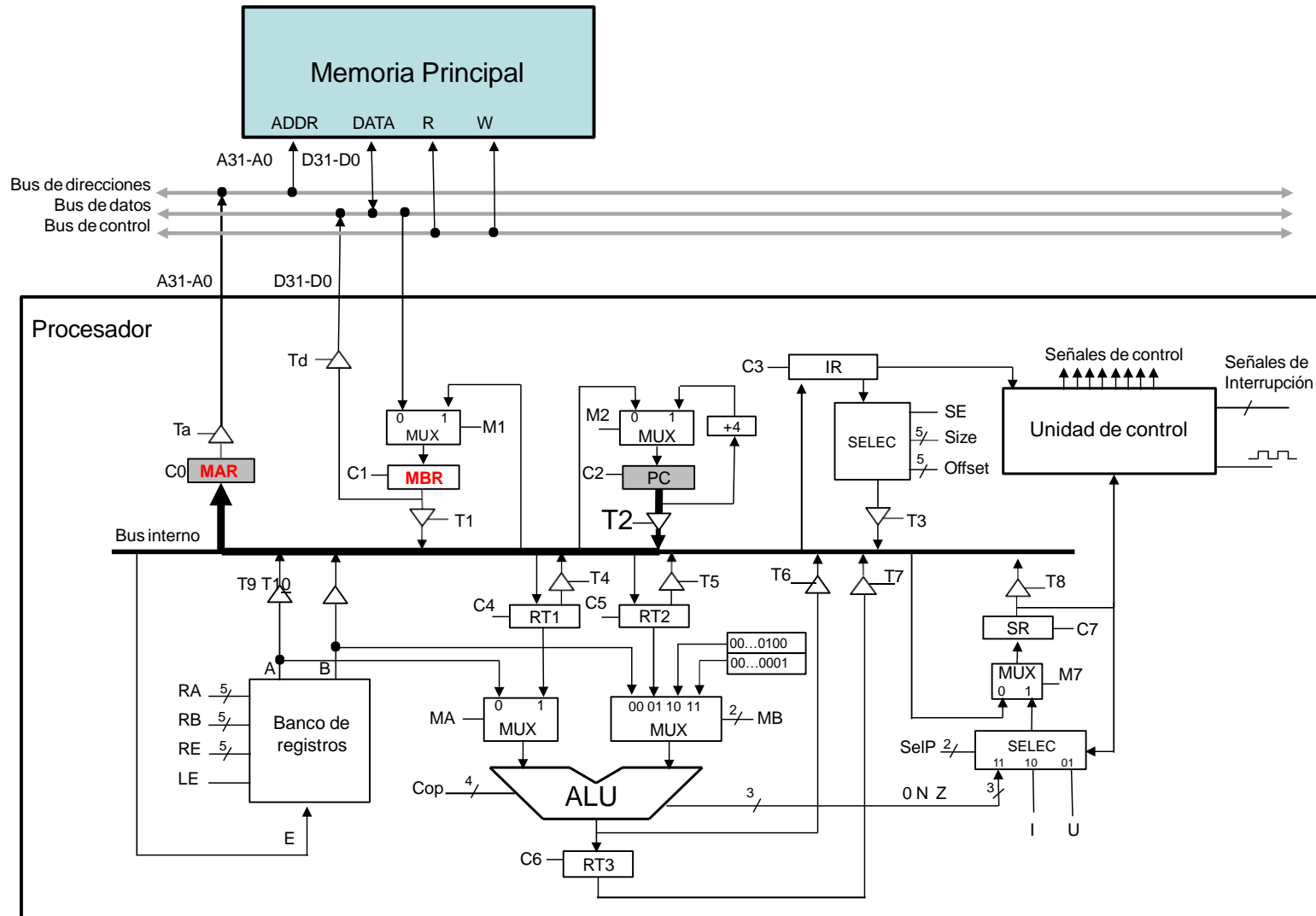
Operación de lectura



Operación de escritura

Figura 4

Camino de datos para la ejecución de la operación elemental $MAR \leftarrow PC$



Señales de control a activar para la ejecución de la operación elemental

$MAR \leftarrow PC$. El resto de señales permanecen inactivas

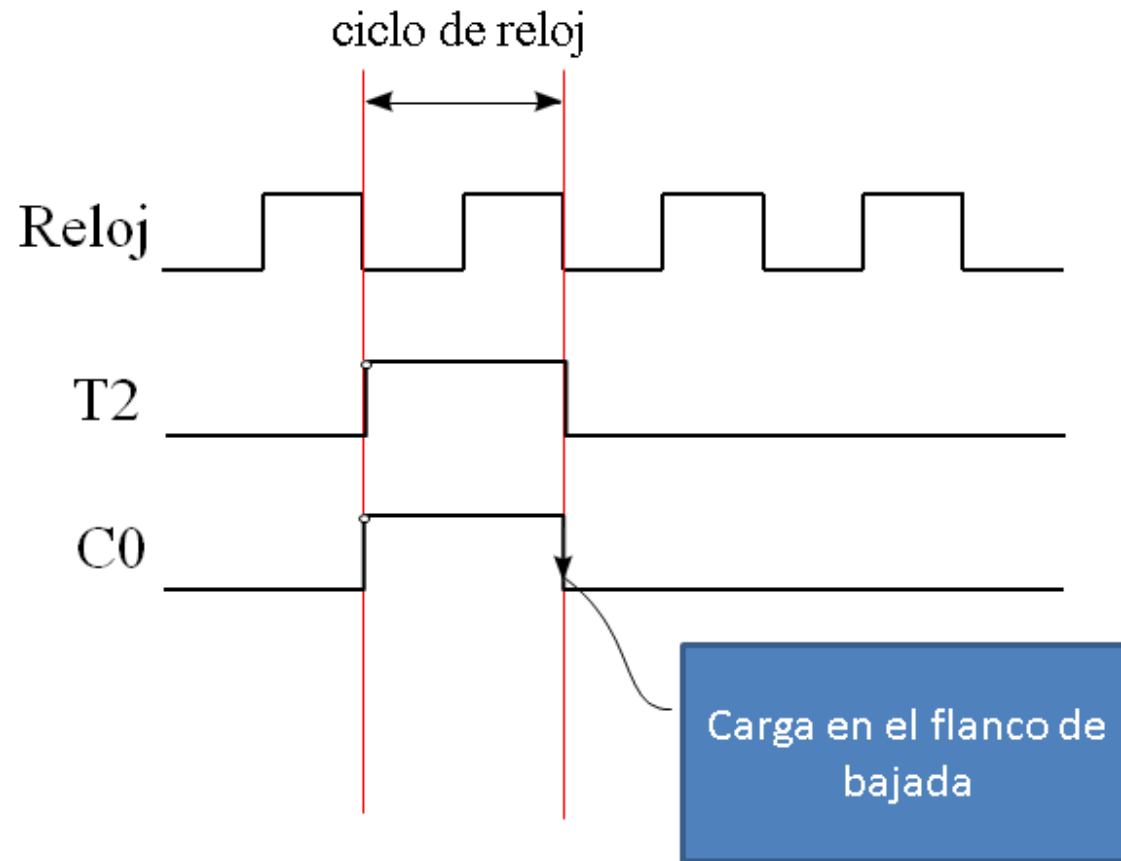


Figura 6

Secuencia de operaciones elementales para la fase de búsqueda de las instrucciones

Ciclo	Operaciones elementales	Señales de control
c1	$MAR \leftarrow PC$	$T2, C0$
c2	$PC \leftarrow PC + 4$ $MBR \leftarrow MP$	$M2, C2$ $Ta, R, C1$
c3	$IR \leftarrow MBR$	$T1, C3$

Figura 7

Ambas operaciones se realizan en paralelo en un mismo ciclo de reloj

Interrupciones

Una interrupción es **síncrona** cuando ocurre como consecuencia directa de la ejecución de la instrucción máquina que se está ejecutando. En caso contrario se denomina **asíncrona**. Las interrupciones se pueden clasificar de la siguiente forma:

Excepciones hardware síncronas:

Se generan como consecuencia de la instrucción en curso. Ocurren cuando existe un problema en la ejecución de la instrucción como, por ejemplo, una división por cero, un desbordamiento de la pila, códigos de operación máquina inválidos , etc.

Excepciones hardware asíncronas:

Son interrupciones asíncronas producidas por un error en el hardware como, por ejemplo, un error de paridad en la memoria.

Interrupciones externas:

Se trata de interrupciones asíncronas producidas por elementos externos conectados al procesador, como son los controladores de dispositivos de E/S.

Ciclo de instrucción con tratamiento de interrupciones

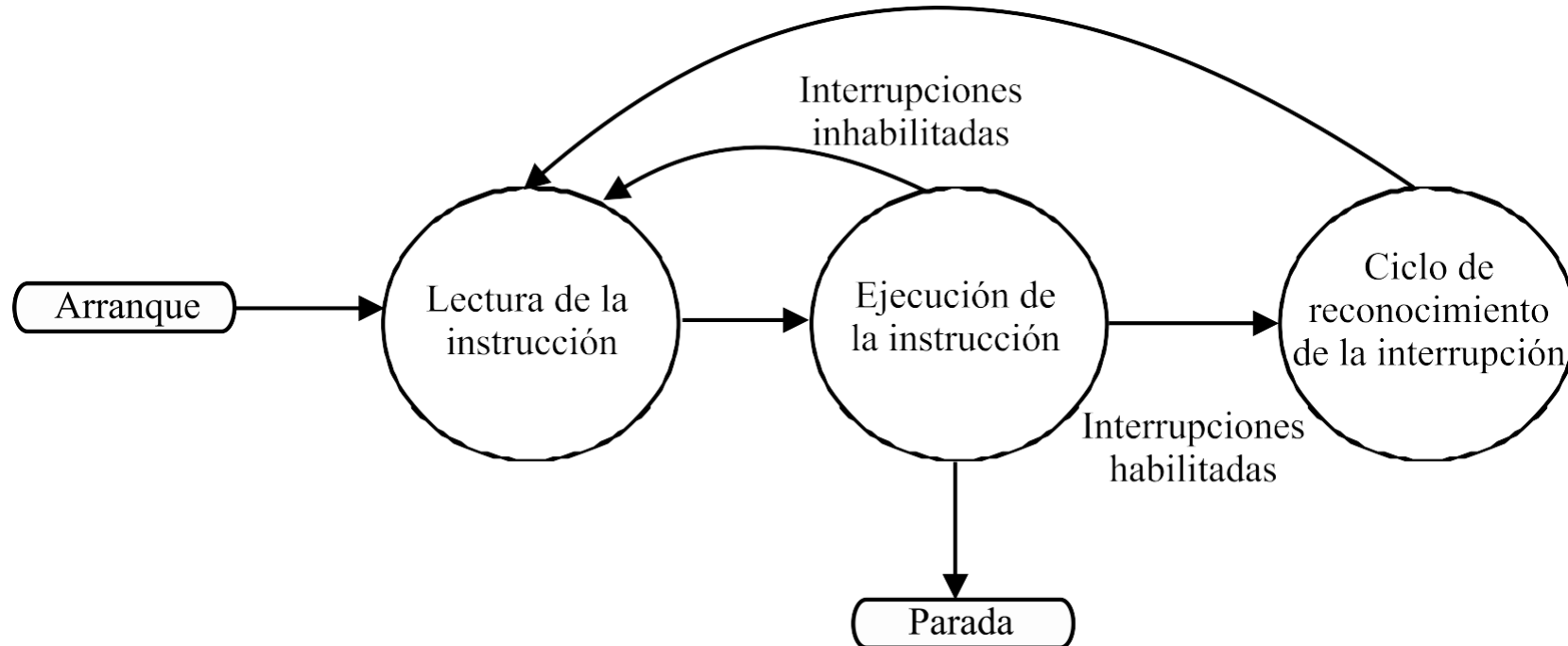


Figura 8

Entradas y salidas de la unidad de control

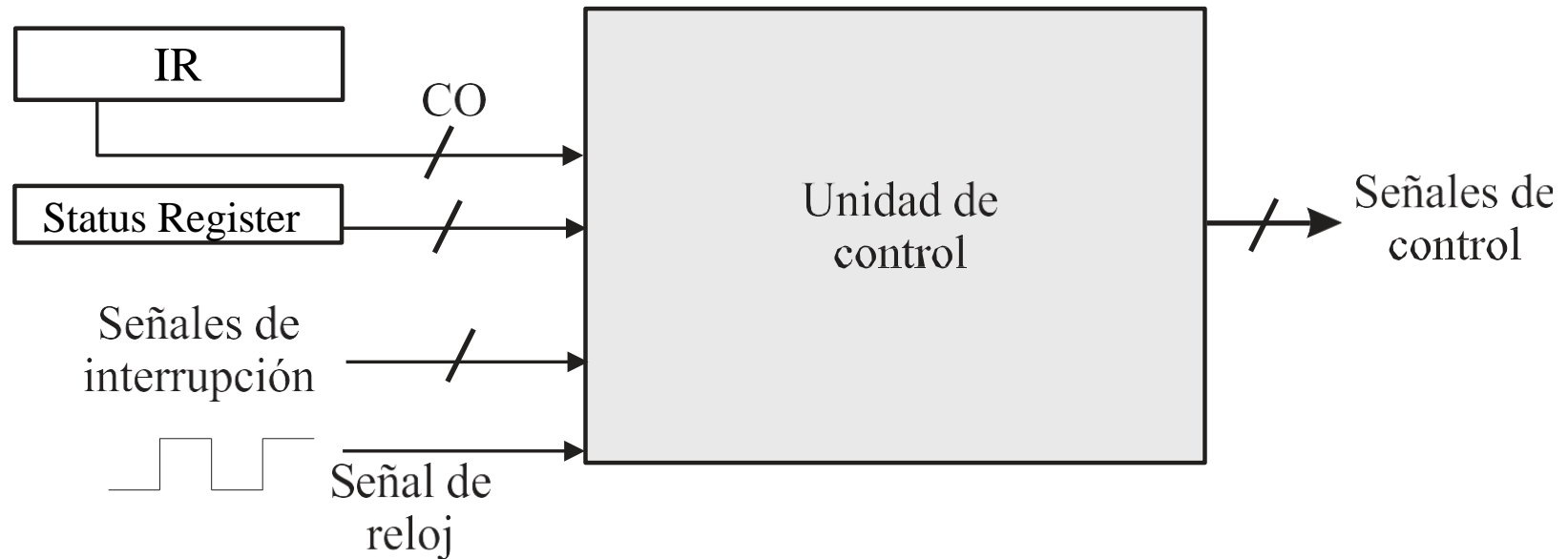


Figura 9

Esquema del procesador - Ejemplo

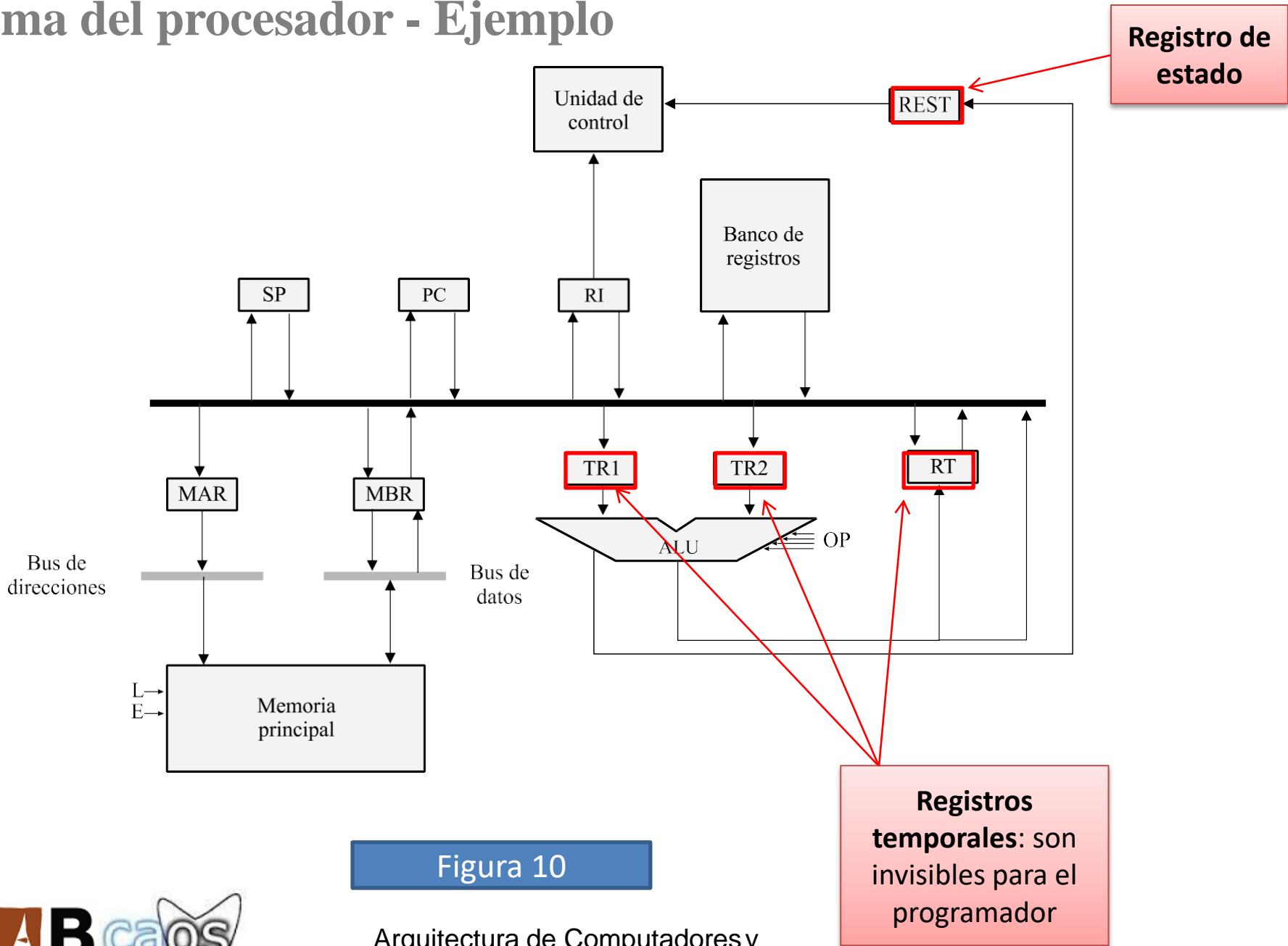


Figura 10

Juego de instrucciones básico

Instrucción	Descripción
LOAD Rk, dirección	Carga en Rk el contenido almacenado en dirección.
LOAD Rk, (Rj)	Carga en Rk el contenido de la dirección almacenada en Rj.
STORE Rk, dirección	Almacena en dirección el contenido de Rk.
STORE Rk, (Rj)	Almacena en la dirección almacenada en Rj el contenido de Rk.
MOVE Ri, Rj	Transfiere el contenido de Rj a Ri.
ADD Ri, Rj	Suma Ri y Rj y deja el resultado en Ri.
SUB Ri, Rj	Resta Ri y Rj deja el resultado en Ri.
INC Ri	Suma uno al contenido del registro Ri.
DEC Ri	Resta uno al contenido del registro Ri.
AND Ri, Rj	Operación lógica AND entre Ri y Rj, el resultado se deja en Ri.
OR Ri, Rj	Operación lógica OR entre Ri y Rj, el resultado se deja en Ri.
JMP dirección	Salto incondicional a dirección.
JMP (Rk)	Salto incondicional a la dirección almacenada en Rk.
CALL dirección	Llamada a subrutina.
RET	Instrucción complementaria a CALL, indica el fin. de subrutina restituyendo la dirección del programa llamante.
BGT Rk, Rj, etiqueta	Si $Rk > Rj$ salta a etiqueta.
BGE Rk, Rj, etiqueta	Si $Rk \geq Rj$ salta a etiqueta.
BLT Rk, Rj, etiqueta	Si $Rk < Rj$ salta a etiqueta.
BLE Rk, Rj, etiqueta	Si $Rk \leq Rj$ salta a etiqueta.
BE Rk, Rj, etiqueta	Si $Rk = Rj$ salta a etiqueta.
BNE Rk, Rj, etiqueta	Si $Rk \neq Rj$ salta a etiqueta.

Ejercicio 1

La Figura 10 muestra la estructura de un computador con un procesador de 32 bits y un banco de registros de 16 registros generales (R1 a R16).

Este computador dispone del juego de instrucciones que se muestra en la Tabla 1.

Indicar las operaciones elementales correspondientes a las siguientes instrucciones:

a) Instrucción **STORE, Rk, dirección**

b) Instrucción **JMP (Rk)**



Ejercicio 1 – Solución a)

a) Instrucción **STORE, Rk, dirección**

A continuación se indican los ciclos y las operaciones elementales a ejecutar para la instrucción:

C1: $MAR \leftarrow PC$
C2: $MBR \leftarrow MP$
 $PC \leftarrow PC + 1$
C3: $RI \leftarrow MBR$
C4: Decodificación
C5: $MBR \leftarrow Rk$
C6: $MAR \leftarrow RI(\text{dirección})$
C7: $MP \leftarrow MBR$

Ejercicio 1 – Solución b)

b) Instrucción **JMP (Rk)**

Las operaciones elementales correspondientes a la instrucción JMP (Rk) se muestran a continuación:

C1: $MAR \leftarrow PC$
C2: $MBR \leftarrow MP$
 $PC \leftarrow PC + 1$
C3: $RI \leftarrow MBR$
C4: Decodificación
C5: $PC \leftarrow Rk$

Ejercicio 2



La Figura 10 presenta el esquema simplificado de un procesador de 32 bits. Sobre esta arquitectura se quiere analizar el impacto que tienen, en cuanto a rendimiento, diferentes juegos de instrucciones sobre la siguiente sentencia de un lenguaje de alto nivel:

$$A = B + C$$

Considerando que los valores correspondientes a las variables A, B y C residen en memoria y que las tres variables corresponden a datos de tipo entero, se quiere analizar dos tipos de modelos de ejecución:

1. Modelo de ejecución Registro - Registro
2. Modelo de ejecución Memoria - Memoria

Se pide:

Para cada uno de los modelos de ejecución anterior, indicar las instrucciones en ensamblador que son necesarias para ejecutar la sentencia de alto nivel $A = B + C$

Ejercicio 2- Solución

Modelo registro – registro:

Los operandos residen en los registros. Para transferir la información entre los registros y la memoria principal se necesitan instrucciones de tipo LOAD y STORE que tienen una dirección de memoria

LOAD R1, B	Carga en el registro R1, el dato almacenado en la dirección dada por B
LOAD R2, C	Carga en el registro R2, el dato almacenado en la dirección dada por C
ADD R1, R2	Suma el contenido de R1 y R2 y deja el resultado en R1
STORE R1, A	Almacena el contenido de R1 en la dirección de memoria A

Ejercicio 2- Solución (continuación)

Modelo memoria– memoria:

Tanto los datos como el resultado residen en memoria principal.

MOVE A, B
ADD A, C

Copia el dato almacenado en la dirección B en la dirección A
Suma el dato almacenado en C con el dato que se ha almacenado en A
y deja el resultado de nuevo en la dirección A