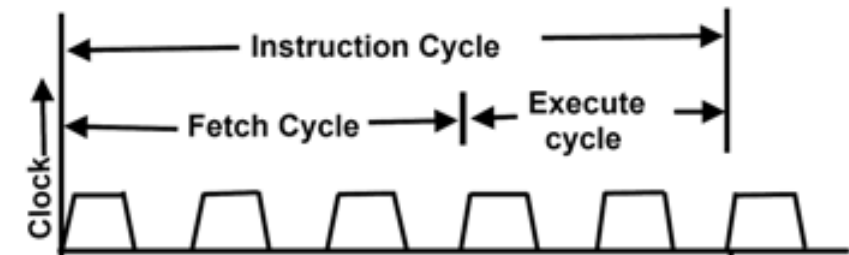
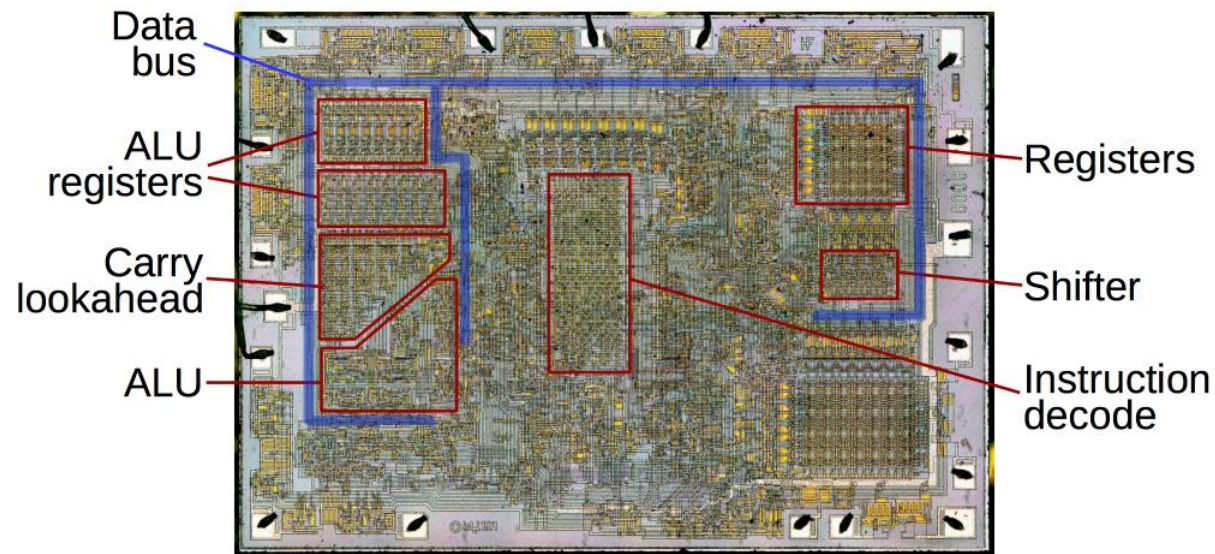
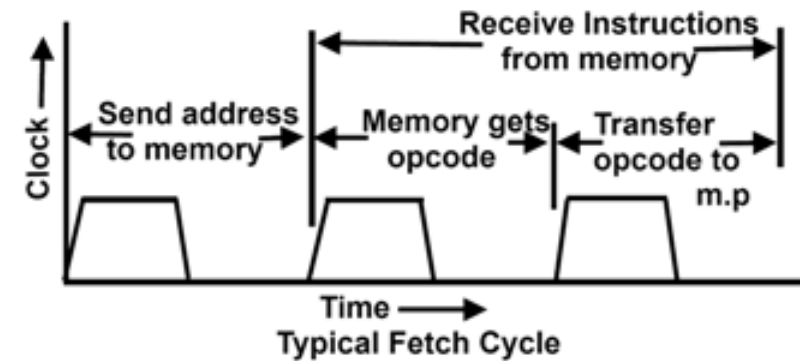


La Unidad de Control



Instruction cycle showing FC, EC and IC



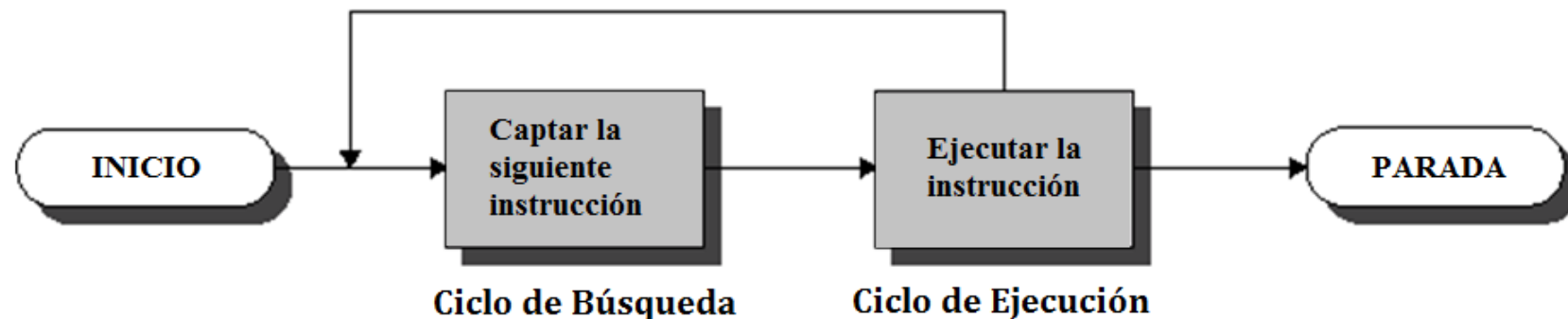
Ciclo de instrucción (repaso)

Para la ejecución de un programa, la CPU realiza una secuencia regular, que se repite hasta finalizar el mismo, denominada *ciclo de instrucción*:

- Traer la instrucción desde la memoria
- Realizar el procesamiento correspondiente a la instrucción específica.

Esto permite subdividir cada instrucción en al menos dos partes básicas:

1. Un ciclo de *búsqueda o captación (fetch)*.
2. Un ciclo de *ejecución*.



Interrupciones

Como se vio hasta aquí, un programa consiste en la ejecución secuencial de sus instrucciones. Sin embargo, existe un mecanismo que permite alterar esa ejecución, denominado ***interrupción***.

Las interrupciones pueden originarse en varias fuentes:

- Como resultado de una ejecución de una instrucción
Ejemplo: desbordamiento aritmético (“*overflow*”), división por cero
- Por un temporizador interno del procesador.
Ejemplo: funciones regulares realizadas por el Sistema Operativo.
- Por una operación de E/S.
Ejemplo: para indicar la finalización normal de una operación en un dispositivo.
- Por un fallo de hardware.
Ejemplo: error de paridad en la memoria, pérdida de energía.

Interrupciones

Además de las mencionadas, muchos procesadores tienen instrucciones explícitas que afectan al procesador de la misma manera que las interrupciones por hardware, generalmente usadas para hacer llamadas a funciones del Sistema Operativo.

Salvo las de este tipo, las interrupciones son eventos asincrónicos a los programas que ejecuta la CPU. Su atención implica dejar de ejecutar estos para pasar a ejecutar otros procesos que atiendan las causas que las originan. Esto puede causar demoras inaceptables en los programas.

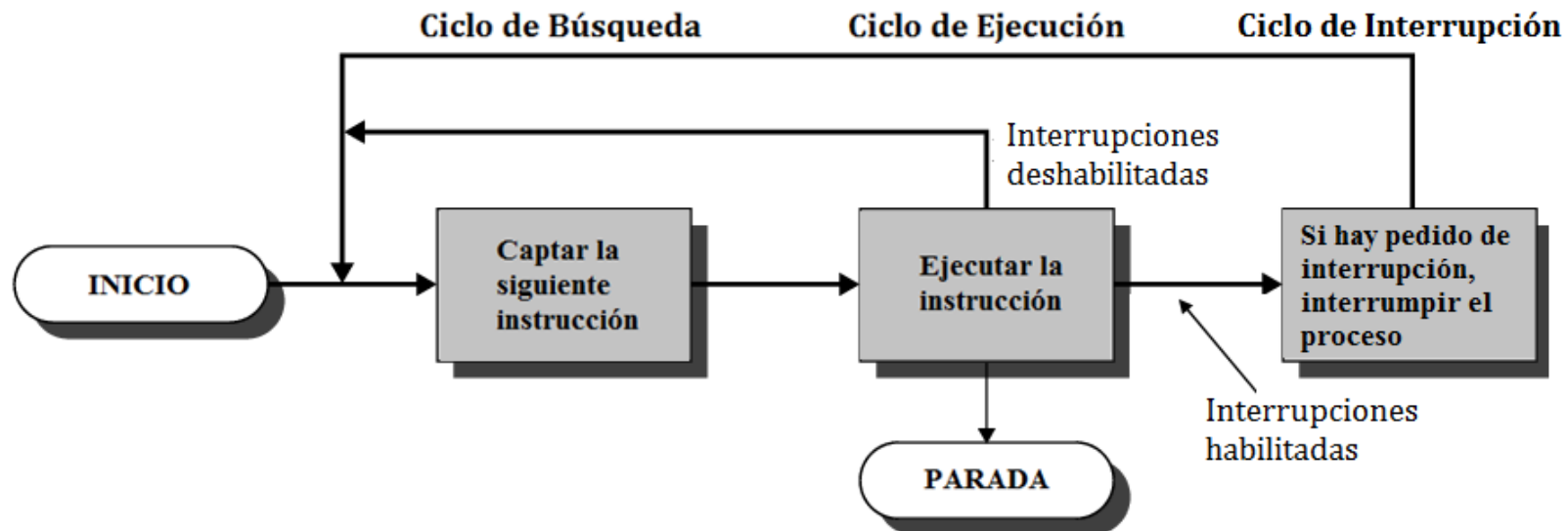
Por lo tanto, en sistemas con múltiples fuentes de interrupciones, estas deben ser jerarquizadas. Un primer paso para esto es separarlas en:

- Enmascarables: pueden ser ignoradas. Esto se logra mediante instrucciones específicas.
- No enmascarables: las que no pueden ignorarse, por indicar eventos peligrosos o de alta prioridad.

Ciclo de instrucción con interrupciones

Es necesario modificar el ciclo de instrucción básico, añadiendo un *ciclo de interrupción* a continuación del de ejecución:

- Se comprueba si se ha solicitado alguna interrupción, indicada por una señal (*flag*) de pedido de interrupción.
- Si no existe un flag de interrupción activo, o el pedido corresponde a una interrupción enmascarada, se capta la siguiente instrucción.
- Si hay alguna interrupción habilitada, se pasa a un programa llamado *gestor de interrupción*.



Gestor de interrupciones

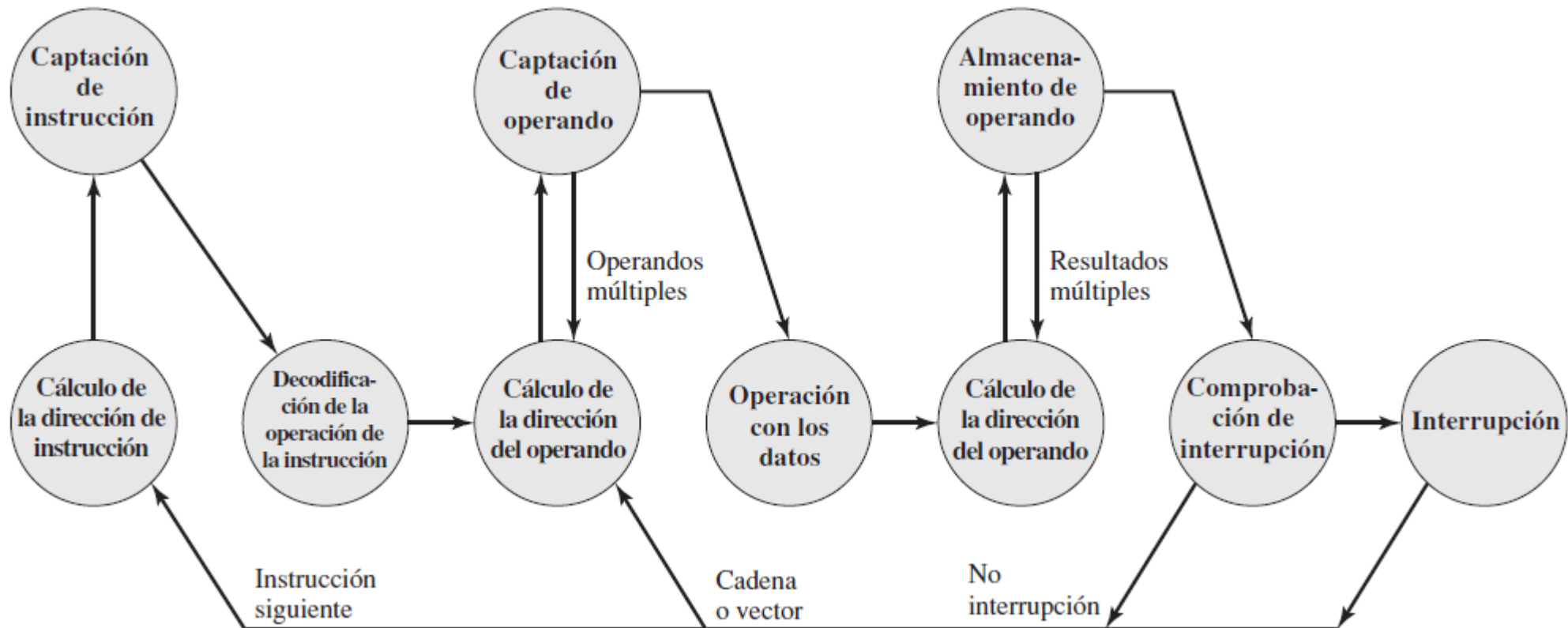
Para la atención de la interrupción, el procesador debe realizar las siguientes acciones:

- Se suspende la ejecución del programa en curso.
- Se guarda el contexto (próxima instrucción a ejecutar y el estado del procesador).
- Se carga el Contador de Programa con la dirección de comienzo de una rutina de gestión de interrupción. Se inhiben otras interrupciones.
- Finalizada la rutina de gestión, el procesador retoma la ejecución del programa del usuario en el punto de interrupción.

Este proceso es similar al explicado para la ejecución de las subrutinas, con la diferencia de que, además de guardarse en la pila el contenido del Contador de Programa, se guardan (y restauran), los contenidos de los registros y los flags del procesador con los contenidos que poseen al momento de abandonar el cauce del programa.

Ciclo de instrucción con interrupciones

En el ciclo de instrucción presentado como diagrama de estados, el ciclo de interrupción se inserta de la siguiente manera:

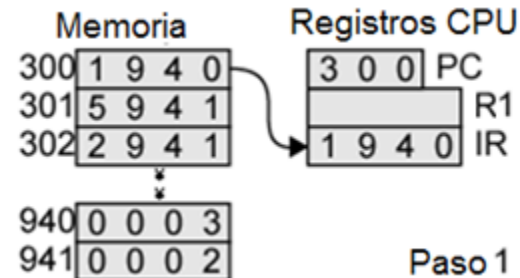


Ejecución de un programa (repaso)

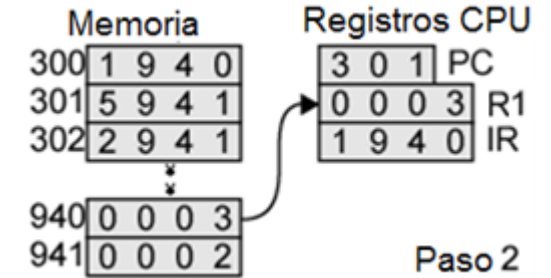
La ejecución del programa vista desde la interacción de la memoria con los registros es:

- Instrucción 1
- Instrucción 2
- Instrucción 3

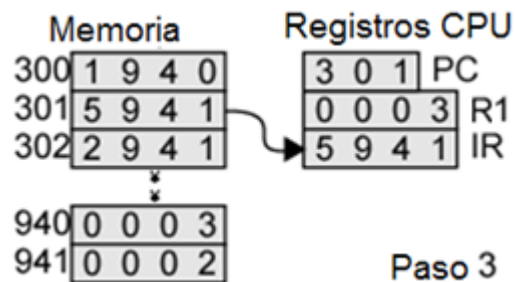
Búsqueda de la instrucción en la memoria:



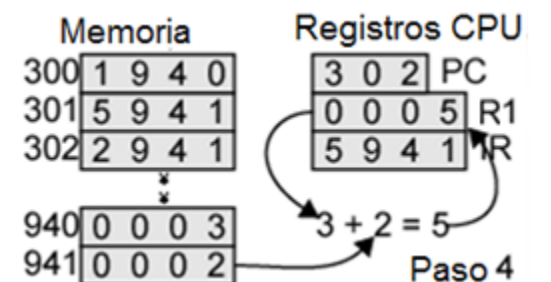
Ejecución de la instrucción en la CPU:



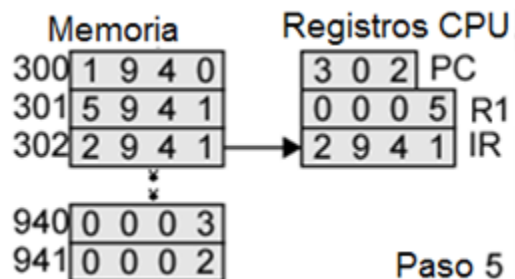
Búsqueda en la memoria:



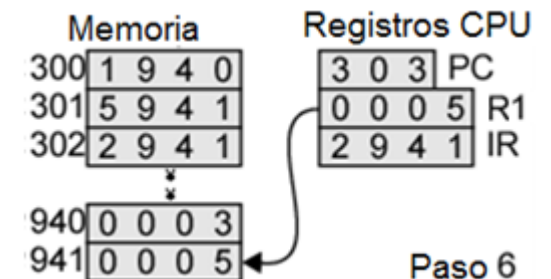
Ejecución en la CPU:



Búsqueda en la memoria:



Ejecución en la CPU:



Ejecución de un programa (repaso)

Y a nivel de las microoperaciones realizadas:

• Instrucción 1

• Instrucción 2

• Instrucción 3

Búsqueda de la instrucción en la memoria:

$t_1: \text{MAR} \leftarrow \text{PC}$
 $t_2: \text{MBR} \leftarrow \text{M}[\text{MAR}],$
 $\text{PC} \leftarrow \text{PC} + 1$
 $t_3: \text{IR} \leftarrow \text{MBR}$

Búsqueda en la memoria:

$t_1: \text{MAR} \leftarrow \text{PC}$
 $t_2: \text{MBR} \leftarrow \text{M}[\text{MAR}],$
 $\text{PC} \leftarrow \text{PC} + 1$
 $t_3: \text{IR} \leftarrow \text{MBR}$

Búsqueda en la memoria:

$t_1: \text{MAR} \leftarrow \text{PC}$
 $t_2: \text{MBR} \leftarrow \text{M}[\text{MAR}],$
 $\text{PC} \leftarrow \text{PC} + 1$
 $t_3: \text{IR} \leftarrow \text{MBR}$

Ejecución de la instrucción en la CPU:

$t_4: \text{MAR} \leftarrow \text{IR}(\text{dirección})$
 $t_5: \text{MBR} \leftarrow \text{M}[\text{MAR}]$
 $t_6: \text{R1} \leftarrow \text{MBR}$

Ejecución en la CPU:

$t_4: \text{MAR} \leftarrow \text{IR}(\text{dirección})$
 $t_5: \text{MBR} \leftarrow \text{M}[\text{MAR}],$
 $\text{T2} \leftarrow \text{R1}$
 $t_6: \text{T1} \leftarrow \text{MBR}$
 $t_7: \text{B} \leftarrow \text{T1} + \text{T2}$

Ejecución en la CPU:

$t_4: \text{MAR} \leftarrow \text{IR}(\text{dirección})$
 $t_5: \text{MBR} \leftarrow \text{R1}$
 $t_6: \text{M}[\text{MAR}] \leftarrow \text{MBR}$

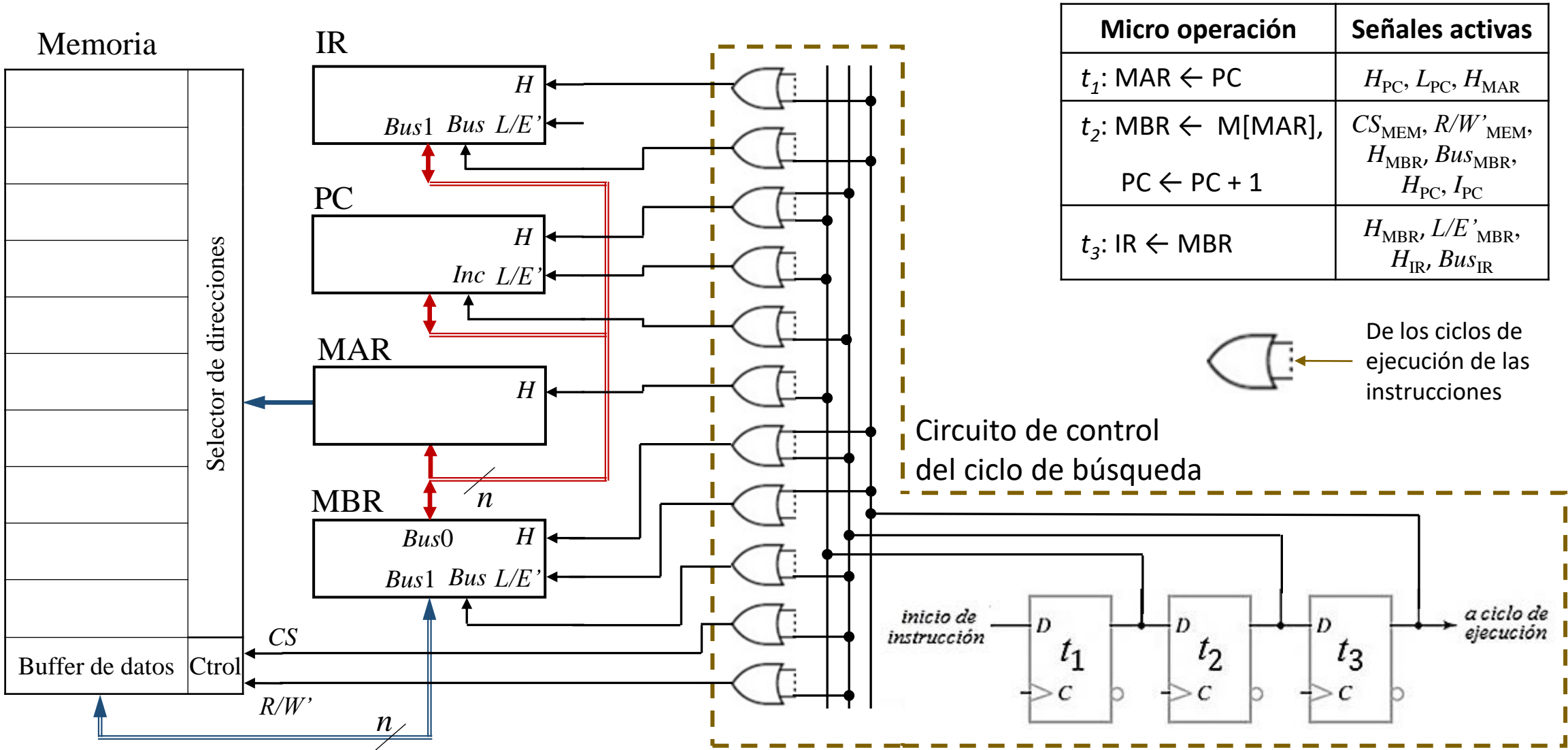
Control del ciclo de búsqueda

Las micro operaciones se materializan a través de la activación de las señales de control de los registros correspondientes. Para el caso del ciclo de búsqueda en el diagrama anterior:

Micro operación	Señales de control activas	Palabra de control (parcial)												
		CS	R/W'	H	L/E'	Bus	H	H	L/E'	Inc	H	L/E'	Bus	...
		MEM		MBR			MAR	PC			IR		...	
t_1 : MAR \leftarrow PC	H_{PC}, L_{PC}, H_{MAR}	0	X	0	X	X	1	1	1	0	0	X	X	...
		MEM		MBR			MAR	PC			IR		...	
t_2 : MBR \leftarrow M[MAR], PC \leftarrow PC + 1	$CS_{MEM}, R/W'_{MEM},$ $H_{MBR}, Bus_{MBR},$ H_{PC}, I_{PC}	1	1	1	0	1	0	1	0	1	0	X	X	...
		MEM		MBR			MAR	PC			IR		...	
t_3 : IR \leftarrow MBR	$H_{MBR}, L/E'_{MBR},$ H_{IR}, Bus_{IR}	0	X	1	1	0	0	0	X	X	1	0	1	...
		MEM		MBR			MAR	PC			IR		...	

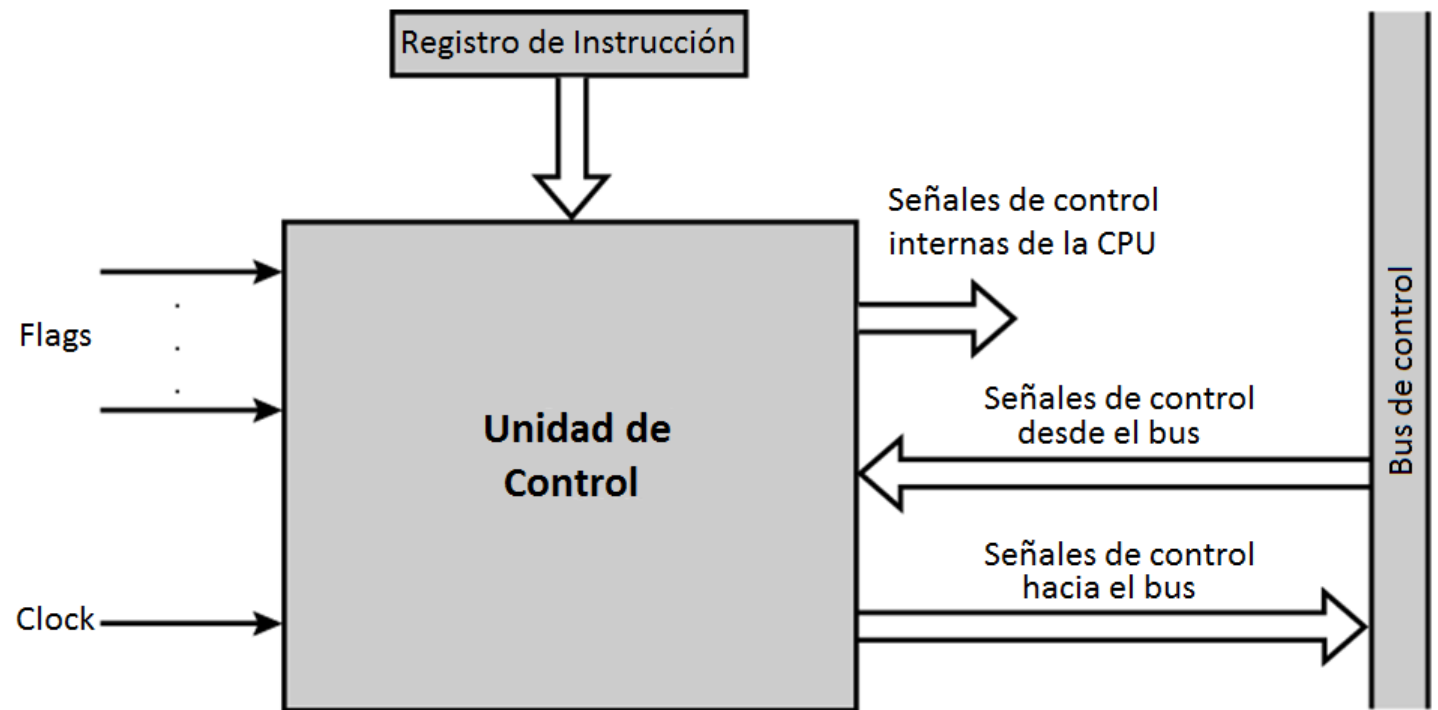
Solo se muestra la parte de la palabra de control correspondiente a los registros que intervienen en el ciclo de búsqueda

Control del ciclo de búsqueda



Control general del ciclo de instrucción

- El circuito de control representado (un secuencial) es la parte correspondiente al **ciclo de búsqueda**, realizada como un registro de desplazamiento (un flip flop por estado).
- Para **cada instrucción** y **cada uno de sus modos de direccionamiento**, resulta necesario diseñar un circuito similar para generar las señales de control de los ciclos de ejecución, y generar las señales de control externo.
- El circuito resultante es la **Unidad de Control**.

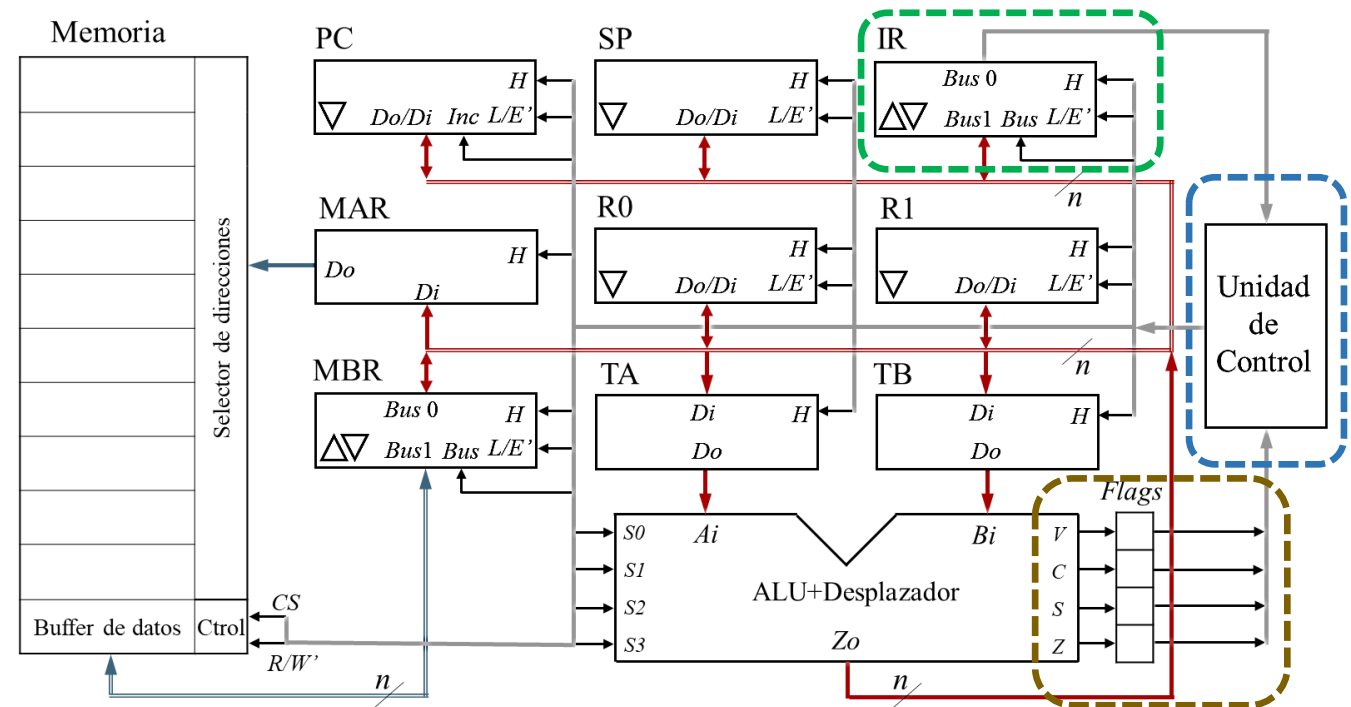


Unidad de Control: Funciones

La Unidad de Control tiene como función el secuenciamiento y la ejecución de las microoperaciones que constituyen el repertorio de instrucciones del procesador.

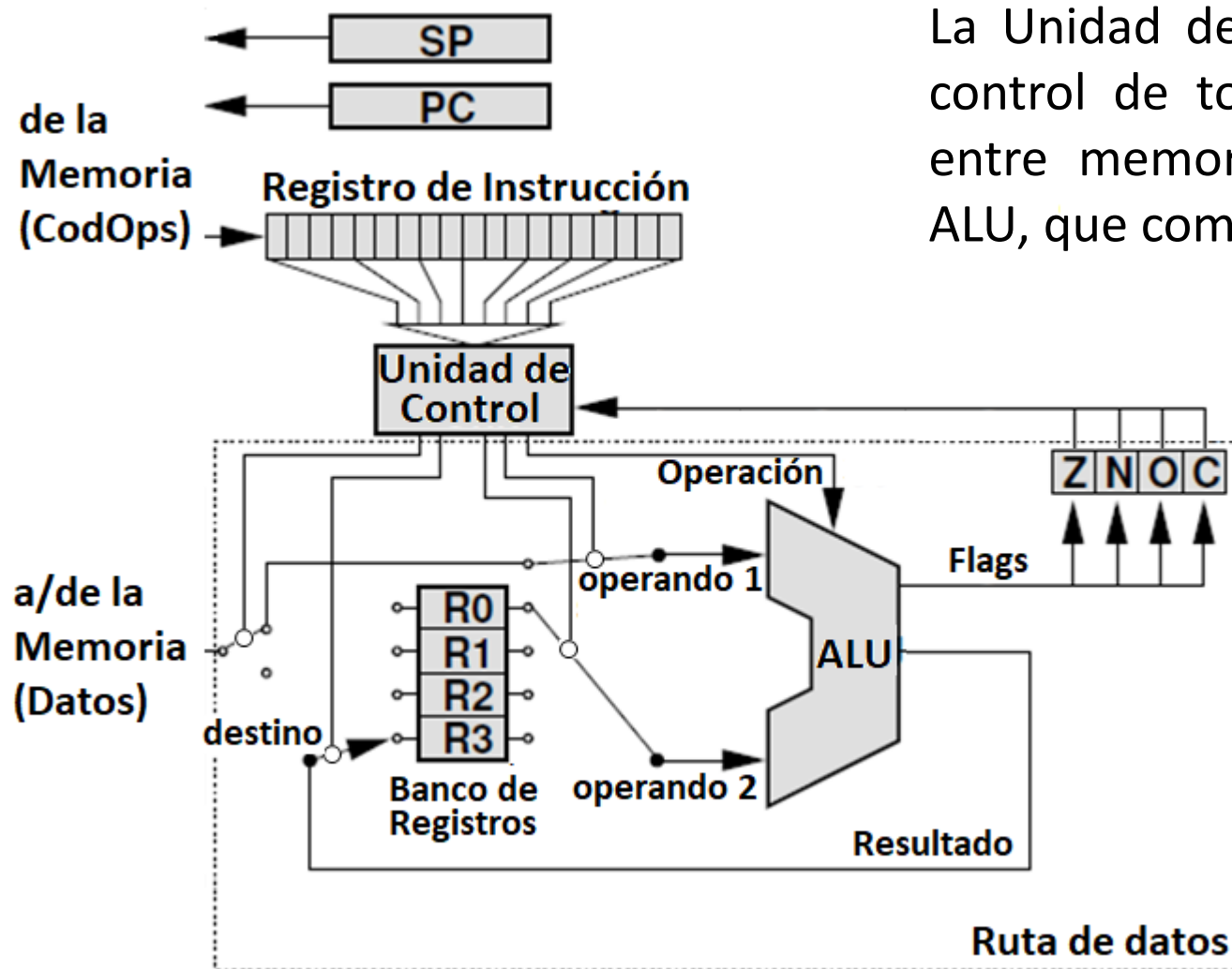
Para ello, recibe como **entradas**:

- Las salidas del Registro de Instrucción (IR);
- Los flags que reciben información de la ALU;
- Las señales de control externas (ej., pedidos de interrupción);
- El clock del sistema.



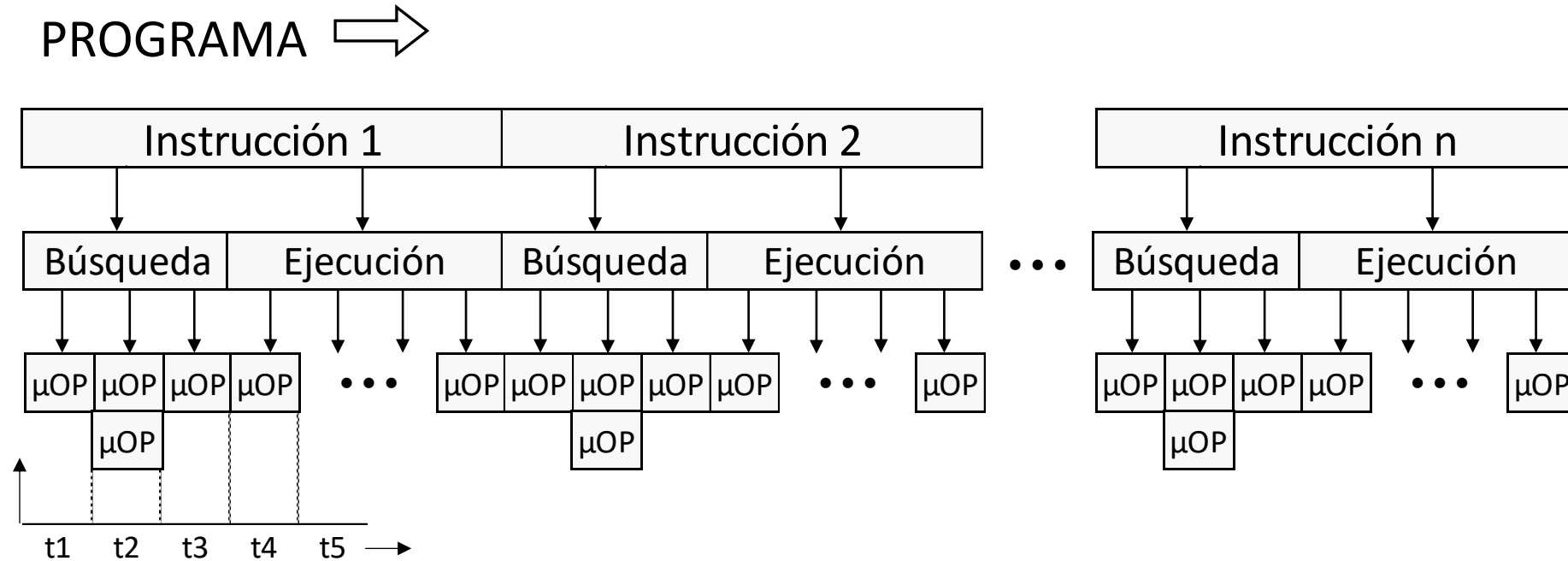
A partir de ellas, genera como **salidas** las señales de habilitación y operación de todos los elementos del sistema.

Funciones de la UC: control de la ruta de datos



La Unidad de Control provee las señales de control de todas las interacciones de datos entre memoria, registros de uso general y ALU, que componen la *ruta de datos*.

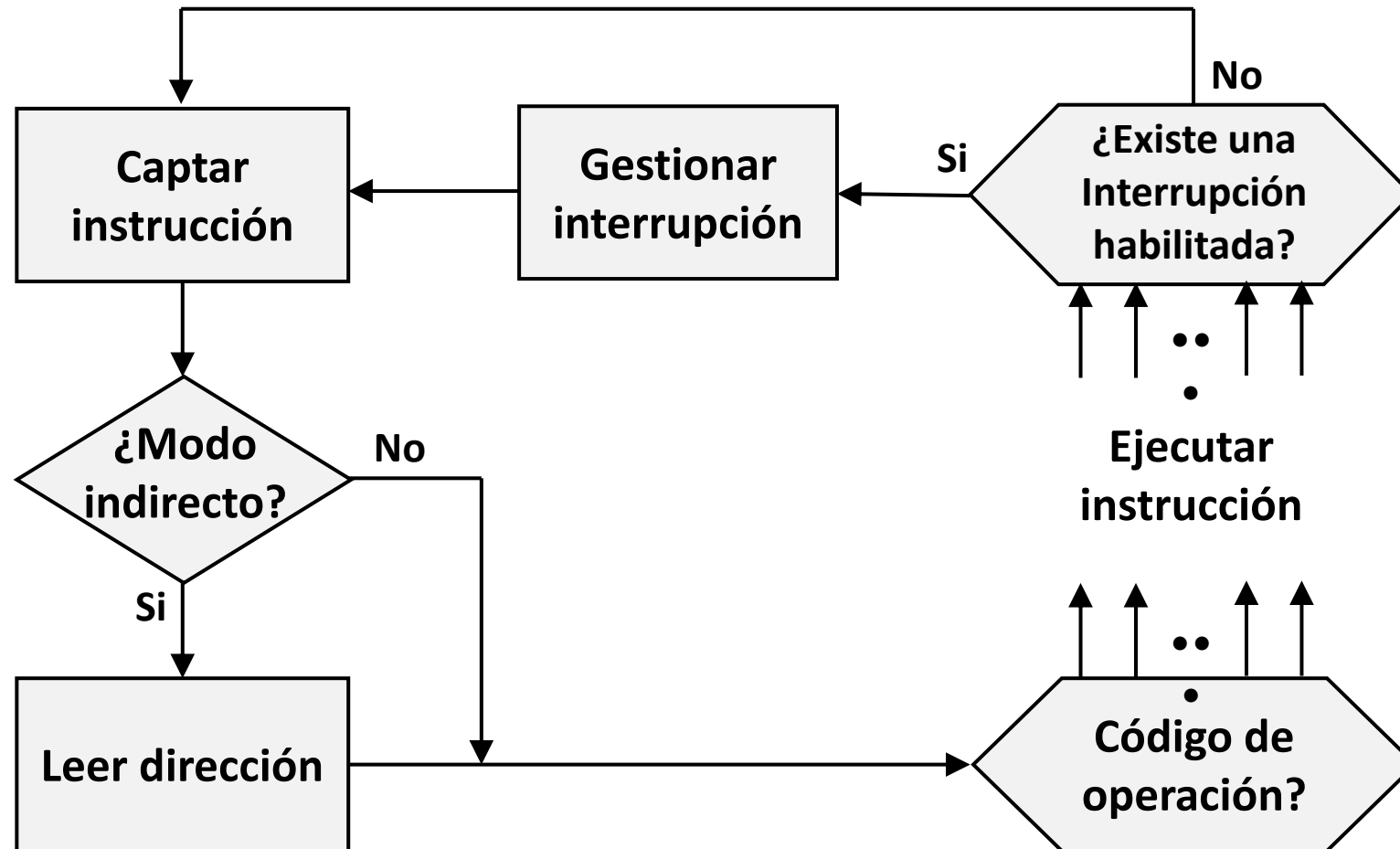
Funciones de la UC: secuenciamiento



La Unidad de Control provee el secuenciamiento de las microoperaciones que constituyen los ciclos de búsqueda y ejecución de cada instrucción del programa.

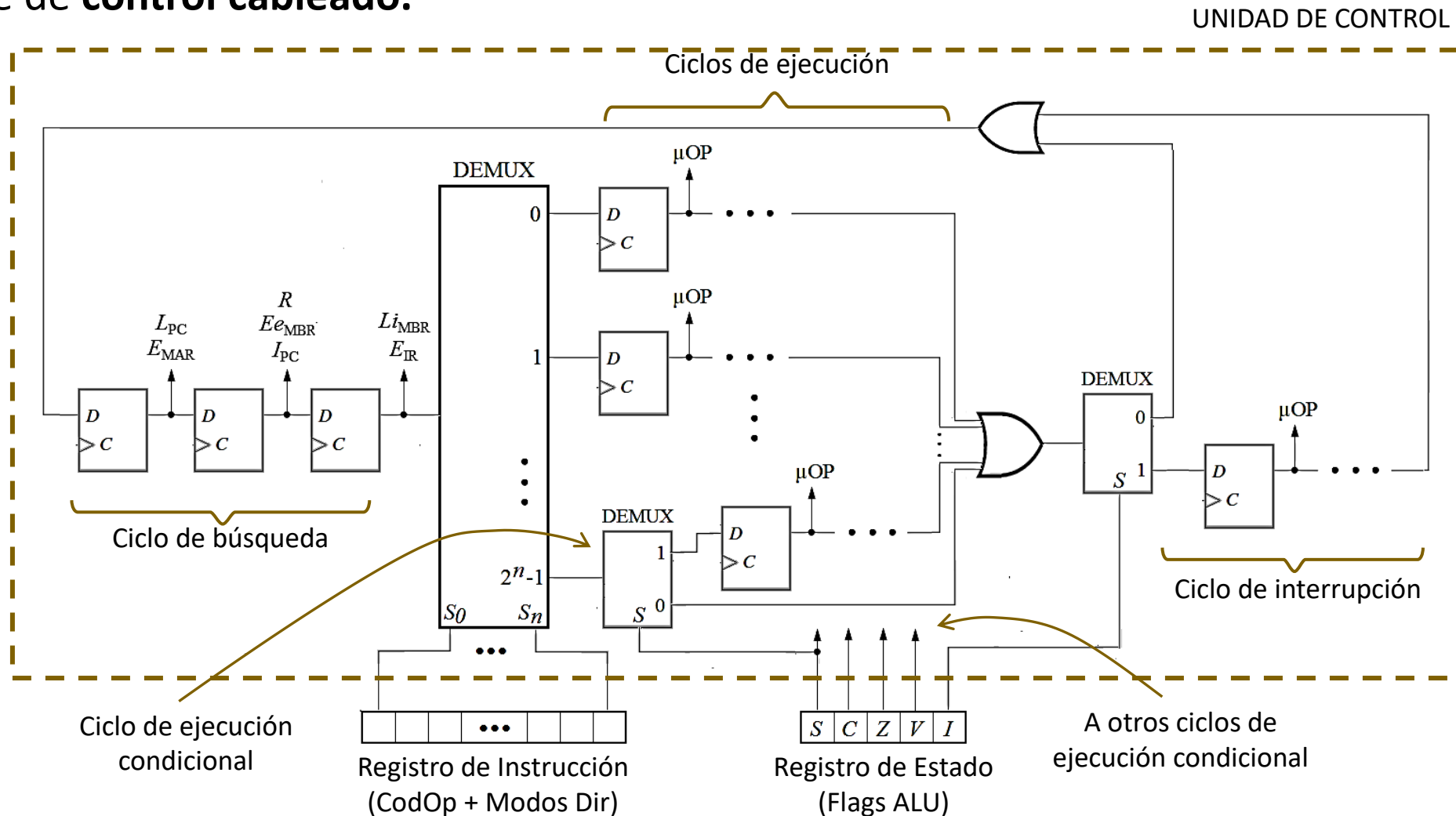
Diseño de la Unidad de Control

El funcionamiento general de la Unidad de Control puede ser descrito con un diagrama de flujo como:



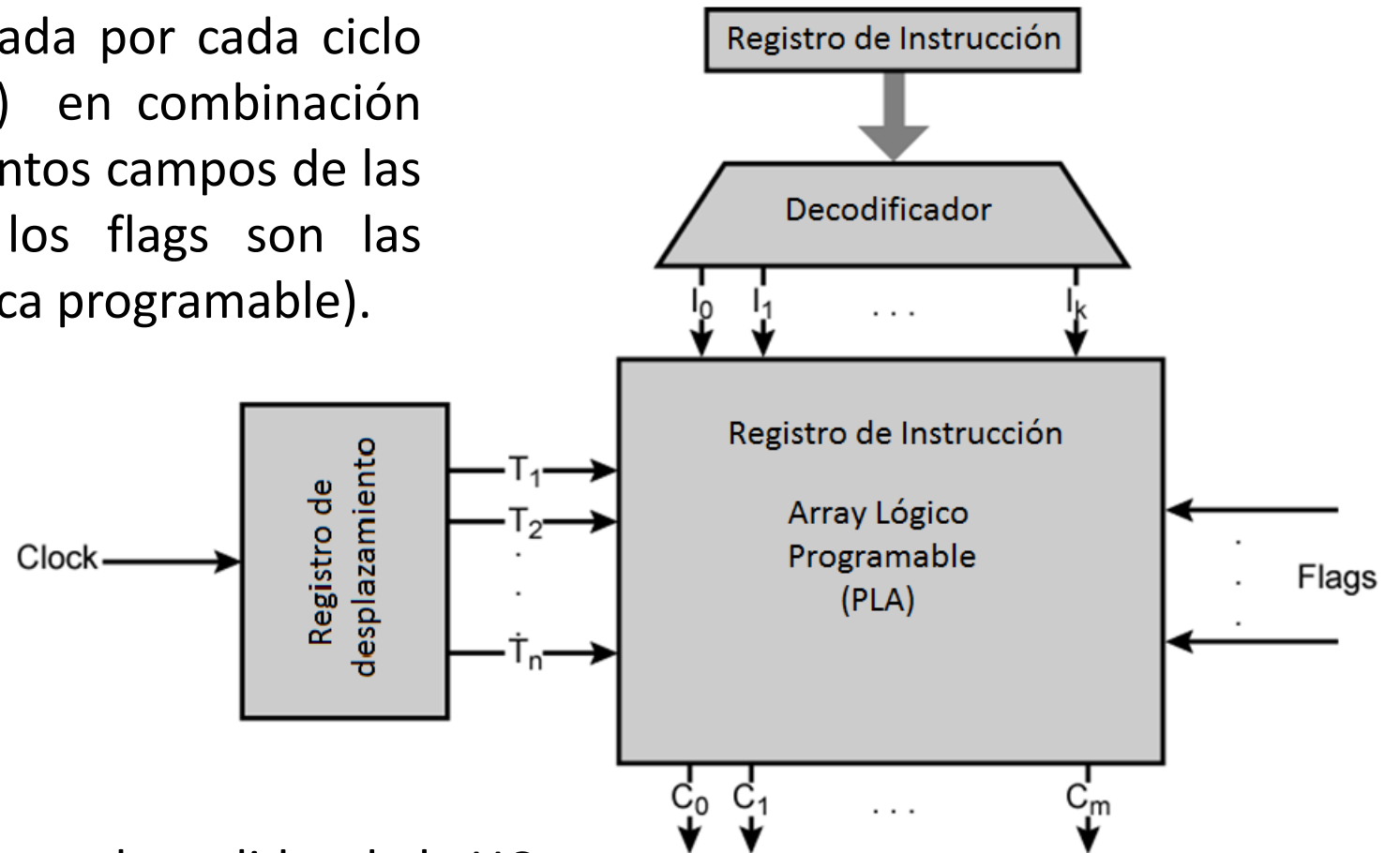
Control cableado: diseño secuencial

La Unidad de Control puede ser diseñada como un **circuito secuencial**. Esta técnica recibe el nombre de **control cableado**.



Control cableado: diseño con PLA

Otra variante es utilizar un registro de secuencia que genera una señal decodificada por cada ciclo (un registro de desplazamiento) en combinación con un decodificador de los distintos campos de las instrucciones, que junto con los flags son las entradas de una PLA (Matriz lógica programable).



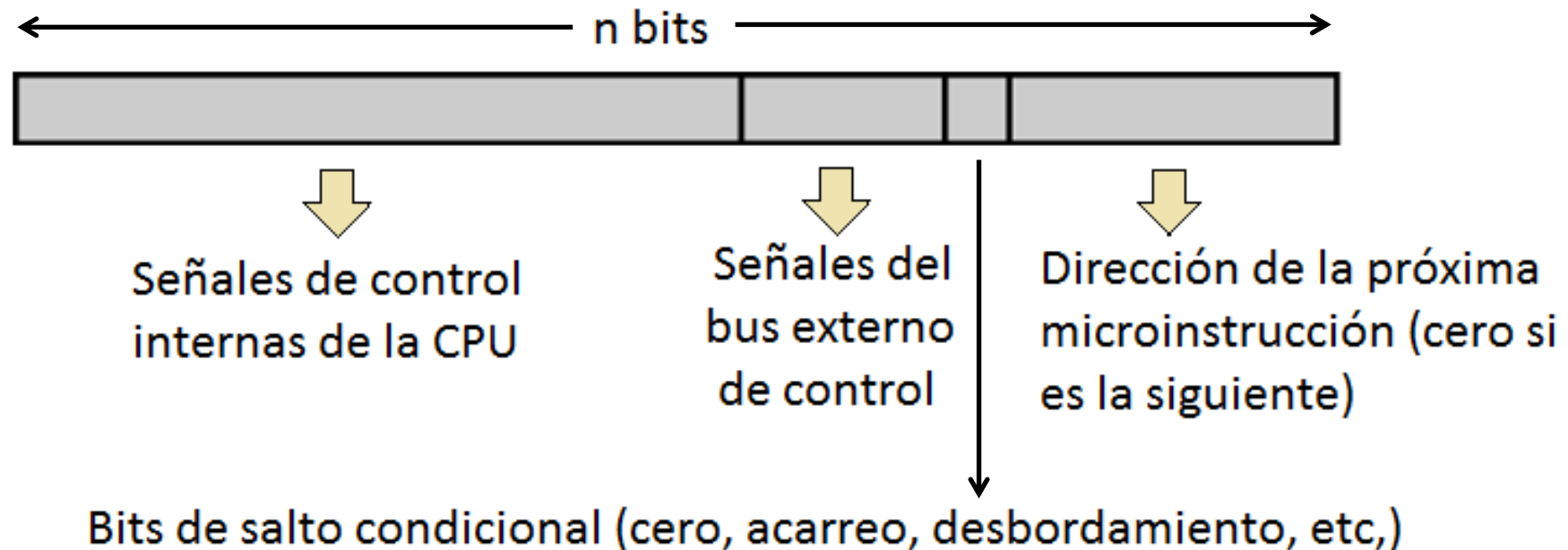
Las funciones de salida de la PLA son las salidas de la UC.

Control microprogramado

- En el control por lógica cableada, la lógica del secuenciamiento y las micro-operaciones resulta muy compleja. Esto genera varios inconvenientes:
 - Dificulta el diseño y el testeo.
 - El diseño resulta inflexible.
 - Esto impide agregar nuevas instrucciones o modificar las existentes
- Una técnica alternativa la constituye la **micro-programación o firmware**.
- Cada instrucción (código de máquina) tiene una *secuencia* de microinstrucciones.
- Todo lo que una Unidad de Control hace es generar un conjunto de señales de control para cada micro-operación durante un ciclo de reloj.
- Cada señal de control está activa o inactiva. Por lo tanto, se puede representar cada señal de control con un bit.
- El conjunto de bits de todas las señales de control conforman una *palabra de control o micro instrucción*.

Control microprogramado

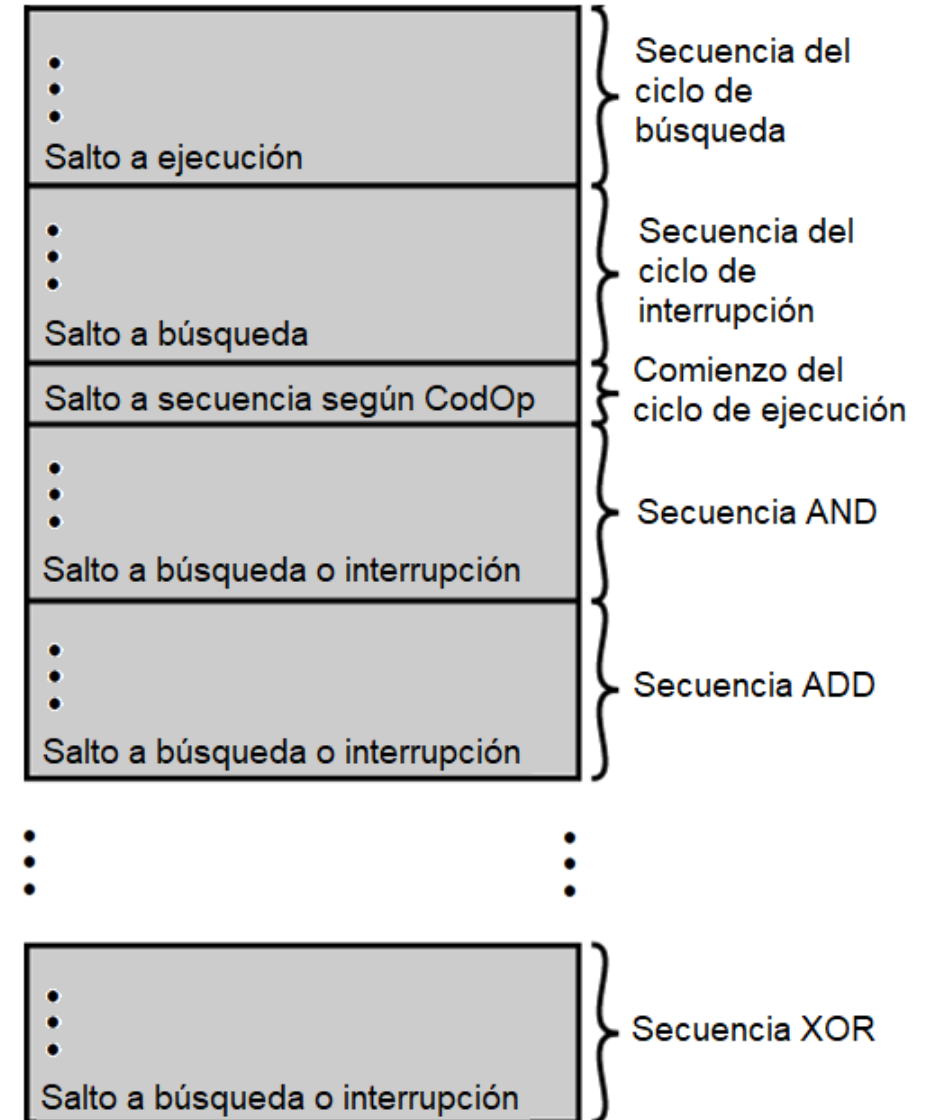
- Todas las palabras de control de todas las microoperaciones que conforman el repertorio de instrucciones pueden almacenarse en una memoria de sólo lectura.
- En el caso de las instrucciones condicionales, se adicionan los bits de control (flags) que establecen la condición, y una dirección de la memoria de control en la que se especifica la siguiente microinstrucción.



Organización de una Memoria de Control

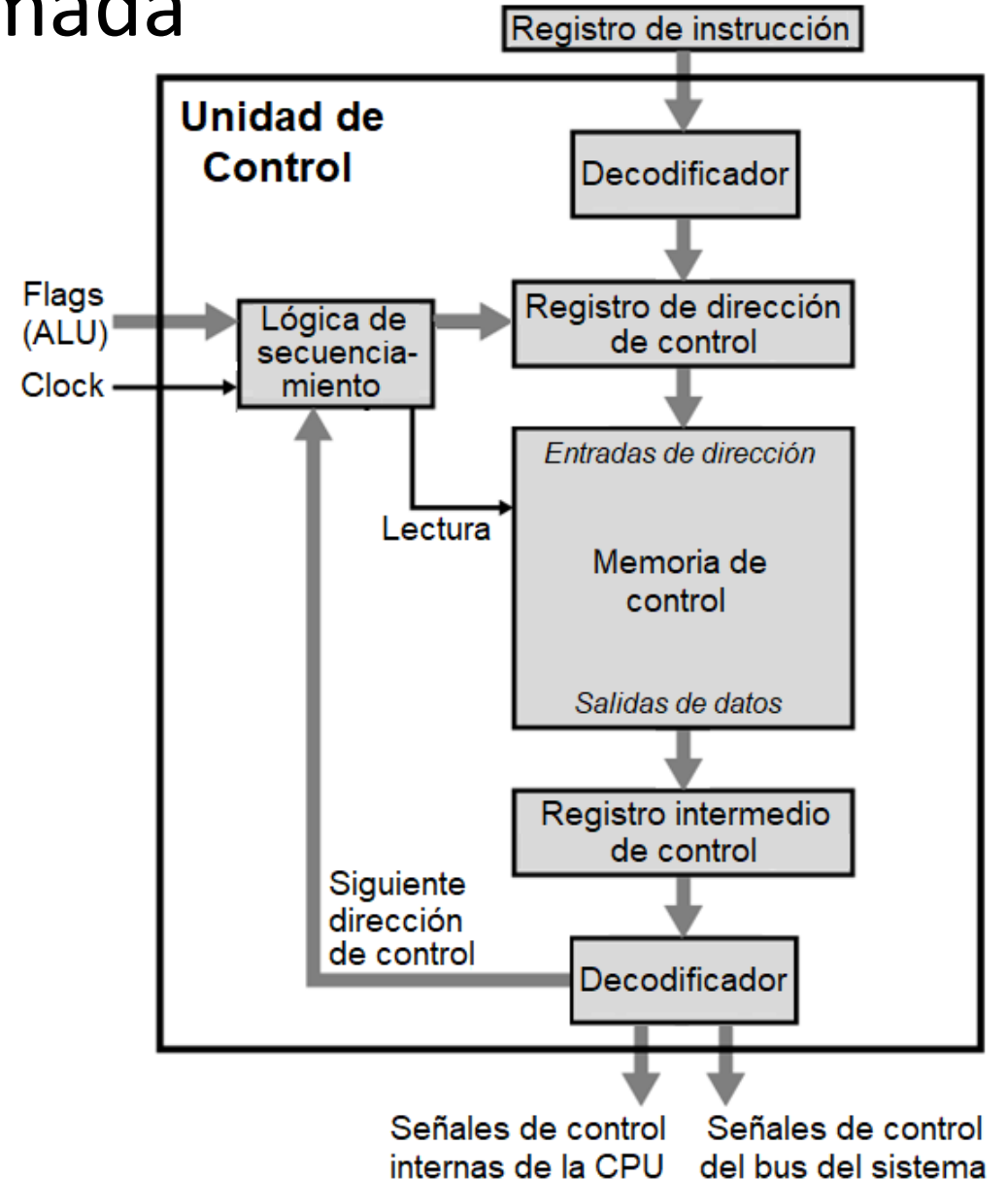
Las secuencias del ciclo de búsqueda, del de interrupción y los ciclos de ejecución de cada instrucción del repertorio del procesador se cargan en la memoria de control.

Al final de cada secuencia, existe una palabra que provoca el salto a la secuencia que corresponde en la ejecución del ciclo de instrucción.



Unidad de Control microprogramada

- La lógica de secuenciamiento envía un comando de lectura a la memoria de control
- La palabra especificada por el registro de dirección de control es leída en el registro intermedio de control.
- El registro intermedio de control almacena las señales de control y contiene la información de la siguiente dirección.
- La lógica de secuenciamiento carga la nueva dirección en el registro de dirección de control, basada en la información proporcionada por el registro intermedio de control y por los indicadores de la ALU.
- Todo esto sucede en un ciclo de reloj.



Unidad de Control microprogramada

Dependiendo de los indicadores (flags) de la ALU y del registro intermedio de control, se decide una de las siguientes tres decisiones:

- Buscar la siguiente microinstrucción
 - Se suma 1 al registro de dirección de control.
- Saltar a una nueva secuencia (por ejemplo, a una de ejecución de la instrucción en curso) si la anterior fue una microinstrucción de salto.
 - El campo de dirección del registro intermedio de control se carga en el registro de dirección de control.
- Saltar a la secuencia de captación de una nueva instrucción de máquina
 - Se carga el registro de dirección de control con la dirección de la secuencia de búsqueda.

Repaso conceptual

1. Explique el concepto de interrupción, y enumere sus posibles fuentes.
2. ¿Cómo se modifican las etapas de un ciclo de instrucción detallado al introducir la atención de las interrupciones?
3. ¿Qué pasos debe realizar el gestor de interrupción?
4. Explique y detalle las funciones de la Unidad de Control
5. ¿Cuáles son las entradas y las salidas de la Unidad de Control?
6. Explique conceptualmente en que consisten las técnicas de diseño de control cableado.
7. Explique conceptualmente en que consisten la técnica de diseño de control microprogramado.

Lecturas recomendadas

- Mano M., Kime, C. - Fundamentos de diseño lógico y de computadoras - 3º Ed. - Prentice Hall. Año 2000.

→ *Capítulo 12*

- Stallings, Williams - Organización y Arquitectura de Computadoras - 5º Ed. - Prentice Hall. Año 2000.

→ *Capítulos 14-15*