La Unidad de Control

- Microoperaciones
- Control del CPU
- Ejecución de Microinstrucciones
- Microprogramación

Introducción

La unidad de control es la parte del CPU que se encarga de que las cosas sucedan, ya que emite señales de control externas al CPU para producir el intercambio de datos con la memoria y los módulos de E/S. También emite sañales de control internas para transferir datos entre registros, hacer que la ALU ejecute una función y otras operaciones internas. La entrada a la unidad de control está compuesta por el registro de instrucción, los indicadores, y señales de control.

La responsabilidad principal de la unidad de control es hacer que se produzca una secuencia de operaciones elementales, lamadas *microoperaciones*, durante el desarrollo de un ciclo de instrucción.

Requisitos funcionales del CPU

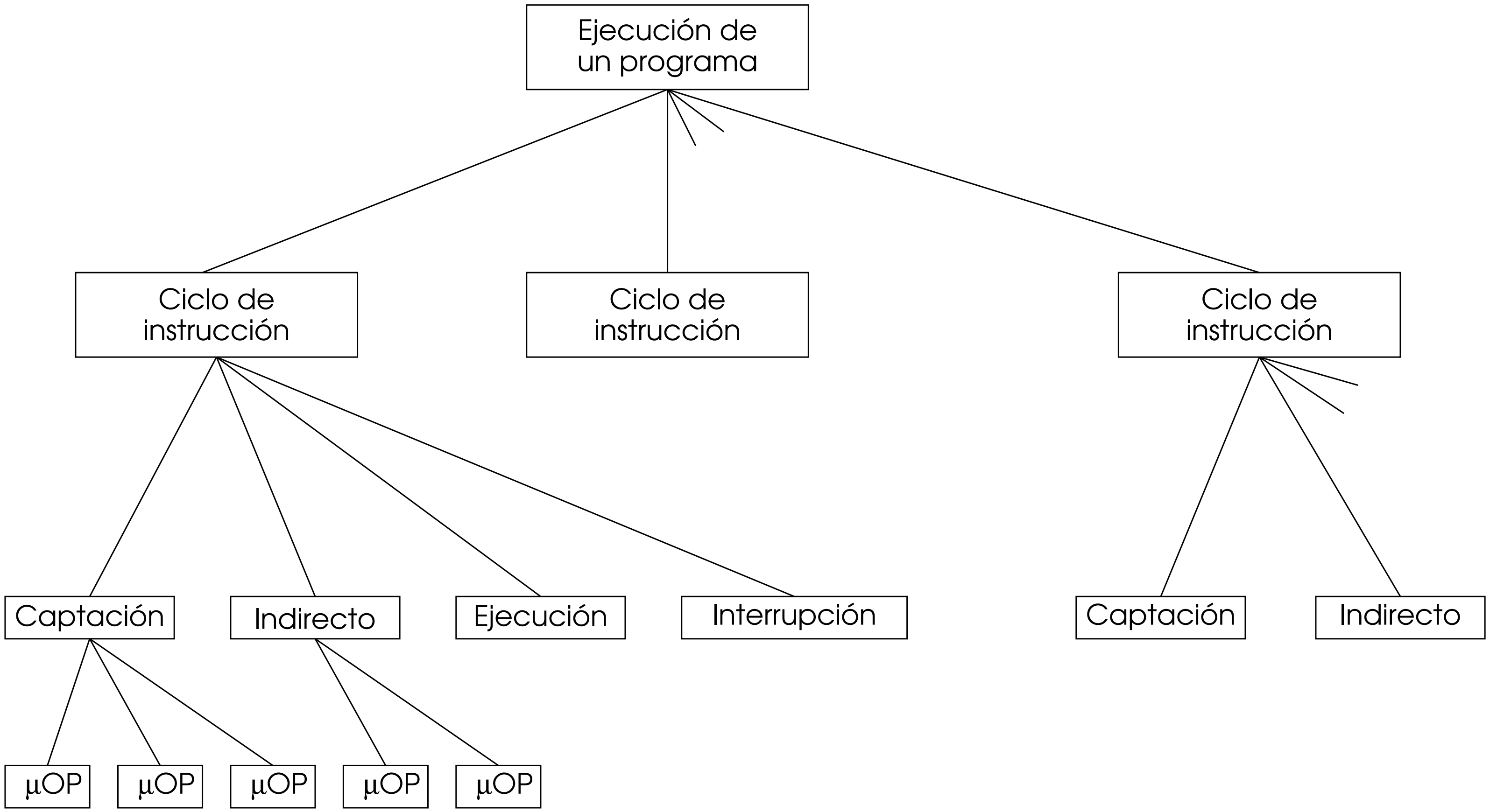
1. Operaciones (códigos de operación).
2. Modos de direccionamiento.
3. Registros.
4. Interfaz con el módulo de E/S.
5. Interfaz con el módulo de memoria.
6. Estructura del procesamiento de interrupciones.

Los puntos del 1 al 3 estan definidos por el conjunto de instrucciones, el 4 y el 5 por el bus del sistema y el 6 por el bus del sistema y el tipo de soporte que ofrece el CPU al sistema operativo. Los seis puntos determinan lo que debe hacer el CPU, en la siguiente parte se examina cómo se realizan las funciones.

Microoperaciones

Cada ciclo de instrucción se puede considerar formado por varias unidades pequeñas. Una subdivisión práctica es captación, ciclo indirecto, ejecución, e interrupción. Los que aparecen siempre son los de captación y de ejecución. Cada uno de los ciclos pequeños implica una serie de pasos y el uso de registros del CPU, estos pasos se conocen como *microoperaciones*.

Elementos que constituyen la ejecución de un programa



El ciclo de captación

Para el estudio que se presenta se supone que se usan cuatro registros:

- *El Contador de programa ("Program Counter", PC)*.
- *El Registro de instrucción ("Instruction Register", IR)*.
- *El Registro de dirección de memoria ("Memory Address Register",MAR)*.
- *El Registro intermedio de memoria ("Memory Buffer Register", MBR)*.

El ciclo de captación consta de tres pasos (de transferencia) y cuatro microoperaciones. Cada microoperación involucra la transferencia de datos hacia dentro o hacia fuera de un registro. La secuencia se puede representar como:

$$
\begin{array}{llll}
t_1: & \text{MAR} & \leftarrow & \text{(PC)} \\
t_2: & \text{MBR} & \leftarrow & \text{Memoria} \\
 & \text{PC} & \leftarrow & \text{(PC)} + 1 \\
t_3: & \text{IR} & \leftarrow & \text{(MBR)}
\end{array}
$$

La tercera operación esta agrupada con la segunda, aunque también puede agruparse con la cuarta sin problema. Los agrupamientos de microoperaciones deben cumplir:

- Seguir la secuencia correcta de eventos.
- Evitar los conflictos.

El ciclo indirecto

Una vez que se ha captado una microinstrucción, el siguiente paso es captar los operandos fuente.

Las microoperaciones involucrados son:

t_1: MAR $\leftarrow$ (IR(Dirección))

t_2: MBR $\leftarrow$ Memoria

t_3: IR $\leftarrow$ (MBR(Dirección))

El ciclo de interrupción

Cuando termina el ciclo de ejecución, se realiza una comprobación para determinar si ha ocurrido alguna interrupción habilitada. Se es así, tiene lugar un ciclo de interrupción. De una máquina a otra varía, más una secuencia simple de eventos es:

t_1:	MBR	$\leftarrow$	(PC)
t_2:	MAR	$\leftarrow$	Dirección de salvaguarda
	PC	$\leftarrow$	Dirección de la rutina
t_3:	Memoria	$\leftarrow$	(MBR)

El ciclo de ejecución

En una máquina con N códigos de operación diferentes, pueden ocurrir N secuencias diferentes de microoperaciones. Por ejemplo una instrucción de suma el contenido de la posición X al registro R1 (**ADD R1, X**), puede tener la siguiente secuencia de microoperaciones:

t_1:	MAR	$\leftarrow$	(IR(Dirección))
t_2:	MBR	$\leftarrow$	Memoria
t_3:	IR	$\leftarrow$	(R1) + (MBR)

Una instrucción que incremente en 1 la posición X y salte si el resultado es cero (**ISZ X**) tendría la siguiente secuencia:

t_1: MAR $\leftarrow$ (IR(Dirección))

t_2: MBR $\leftarrow$ Memoria

t_3: MBR $\leftarrow$ (MBR)+1

t_4: Memoria $\leftarrow$ (MBR)

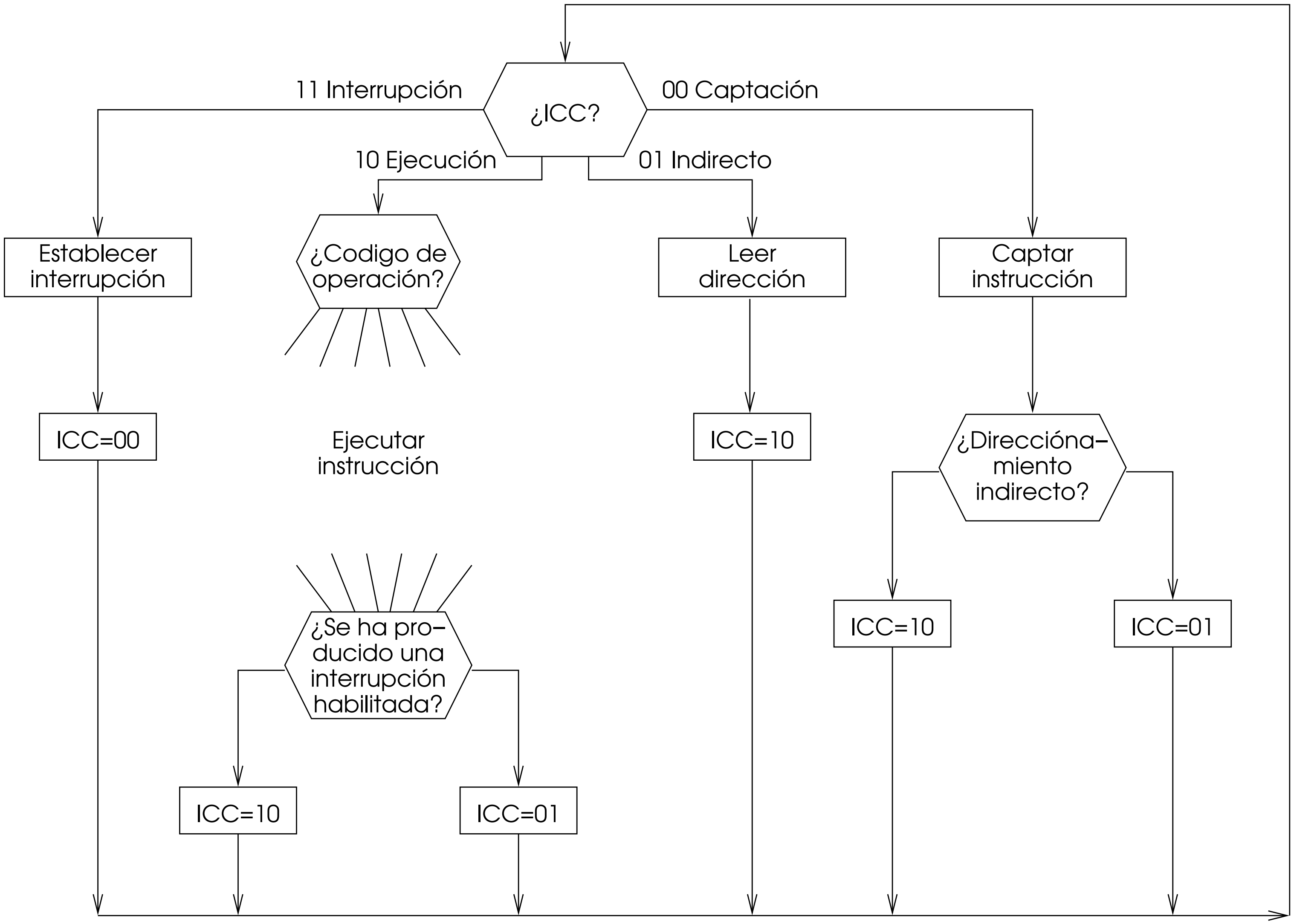
Si (MBR=0) entonces (PC $\leftarrow$ (PC) + 1)

El ciclo de instrucción

Suponiendo que hay un registro de 2 bits llamado *código de ciclo de instrucción (“instruction cycle code”, ICC)* . El ICC designa el estado del CPU en el que se encuentra:

- 00: Captación.
- 01: Indirecto.
- 10: Ejecución.
- 11: Interrupción.

Diagrama de flujo del ciclo de instrucción



Control del CPU

Para caracterizar la unidad de control se requiere:

- Definir los elementos básicos del CPU.
- Describir las microoperaciones que ejecuta el CPU.
- Determinar las funciones que debe realizar la unidad de control para hacer que se ejecuten las microoperaciones.

Las microoperaciones se pueden clasificar en una de las siguientes categorías:

- Transferencia de datos de un registro a otro.
- Transferencia de datos de un registro a una interfaz externa.
- Trasnferencia de datos de una interfaz externa a un registro.
- Operaciones aritméticas o lógicas, usando registros para entra-

da y salida.

La unidad de control realiza dos tareas básicas:

- *Secuenciamiento*. La unidad de control hace que el CPU avance a través de una serie de microoperaciones en la secuencia oportuna, basada en el programa que se está ejecutando.
- *Ejecución*. La unidad de control hace que se ejecute cada microoperación.

La clave de cómo funciona la unidad de control es mediante el uso de señales de control.

Señales de control

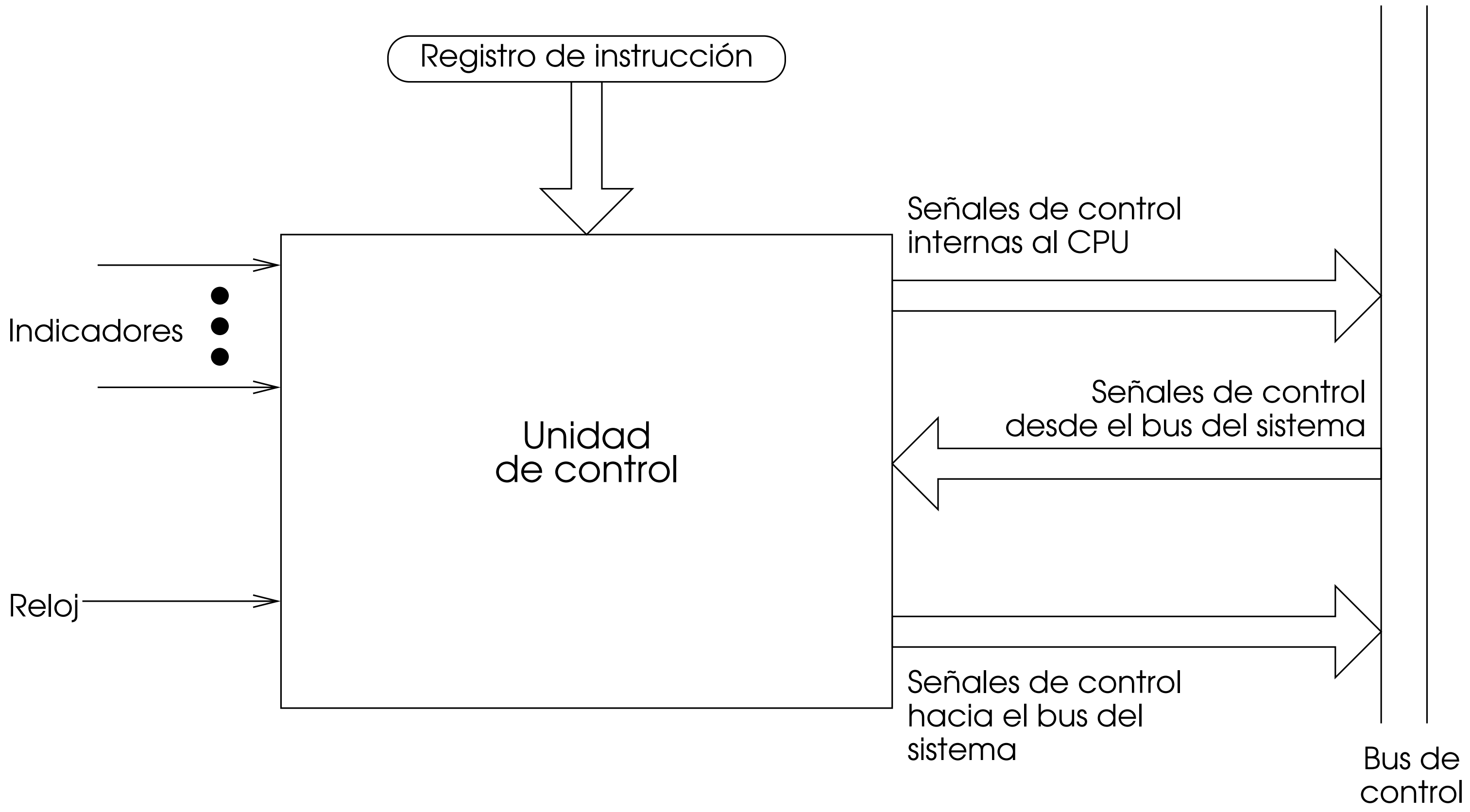
Las entradas son:

- *Reloj*. La unidad de control hace que se ejecute una microoperación (o un conjunto) en cada pulso de reloj.
- *Registro de instrucción*. Determina que microoperación se debe realizar.
- *Indicadores*. Usado para determinar el estado del CPU y el resultado de operaciones anteriores de la ALU.
- *Señales de control del bus de control*. Señales de control desde el bus del sistema.

Las salidas son:

- *Señales de control internas del CPU* Para transferir datos de un registro a otro y para activar funciones específicas de la ALU.
- *Señales de control hacia el bus de control* Señales de control de la memoria y señales de control de los módulos de E/S.

Modelo de la unidad de control



Para el ciclo de captación la unidad de control envia simultáneamente las siguientes señales de control:

1. Una señal para abrir puertas que permita que el contenido de MAR aparezca en el bus de direcciones.

2. Una señal de lectura de memoria en el bus de control.

3. Una señal para abrir puertas que permita almacenar el contenido del bus de datos en MBR.

4. Señales de control de la lógica que sumen 1 al contenido de PC y almacene el resultado.

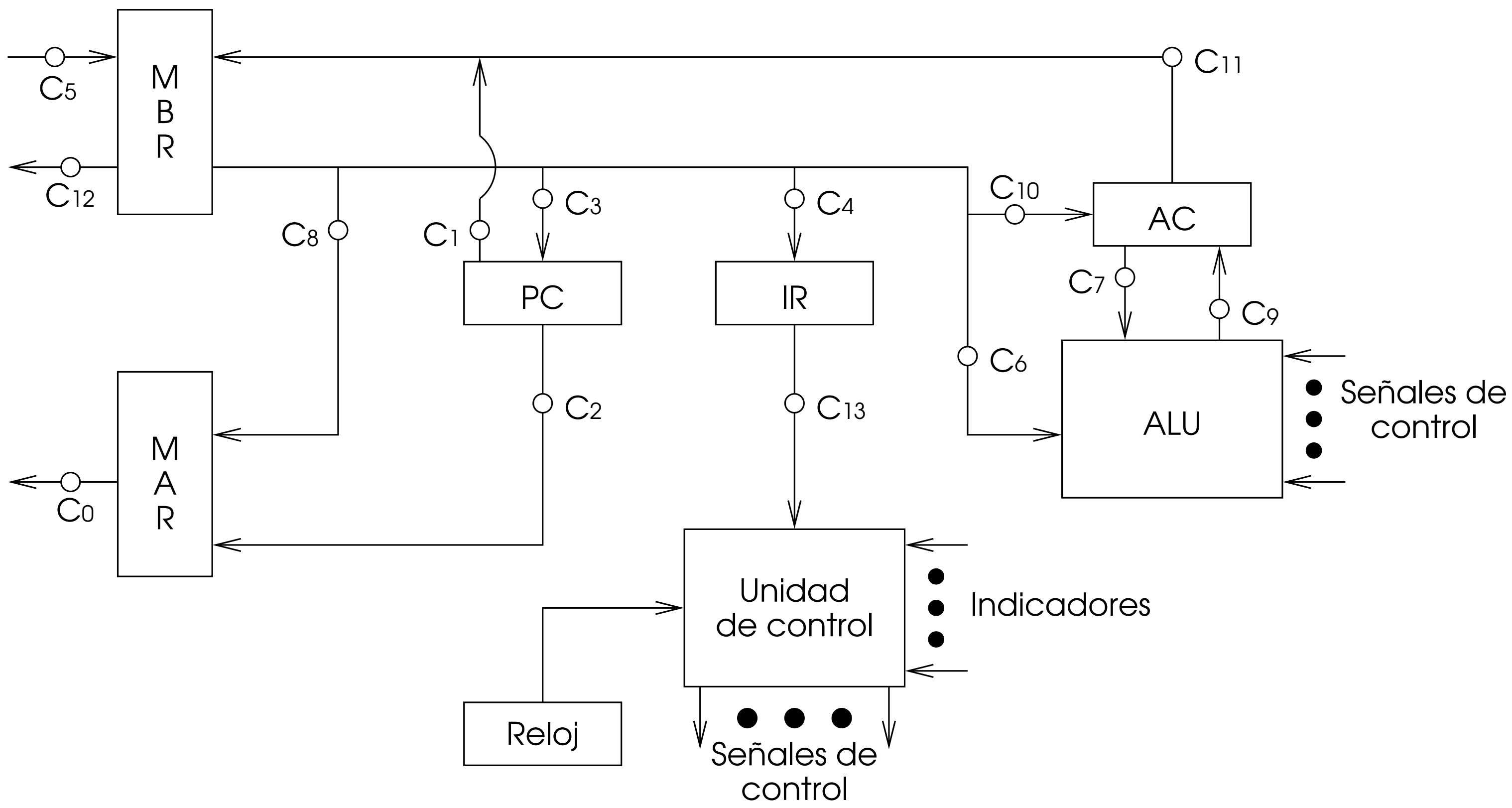
Después de lo anterior, la unidad de control envía una señal de control para abrir las puertas entre MBR e IR.

Ejemplo de señales de control

Se ilustra el funcionamiento de la unidad de control con una CPU sencilla con un único acumulador. En cada ciclo de reloj, la unidad de control lee todas sus entradas y emite un conjunto de señales de control. Las señales de control van hacia tres destinos distintos:

- *Caminos de datos*
- *ALU*
- *Bus del sistema*

Caminos de datos y señales de control



Microoperaciones y señales de control		
Captación	t_1:MAR $\leftarrow$ (PC)	C_2
	t_2:MBR $\leftarrow$ Memoria	C_5, C_R
	PC $\leftarrow$ (PC)+1	
	t_3:IR $\leftarrow$ (MBR)	C_4
Indirecto	t_1:MAR $\leftarrow$ (IR(Dirección))	C_8
	t_2:MBR $\leftarrow$ Memoria	C_5, C_4
	t_3:IR(Dirección) $\leftarrow$ (MBR(Dirección))	C_4
Interrupción	t_1:MBR $\leftarrow$ (PC)	C_1
	t_2:MAR $\leftarrow$ Dirección de salvaguarda	
	PC $\leftarrow$ Dirección de la rutina	
	t_3:Memoria $\leftarrow$ (MBR)	C_{12}, C_W

C_R =Señal de control de lectura (“read”) hacia el bus del sistema

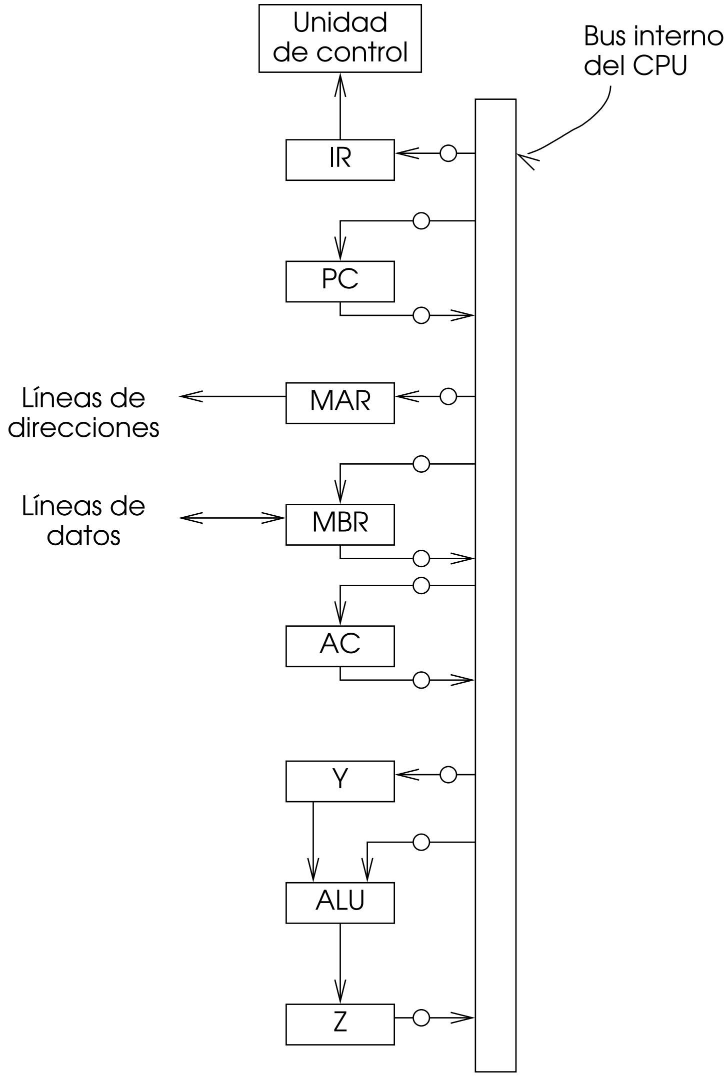
C_W =Señal de control de escritura (“write”) hacia el bus del sistema

Organización interna del CPU

Usando un bus interno al CPU puede disponerse del modo que se muestra más adelante. El registro Y proporciona un almacenamiento temporal de la otra entrada. El registro Z proporciona el almacenamiento temporal de salida.

$$
\begin{array}{lcll}
t_1: & \text{MAR} & \leftarrow & \text{(IR(Dirección))} \\
t_2: & \text{MBR} & \leftarrow & \text{Memoria} \\
t_3: & \text{Y} & \leftarrow & \text{(MBR)} \\
t_4: & \text{Z} & \leftarrow & \text{(AC)+(Y)} \\
t_5: & \text{AC} & \leftarrow & \text{(Z)}
\end{array}
$$

CPU con bus interno



Implementación Cableada

En una implementación cableada de la unidad de control se tiene un circuito combinacional, en donde las señales lógicas de entrada se trsforman en un cojunto de señales de salida, que son las señales de control.

Las **entradas de la unidad de control** clave son el registro de instrucción, el reloj, los indicadores y las señales de control del bus. Las primeras dos entradas no son útiles directamente a la unidad de control y en los dos últimos grupos, cada bit individual tiene algún significado.

El registro de instrucción contiene el código de operación, con el cual la unidad de control realiza acciones diferentes, ya que emite una combinación diferente de señales de control para cada instrucción, es decir para una entrada lógica única debe haber un código de operación. Lo anterior se realiza con un *decodificador*.

El reloj de la unidad de control emite una secuencia repetitiva de pulsos, que sirven para delimitar la duración de las microoperaciones. El período de los pulsos debe ser suficientemente largo para permitir la propagación de las señales.

Lógica de la Unidad de Control

Para cada señal de control se debe obtener una expresión booleana de la señal como una función de las entradas. Considerando el ejemplo visto previamente con las secuencias de microoperaciones y de señales de control necesarias para controlar tres de las cuatro fases del ciclo de instrucción.

Si se considera una señal de control, C_5, con la que se hace que se lean datos del bus de datos externo en MBR, la cual es usada por los subciclos de captación e indirecto, si se definen dos nuevas señales de control, P y Q, con el siguiente significado:
PQ = 00 Ciclo de captación
PQ = 01 Ciclo indirecto
PQ = 10 Ciclo de ejecución
PQ = 11 Ciclo de interrupción

Entonces C_5 se debe poner a uno en la segunda unidad de tiempo de los ciclos respectivos, o

$$C_5 = \bar{\mathsf{P}} \cdot \bar{\mathsf{Q}} \cdot T_2 + \bar{\mathsf{P}} \cdot \mathsf{Q} \cdot T_2$$

La señal C_5 también es usada en el ciclo de ejecución, si se supone que sólo haya tres instrucciones que leen de la memoria: LDA, ADD y AND. La señal C_5 se define como:

$$C_5 = \bar{\mathsf{P}} \cdot \bar{\mathsf{Q}} \cdot T_2 + \bar{\mathsf{P}} \cdot \mathsf{Q} \cdot T_2 + \mathsf{P} \cdot \bar{\mathsf{Q}} \cdot (\mathsf{LDA} + \mathsf{ADD} + \mathsf{AND}) \cdot T_2$$

El resultado sería un conjunto de ecuaciones booleanas que definiría el comportamiento de la unidad de control, y por tanto del CPU.

Al final de cada subciclo, la unidad de control emite una señal que provoca que el generador de temporización se reinicialice y emita T_1, también establece los valores adecuados de P y Q para definir el siguiente subciclo a ejecutar.

Unidad de control con entrads decodificadas