

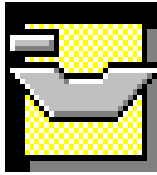


ARQUITECTURAS DE COMPUTADORES

2º CURSO INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

TEMA 2 – UNIDAD DE CONTROL

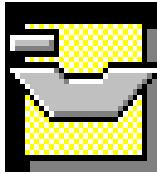
**JOSÉ GARCÍA RODRÍGUEZ
JOSÉ ANTONIO SERRA PÉREZ**



El procesador

La Unidad Central de Proceso

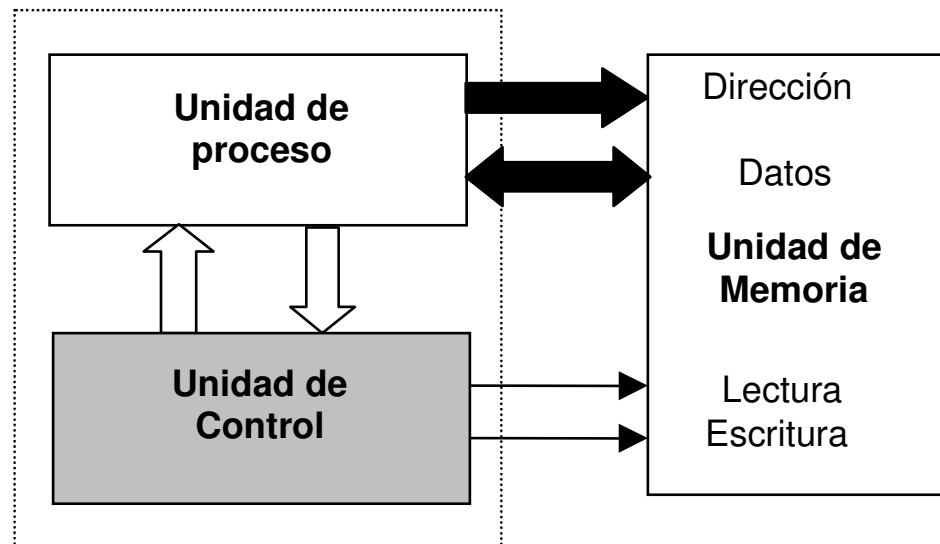
- ◆ Introducción
- ◆ Empleo de buses
- ◆ Fases de Ejecución de las Instrucciones
 - ◆ Introducción
 - ◆ Establecimiento de las fases
 - ◆ La ruta de datos
- ◆ Unidad de Control
 - ◆ identificación de las señales de control
 - ◆ Activación de las señales de control
- ◆ Diseño de la Unidad de control
 - ◆ Método de la tabla de estados
 - ◆ Método del Contador de secuencia
 - ◆ Abstracción de la máquina
- ◆ Conclusiones

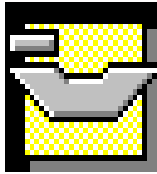


EL procesador

Introducción

- ◆ Mecanismo de ejecución de las instrucciones
- ◆ Diseño de la Unidad de Control

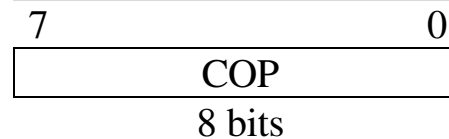




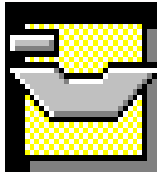
Instrucciones

Repertorio de instrucciones elegido

Formato tipo 1

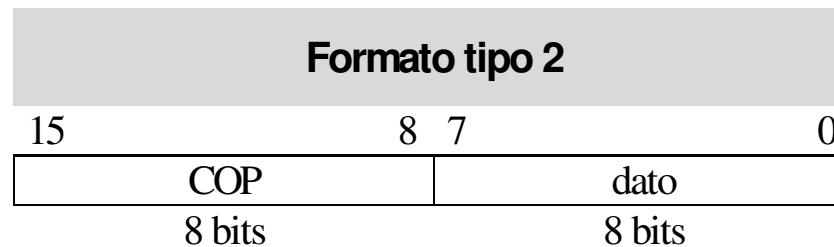


Operación	Sintaxis	Descripción	Cód. Oper.
Suma	ADD r1	$A \leftarrow A + r1$	30h, 31h, 32h, 33h, 45h
Resta	SUB r1	$A \leftarrow A - r1$	18h, 19h, 1Ah, 1Bh, 46h
And	ANA r1	$A \leftarrow A \text{ and } r1$	20h, 21h, 22h, 23h, 48h
Or	ORA r1	$A \leftarrow A \text{ or } r1$	24h, 25h, 26h, 27h, 49h



Instrucciones

Repertorio de instrucciones elegido



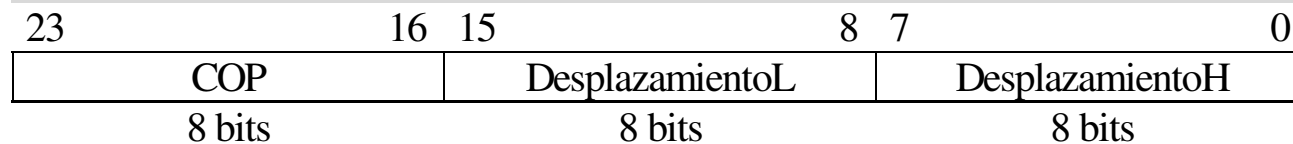
Operación	Sintaxis	Descripción	Cód. Oper.
Suma Inmediata	ADI dato	$A \leftarrow A + dato$	35h
Resta Inmediata	SUI dato	$A \leftarrow A - dato$	36h
And Inmediata	ANI dato	$A \leftarrow A \text{ and } dato$	68h
Or Inmediata	ORI dato	$A \leftarrow A \text{ or } dato$	69h



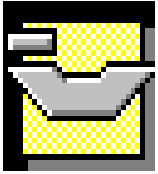
Instrucciones

Repertorio de instrucciones elegido

Formato tipo 3



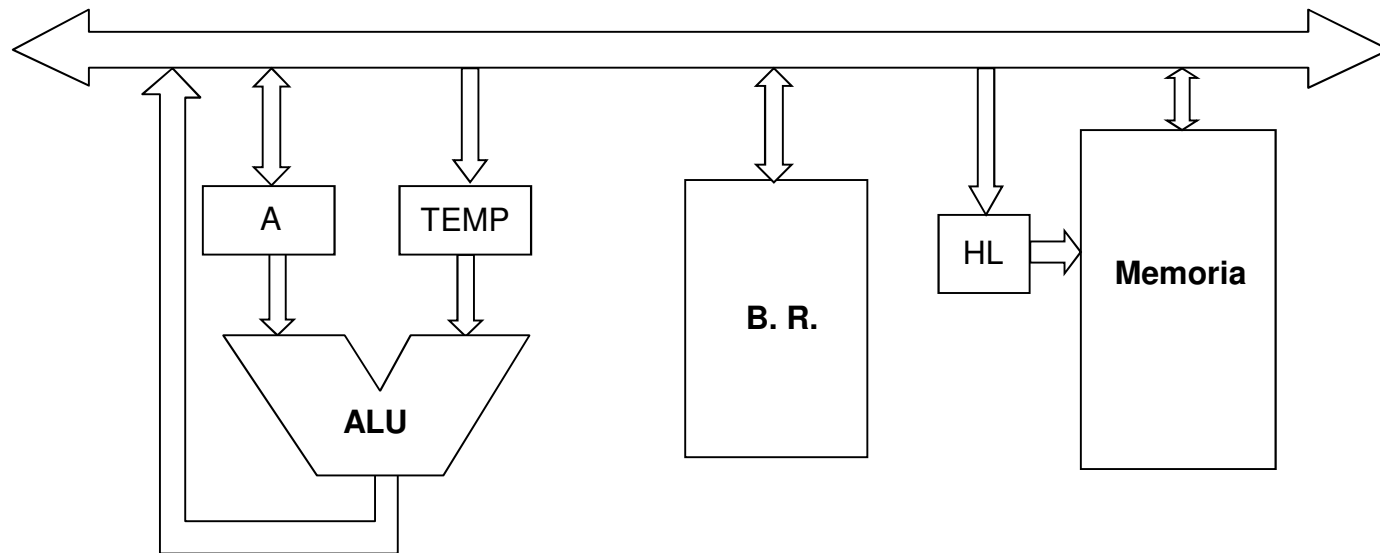
Operación	Sintaxis	Descripción	Cód. Oper.
Cargar	LDA dir	$A \leftarrow M(dir)$	70h
Almacenar	STA dir	$M(dir) \leftarrow A$	71h
Salto incondicional	JMP dir	$PC \leftarrow dir$	74h
Salto si FZ=1	JZ dir	$Si FZ = 1 \Rightarrow PC \leftarrow dir$	72h

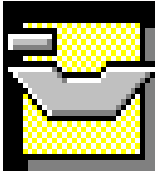


Empleo de
de buses

Ruta de datos con bus único

- ◆ Bus único de datos de 8 bits
- ◆ Bus único de direcciones de 16 bits
- ◆ Necesidad de registros temporales para liberar al bus

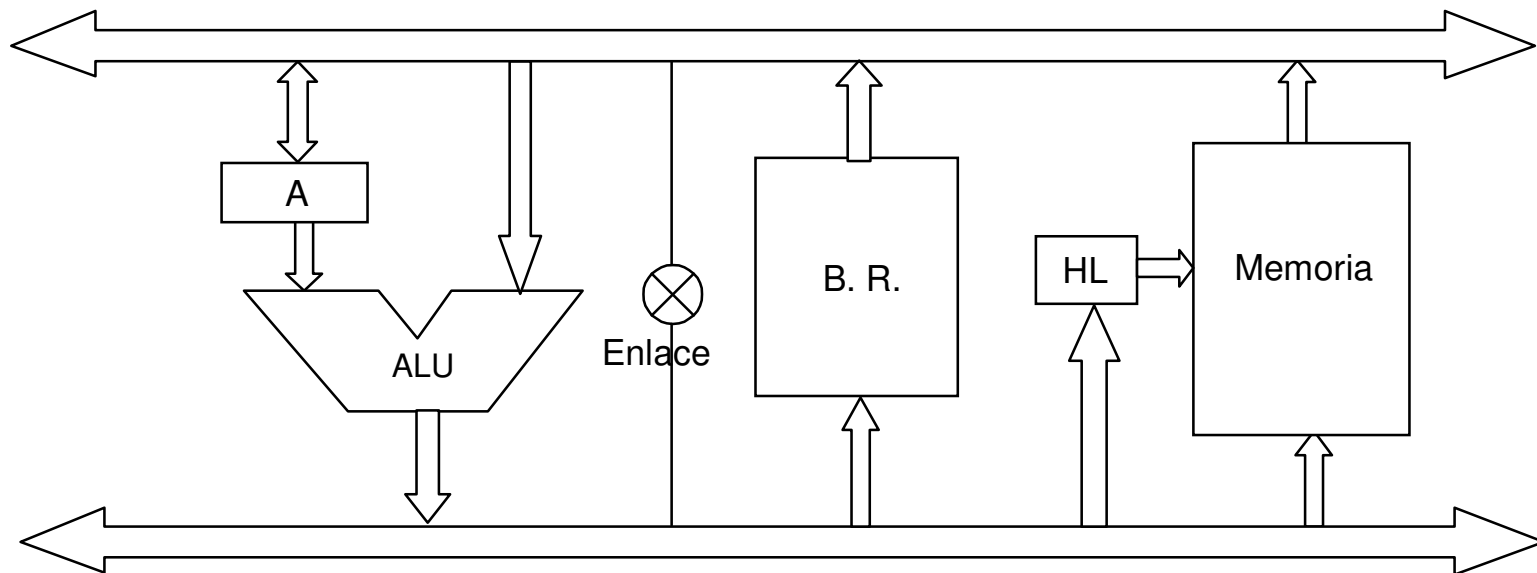


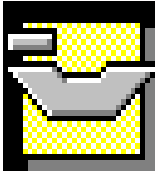


Empleo de
de buses

Ruta de datos con dos buses

- ◆ 2 buses de 8 bits
- ◆ Comunicación con la memoria a través de HL
- ◆ Enlace de bus

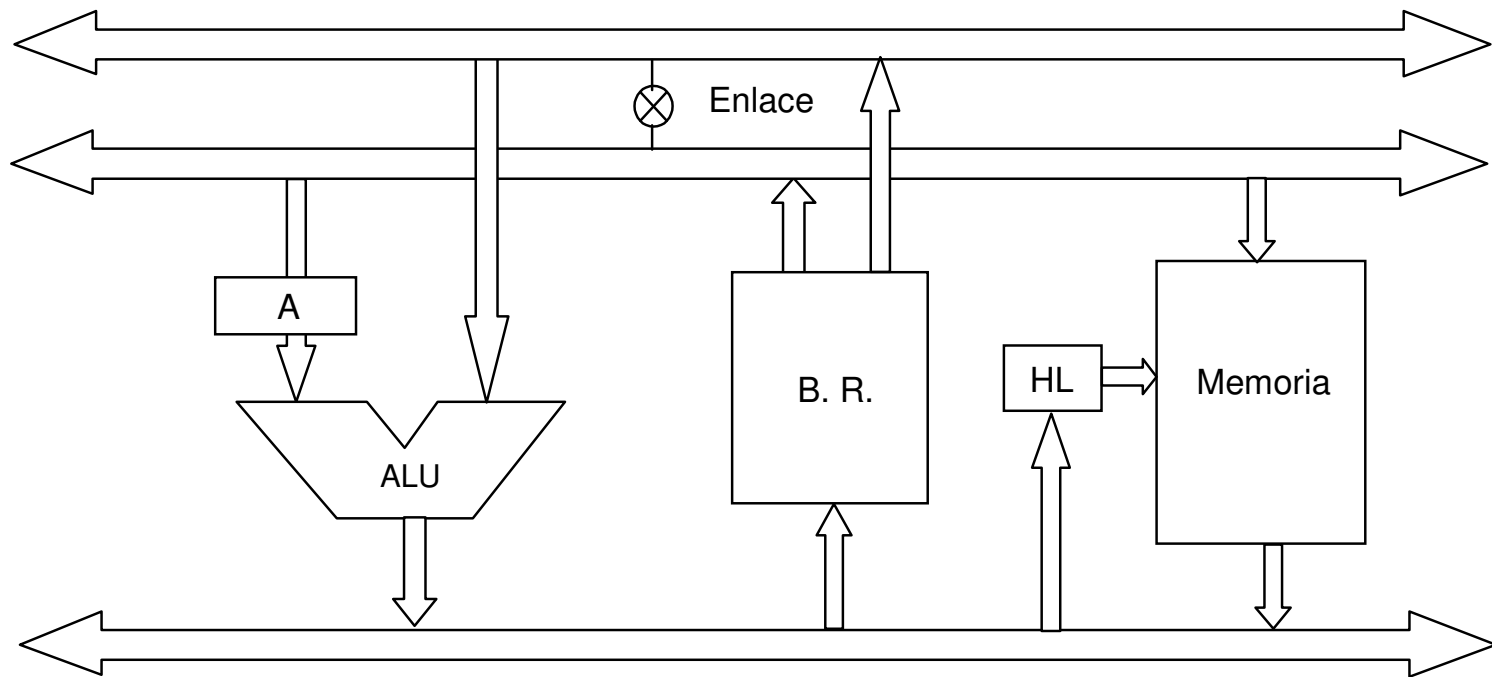




Empleo de
de buses

Ruta de datos con tres buses

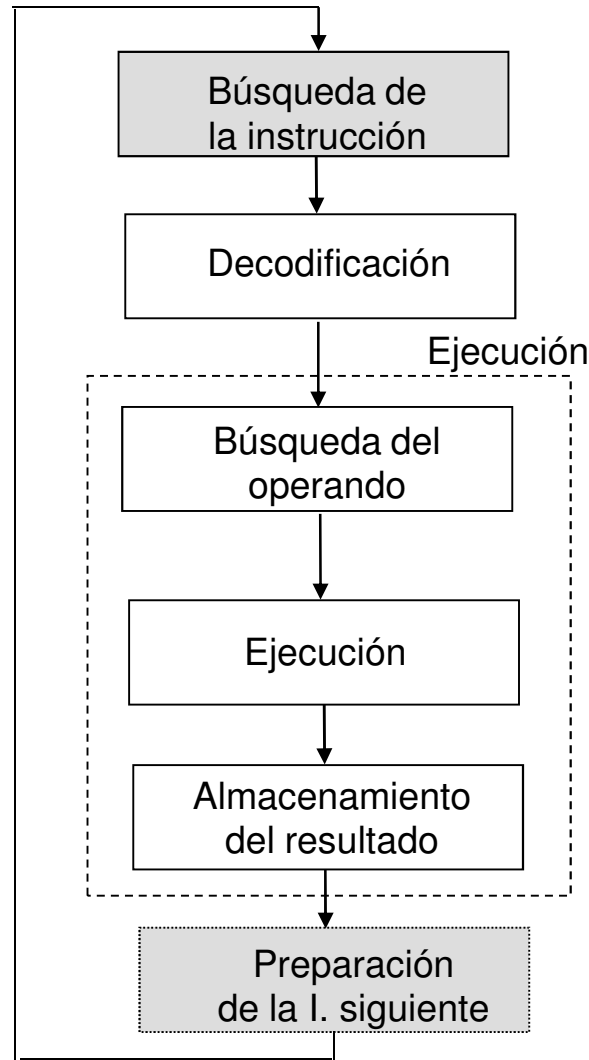
- ◆ 3 buses de 8 bits
- ◆ Comunicación con la memoria a través de HL
- ◆ Enlace de bus



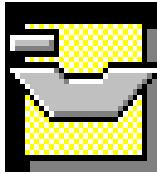


Introducción

Fases de ejec. de las instrucciones



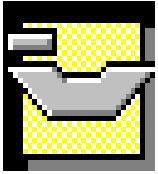
- ◆ Diseño disparado por flanco.
- ◆ Cada fase de ejecución durará un ciclo de reloj.
- ◆ La duración del ciclo de reloj dependerá de la duración de la acción más lenta.
- ◆ No se puede realizar una lectura y una escritura en el banco de registros en el mismo ciclo de reloj.



Fases de
ejecución

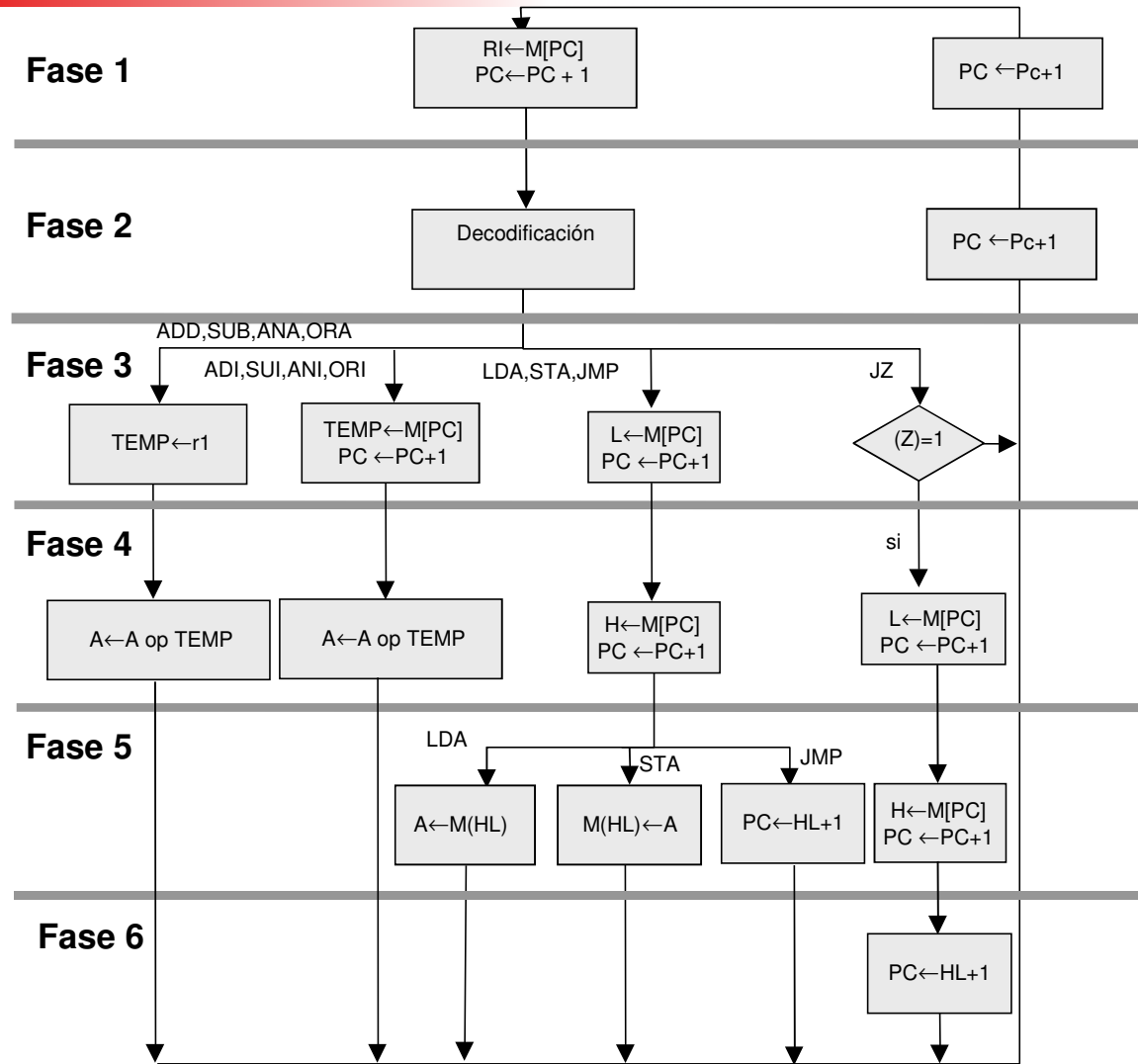
Consideraciones

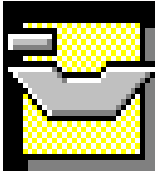
- ◆ Funciones básicas de la MaNoTas:
 - ◆ Acceso al Banco de Registros
 - ◆ Acceso a la Memoria
 - ◆ Operaciones en la ALU
- ◆ Suposiciones:
 - ◆ El tiempo de estas funciones es igual a un ciclo de reloj.
 - ◆ El coste del resto de los elementos es cero.
- ◆ Las acciones asociadas a una fase ocurren en paralelo.
- ◆ Las acciones asociadas a fases sucesivas ocurren en serie.



Fases de ejecución

Establecimiento de las fases





Fases de
Ejecución

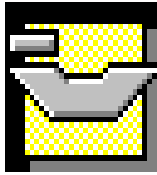
Establecimiento de las fases (1)

◆ Fase 1: Búsqueda de la instrucción

- ◆ $RI \leftarrow M[PC]$
- ◆ $PC \leftarrow PC + 1$

◆ Fase 2. Decodificación

- ◆ Estas dos fases son comunes a todas las instrucciones



Fases de
Ejecución

Establecimiento de las fases (2)

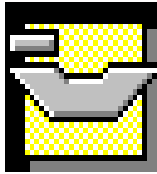
- ◆ **Fase 3: Obtención de operandos y evaluación del código de condición Z.**
 - ◆ **Caso I.** Instrucciones Aritmético-Lógicas
 - ◆ Modo de direccionamiento directo a registro
 - ◆ $TEMP \leftarrow r1$
 - ◆ Modo de direccionamiento inmediato
 - ◆ $TEMP \leftarrow M[PC]$ ($TEMP \leftarrow \text{dato}$)
 - ◆ $PC \leftarrow PC + 1$
 - ◆ **Caso II.** Instrucciones de referencia a memoria y salto incondicional
 - ◆ $L \leftarrow M[PC]$ ($L \leftarrow \text{DirL}$)
 - ◆ $PC \leftarrow PC + 1$
 - ◆ **Caso III.** Instrucción de salto condicional
 - ◆ Z?



Fases de
Ejecución

Establecimiento de las fases (3)

- ◆ **Fase 4: Obtención de operandos, ejecución y conclusión de las instrucciones Aritmético-Lógicas.**
 - ◆ **Caso II.** Instrucciones Aritmético-Lógicas
 - ◆ $A \leftarrow A \text{ op TEMP}$
 - ◆ **Caso II.** Instrucciones de referencia a memoria y salto incondicional
 - ◆ $H \leftarrow M[PC]$ ($H \leftarrow \text{DirH}$)
 - ◆ $PC \leftarrow PC + 1$
 - ◆ **Caso III.** Instrucción de salto condicional
 - ◆ $L \leftarrow M[PC]$ ($L \leftarrow \text{DirL}$)
 - ◆ $PC \leftarrow PC + 1$



Fases de
Ejecución

Establecimiento de las fases (4)

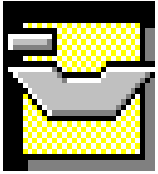
- ◆ **Fase 5: Conclusión de las instrucciones de acceso a memoria y salto incondicional y obtención de operandos.**
 - ◆ **Caso II.** Instrucciones de referencia a memoria
 - ◆ Instrucción de carga
 - ◆ $A \leftarrow M[HL]$
 - ◆ Instrucción de almacenamiento
 - ◆ $M[HL] \leftarrow A$
 - ◆ **Caso II.** Instrucción salto incondicional
 - ◆ $PC \leftarrow HL + 1$
 - ◆ **Caso III.** Instrucción de salto condicional
 - ◆ $H \leftarrow M[PC]$ ($H \leftarrow DirH$)
 - ◆ $PC \leftarrow PC + 1$



Fases de
Ejecución

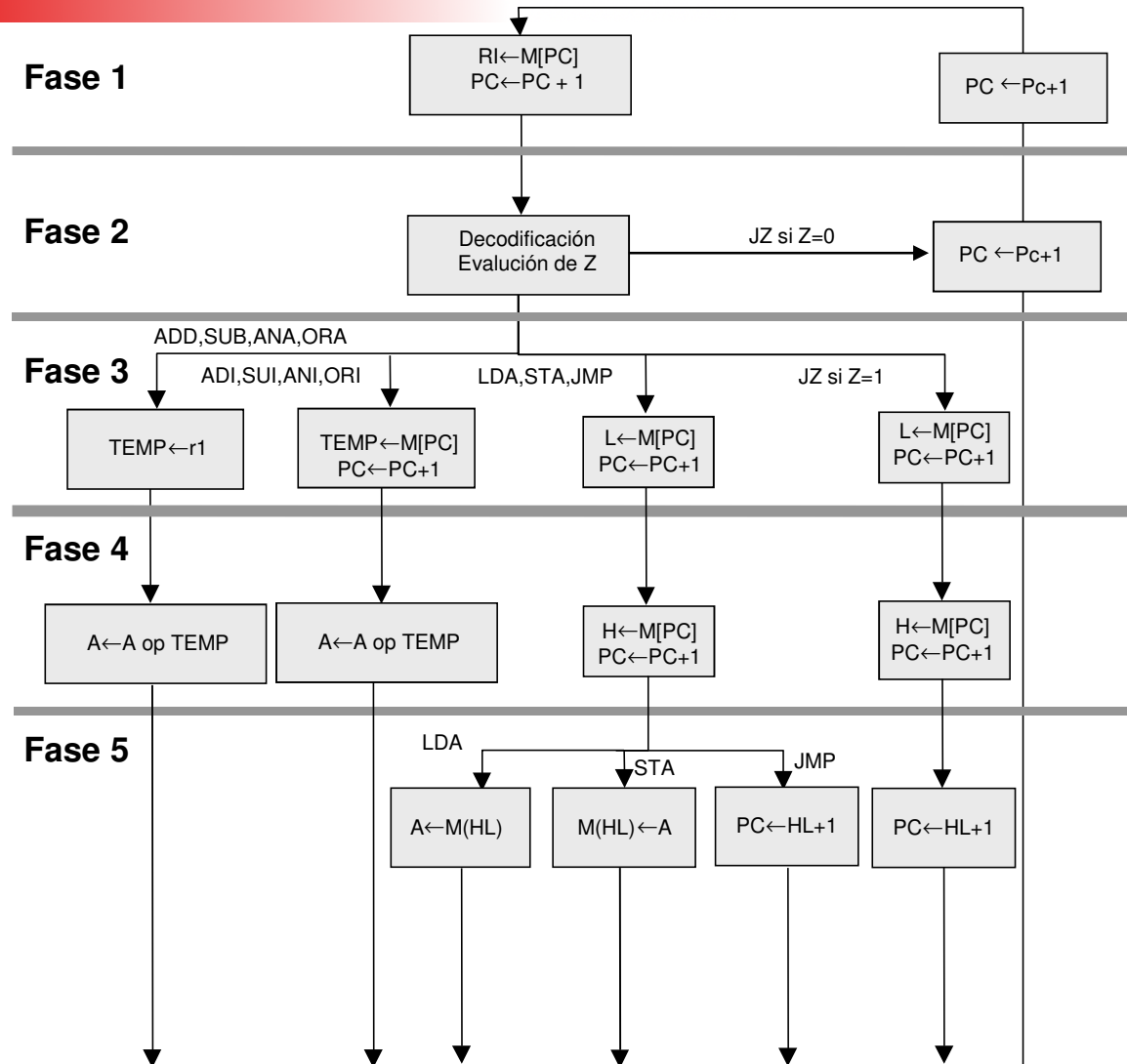
Establecimiento de las fases (5)

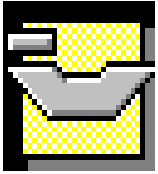
- ◆ **Fase 6: Conclusión de la instrucción de salto condicional.**
 - ◆ $PC \leftarrow HL + 1$



Fases de ejecución

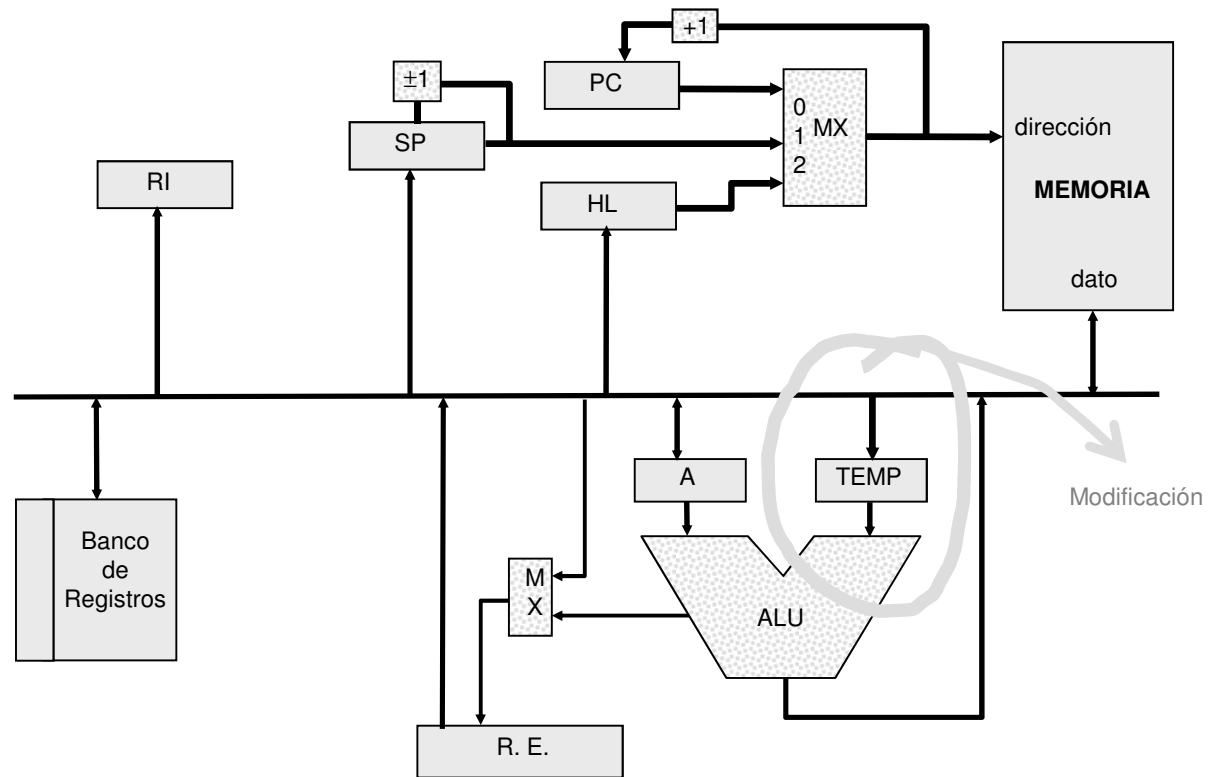
Diagrama de fases final

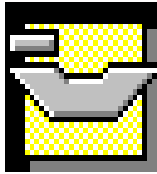




Fases de ejecución

La ruta de datos



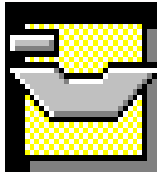


Unidad
de Control

Identificación de las señales de control⁽¹⁾

- ◆ Señales de control de salida para el banco de registros

Señal	Descripción
	CONTROL DE BANCO DE REGISTROS
SELreg1 SELreg0	Seleccionan el registro al que se desea acceder del banco de registros: B, C, D o E.
Lreg	Si se encuentra activada (valor 1), permite que el dato que se encuentra en el bus se almacene en el registro seleccionado.
Sreg	Si se encuentra activada (valor 1) permite que el dato que se encuentra en el registro seleccionado aparezca en el bus. En caso contrario



Unidad
de Control

Identificación de las señales de control⁽²⁾

- ◆ Señales de control de salida para la memoria y la ALU

Señal	Descripción
CONTROL DE LA MEMORIA	
Lmem	Señal de lectura de la memoria. Si su valor es 1, el dato almacenado en la posición de memoria que hay en el bus de direcciones se coloca en el bus de datos.
Emem	Señal de escritura de la memoria. Si su valor es 1, el dato que se encuentra en el bus de datos se almacena en la posición de memoria que hay en el bus de direcciones.
SDir2 SDir1	Señales de control al multiplexor que selecciona el dato a escribir en el bus de direcciones para acceder a una posición de memoria. Dependiendo de su valor se accederá a la posición de memoria especificada por el registro PC, SP o HL..
CONTROL DE LA UNIDAD ARITMÉTICA Y LÓGICA	
ALU2 ALU1 ALU0	Estas tres señales de control determinan la operación a realizar por la ALU: suma, resta, and, or , or exclusiva, incremento y decremento.
Salu	Si se encuentra activada permite que el resultado de la ALU aparezca en el bus de datos.



Unidad
de Control

Identificación de las señales de control⁽³⁾

- ◆ Señales de control de salida para el registro de estado

Señal	Descripción
	CONTROL DEL REGISTRO DE ESTADO
LF	Si se encuentra activa, carga los datos de la entrada en el registro de estado. El dato puede provenir de la ALU o del registro acumulador.
SF	Si se encuentra activa, escribe en el bus de datos el contenido del registro de estado.
SelO	Señal de control a un multiplexor que selecciona como dato a escribir en el registro de estado como flag de Overflow, el bit 2 del registro acumulador o el indicador de overflow procedente de la ALU.
SelC	Señal de control a un multiplexor que selecciona como dato a escribir en el registro de estado como flag de Carry, el bit 1 del registro acumulador o el indicador de Carry procedente de la ALU.
SelZ	Señal de control a un multiplexor que selecciona como dato a escribir en el registro de estado como flag de Cero, el bit 0 del registro acumulador o el indicador de Cero procedente de la ALU.

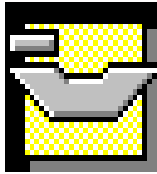


Unidad
de Control

Identificación de las señales de control⁽⁴⁾

- ◆ Señales de control de salida para los registros y señales de control de entrada.

Señal	Descripción
CONTROL DE REGISTROS	
Lri	Si está activa, el dato que se encuentra en el bus de datos se guardará en el registro de instrucción.
Lpc	Si esta activa, ordena la carga del registro Contador de Programa con el dato que se encuentra a su entrada.
LspL	Si está activa, ordena la carga de la parte baja del registro SP con el dato que se encuentra en el bus de datos.
LspH	Si está activa, ordena la carga de la parte alta del registro SP con el dato que se encuentra en el bus de datos.
Isp	Si está activa, incrementa el contenido del registro SP en una unidad.
Dsp	Si está activa, decrementa el contenido del registro SP en una unidad.
LdirL	Si está activa, ordena la carga de la parte baja del registro HL con el dato que se encuentra en el bus de datos.
LdirH	Si está activa, ordena la carga de la parte alta del registro HL con el dato que se encuentra en el bus de datos.
Lac	Si esta activa, el dato que se encuentra en el bus de datos se guardará en el registro acumulador.
Sac	Si está activa, el contenido del registro acumulador aparecerá en el bus de datos.
Ltemp	Si está activa, el dato que se encuentra en el bus de datos, se guardará en el registro temporal que se encuentra en la segunda entrada de la ALU.
ENTRADAS DE CONTROL	
Z	Representa el indicador Z generado por la ALU.
COP	Código de operación, es el contenido del registro de Instrucción (RI).



Unidad
de Control

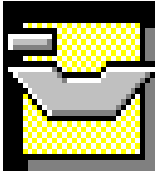
Identificación de las señales de control⁽⁵⁾

- ◆ Señales de control a los multiplexores, Banco de registros y la ALU.

SELreg1	SELreg0	Registro	SDir2	SDir1	Dirección
0	0	B	0	0	PC
0	1	C	0	1	SP
1	0	D	1	0	HL
1	1	E	1	1	No utilizada

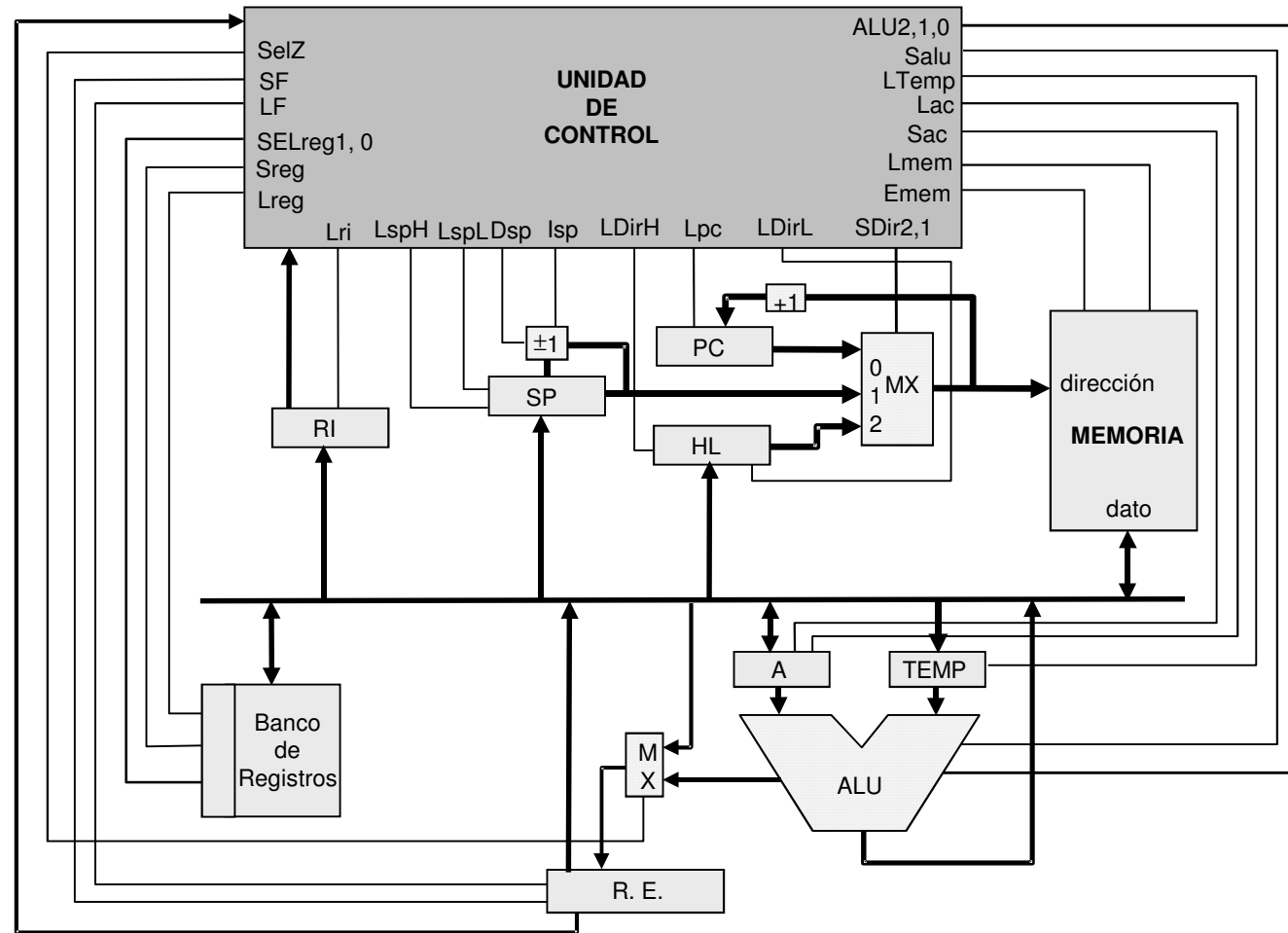
ALU2	ALU1	ALU0	Operación
0	0	0	Suma
0	0	1	Resta
0	1	0	And
0	1	1	Or
1	0	0	Xor
1	0	1	Incremento
1	1	0	Decremento
1	1	1	No utilizado

SelO	Acción	SelC	Acción	SelZ	Acción
0	$O \leftarrow ALU_O$	0	$C \leftarrow ALU_C$	0	$Z \leftarrow ALU_Z$
1	$O \leftarrow A_2$	1	$C \leftarrow A_1$	1	$Z \leftarrow A_0$



Unidad
de Control

Ruta de datos y control



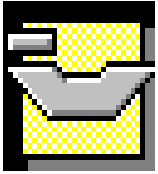


Unidad
de Control

Activación de las señales de control

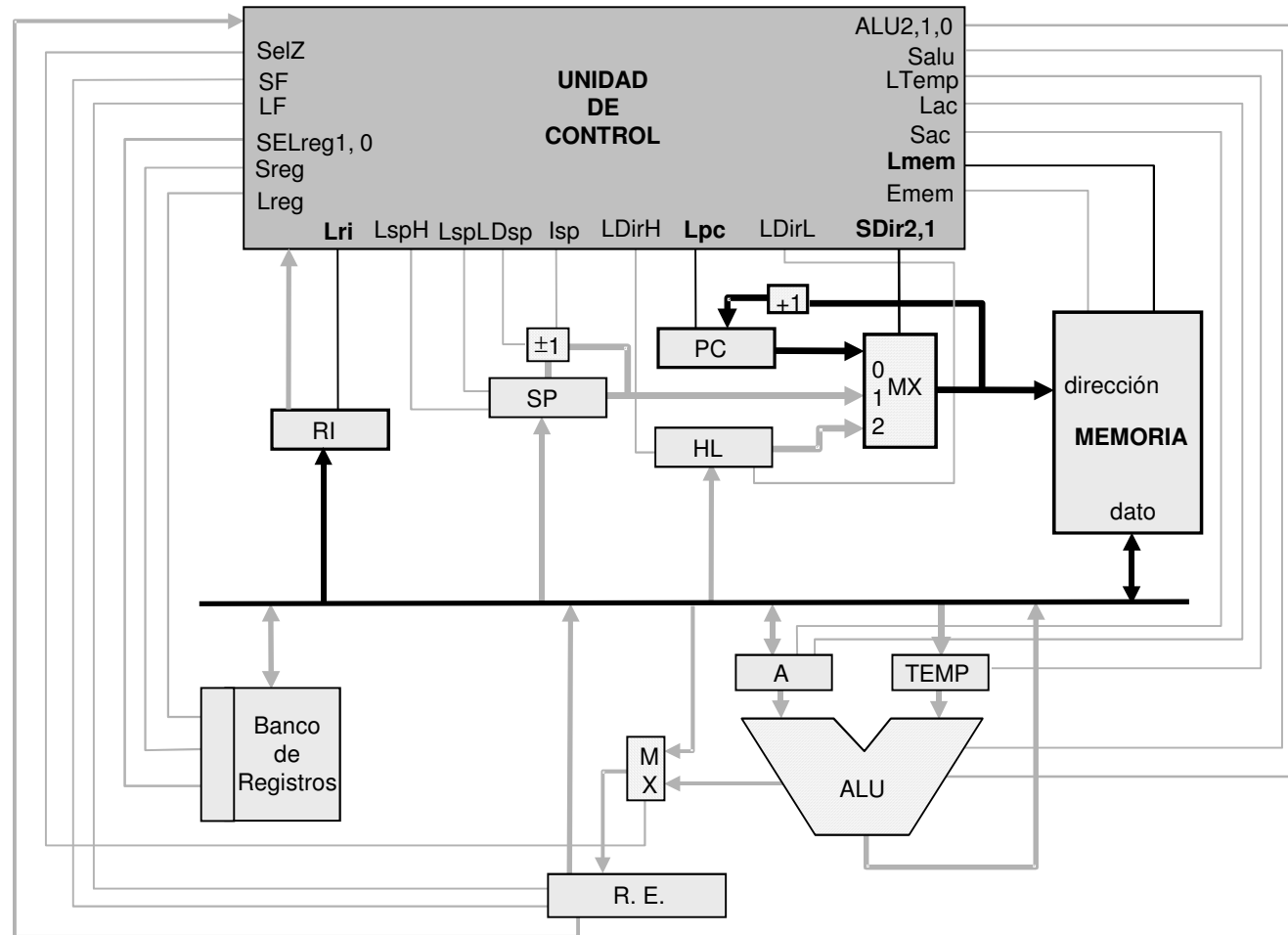
No se considerarán las instrucciones: ADD A, SUB A, ANA A y ORA A

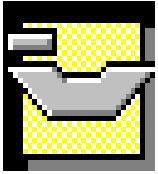
Fases	Operación	Activación de señales
FASE 1		
	$RI \leftarrow M[PC]$ $PC \leftarrow PC + 1$	SDir2, SDir1 (=00), Lmem, Lri, Lpc
FASE 2		
	Decodificación y evaluación de Z	
FASE 3		
ADD, SUB, ANA, ORA	$TEMP \leftarrow r1$	SELreg1, SELreg0 (=r1), Sreg, Ltemp
ADI, SUI, ANI, ORI	$TEMP \leftarrow M[PC]$ $PC \leftarrow PC + 1$	Ltemp, Lmem, SDir2, Sdir1 (=00) Lpc
LDA, STA, JMP, JZ	$L \leftarrow M[PC]$ $PC \leftarrow PC + 1$	LdirL, Lmem, SDir2, Sdir1 (=00) Lpc
FASE 4		
Aritmético-Lógicas	$A \leftarrow A \text{ op } TEMP$	ALU2, ALU1, ALU0 (=operación), Salu, Lac
Transferencia y salto	$H \leftarrow M[PC]$ $PC \leftarrow PC + 1$	LdirH, Lmem, SDir2, Sdir1 (=00) Lpc
FASE 5		
LDA	$A \leftarrow M(HL)$	SDir2, SDir1 (=10), Lmem, Lac
STA	$M(HL) \leftarrow A$	SDir2, Sdir1 (=10), Emem, Sac
JMP, JZ	$PC \leftarrow HL + 1$	Sdir2, Sdir1 (=10), Lpc



Unidad
de Control

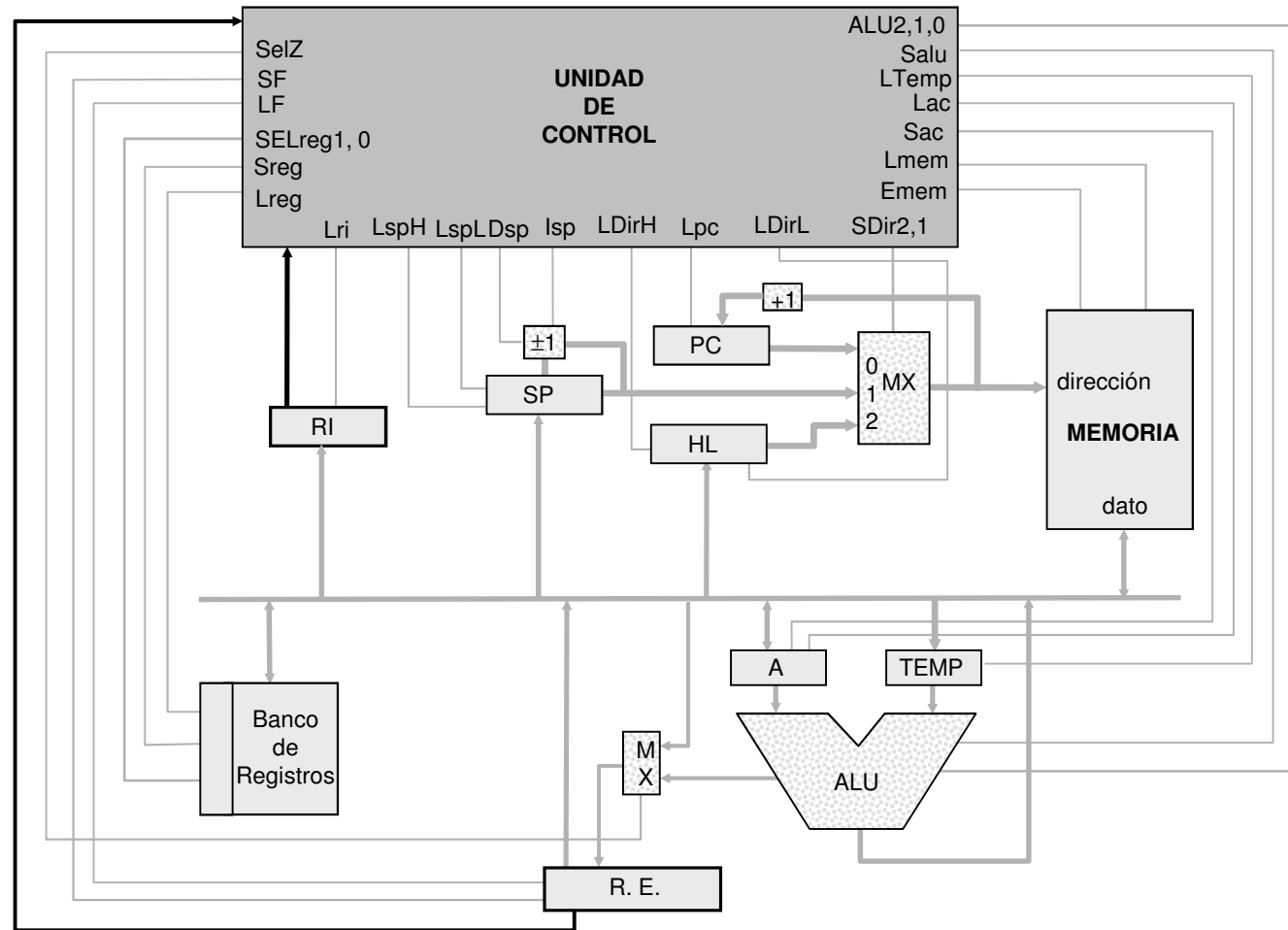
Fase 1 en la ejecución de LDA

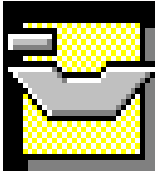




Unidad
de Control

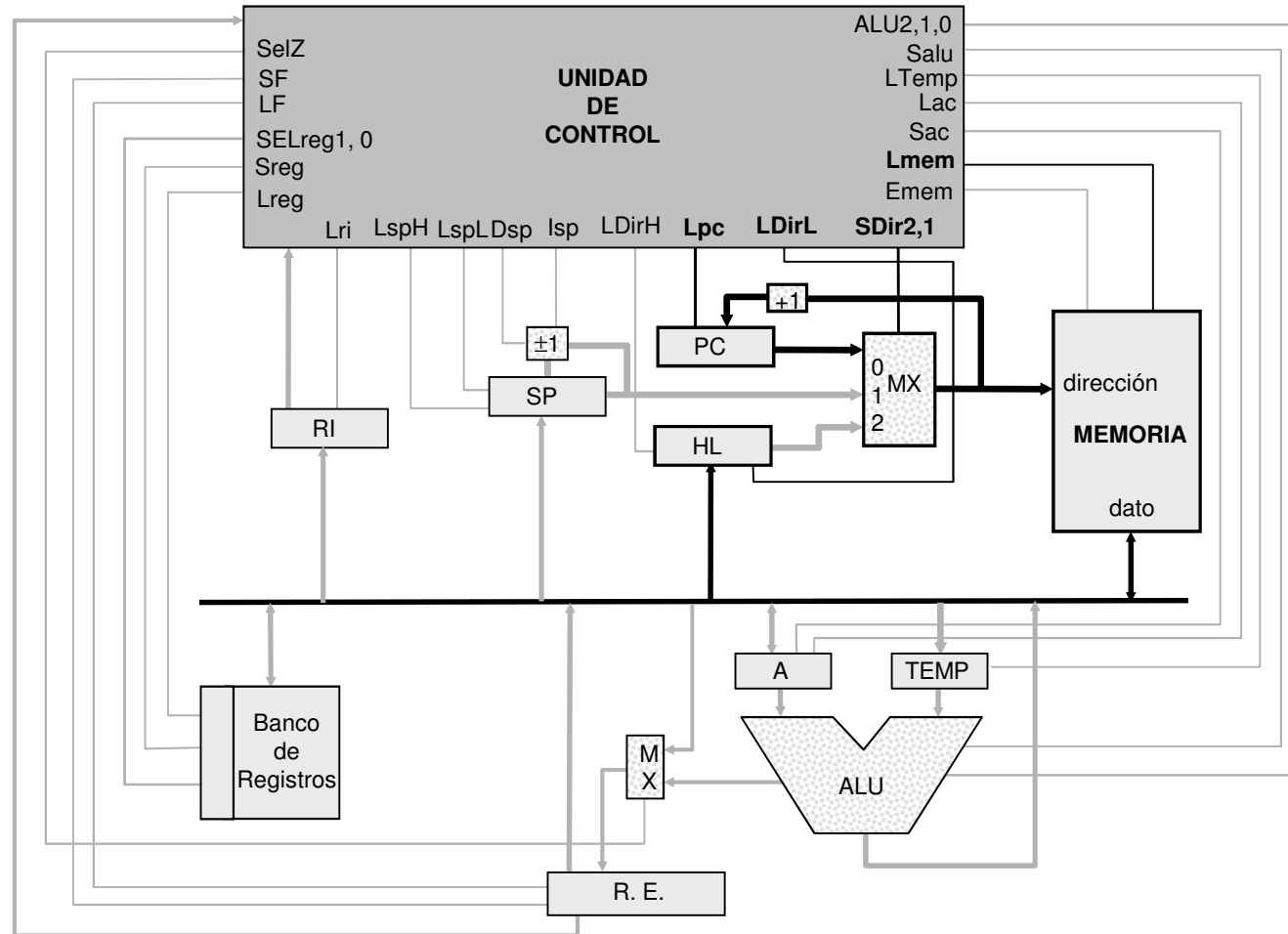
Fase 2 en la ejecución de LDA

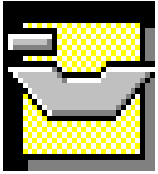




Unidad
de Control

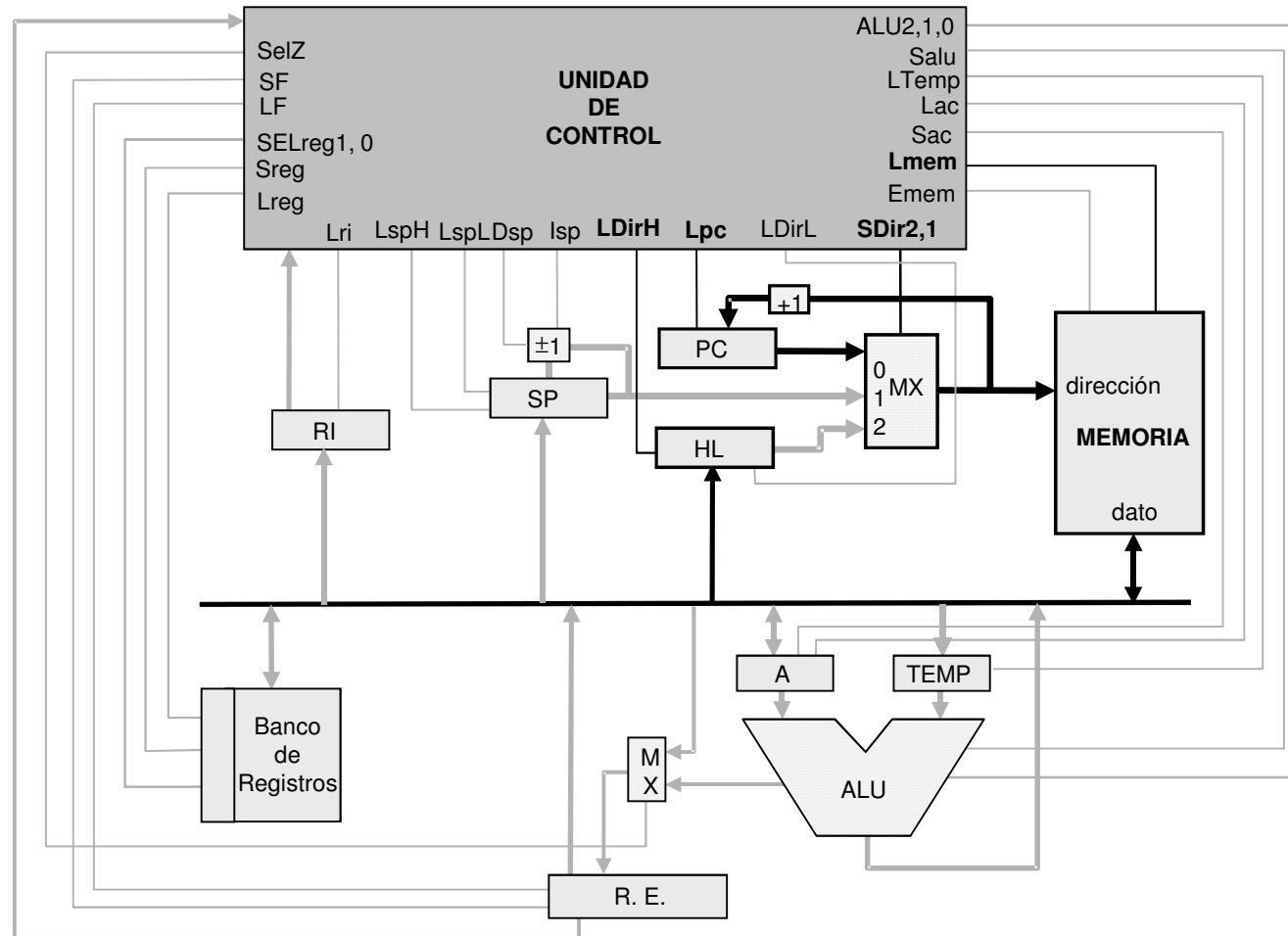
Fase 3 en la ejecución de LDA

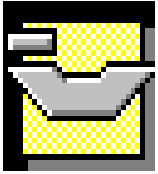




Unidad
de Control

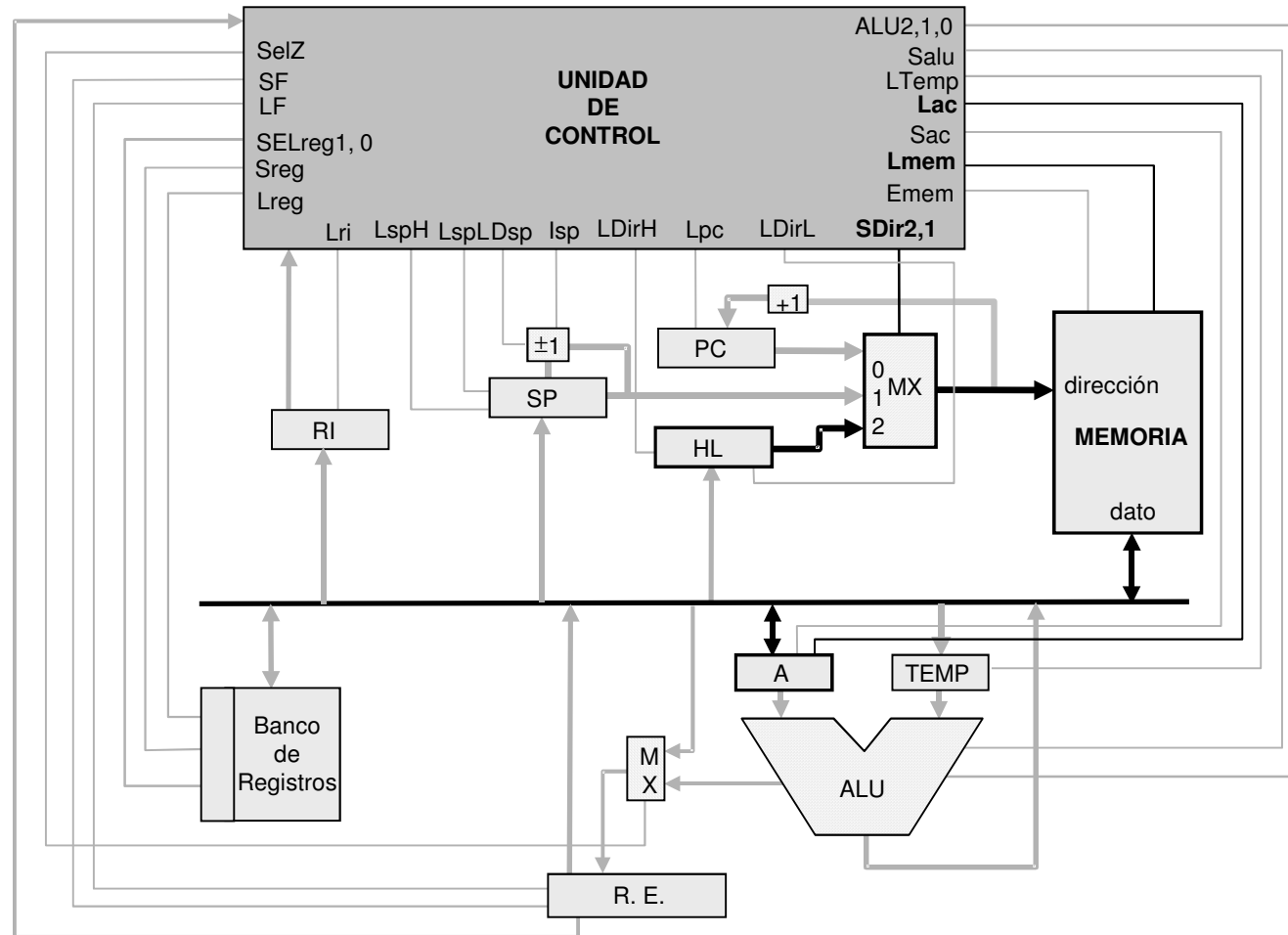
Fase 4 en la ejecución de LDA

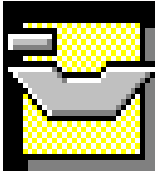




Unidad
de Control

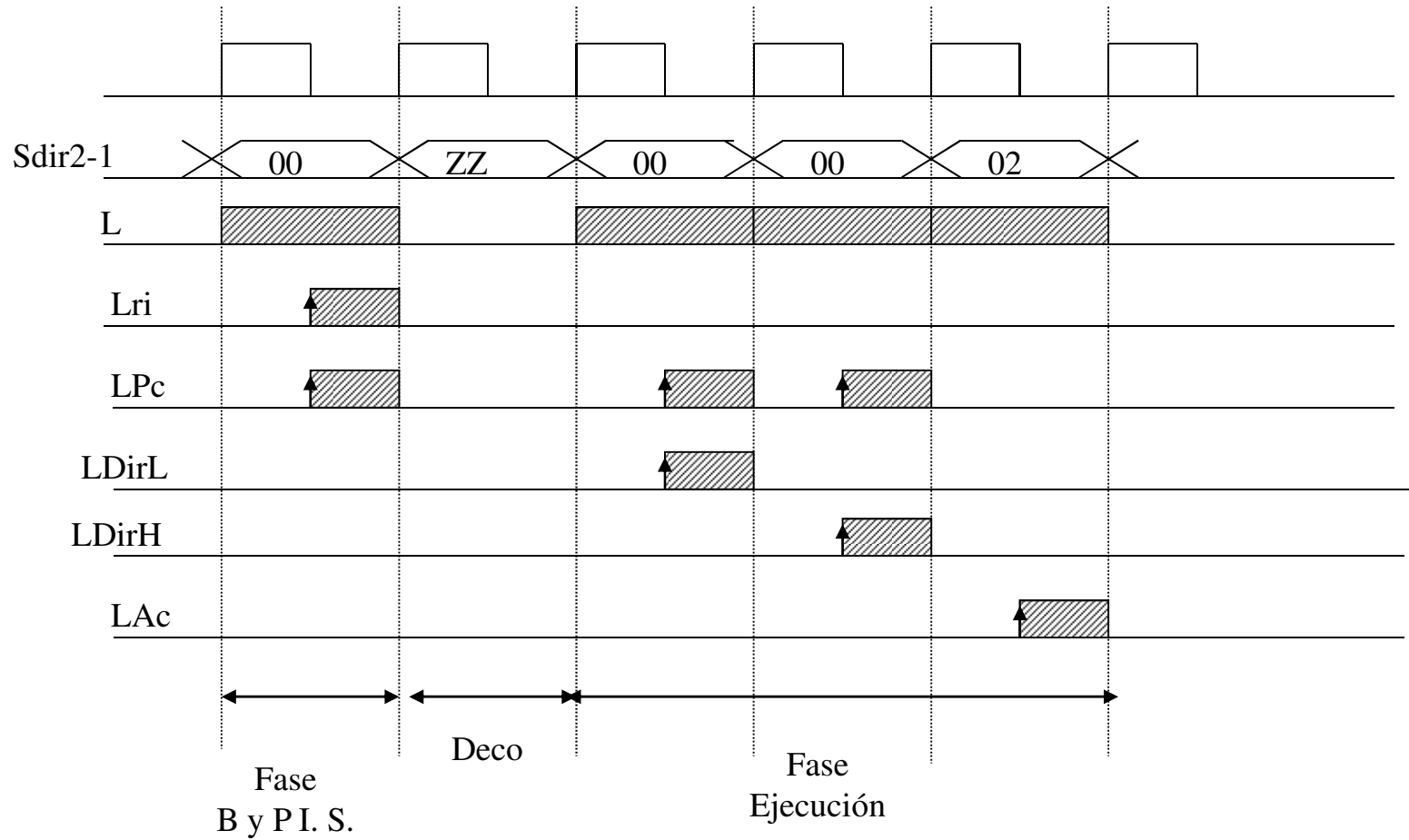
Fase 5 en la ejecución de LDA





Unidad de Control

LDA dir





Diseño de
la Unidad
de Control

Diseño de la Unidad de Control

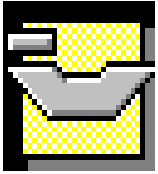
- ◆ **Control Cableado.**
 - ◆ Implementación en Hardware, por lo tanto muy rápido.
 - ◆ No es flexible: una modificación posterior implica cambiar el circuito entero.
- ◆ **Control Microprogramado.**
 - ◆ Representación programada para el control.
 - ◆ Más lento, al tener que acceder a la memoria de control.
 - ◆ Flexible, permite cambios posterior sin tener que modificar el circuito entero.
- ◆ **Dos métodos para el diseño de la Unidad de Control Cableada**
 - ◆ Método de la tabla de estados
 - ◆ Método del contador de secuencia



Unidad
de Control
Cableada

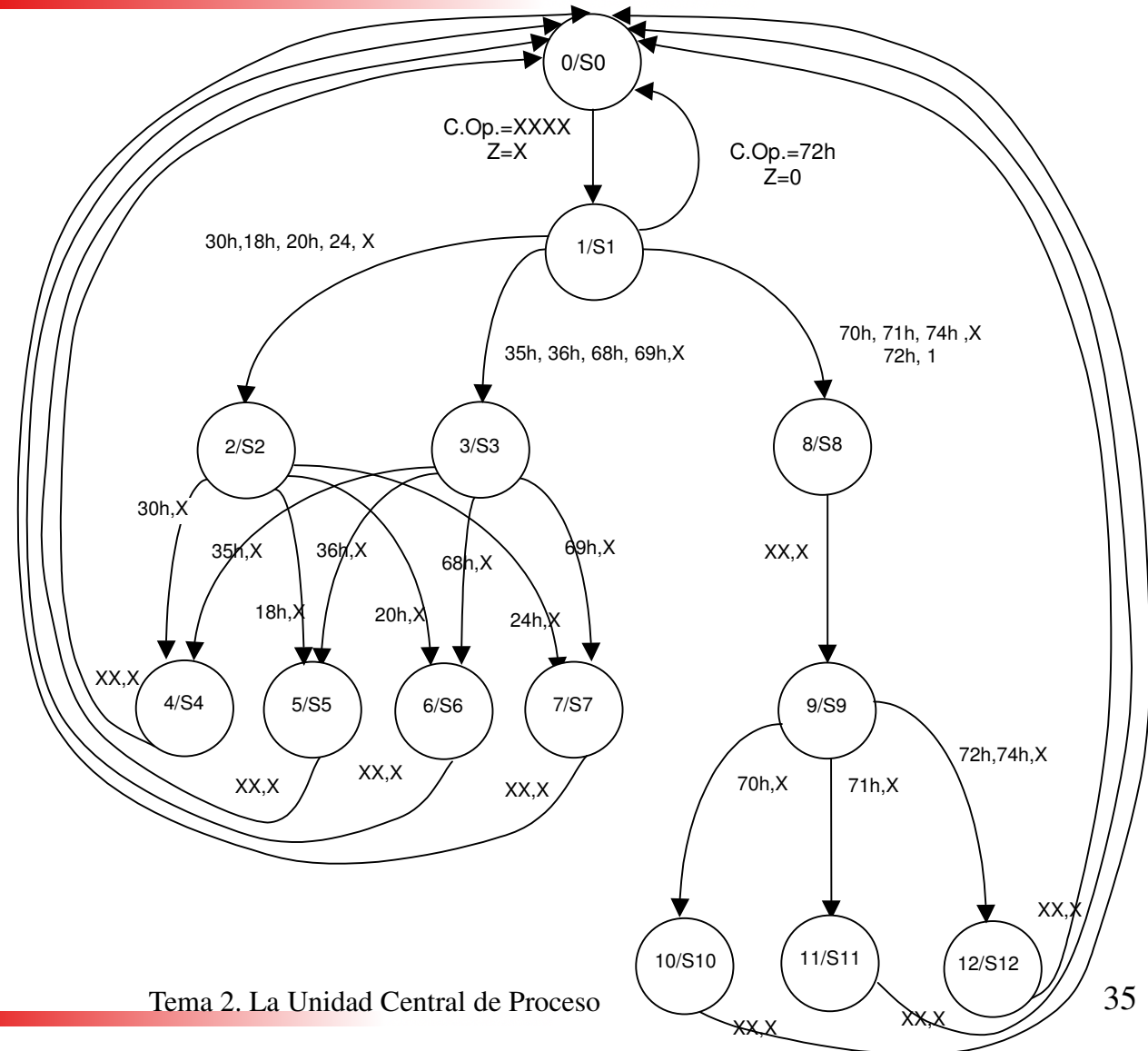
Método de la tabla de estados

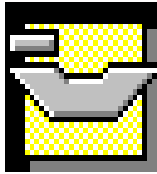
- ◆ Basada en una máquina de estados finitos.
- ◆ Una máquina de estados finitos consta:
 - ◆ Memoria Interna que contiene el estado y
 - ◆ Dos funciones combinatoriales:
 - ◆ La función de estado siguiente
 - ◆ La función de salida
 - ◆ Cada estado corresponde a un ciclo de reloj y contiene las operaciones a realizar en ese ciclo.
 - ◆ La función de estado siguiente es una función combinatorial que a partir de las entradas y el estado actual determina el estado siguiente.
 - ◆ La función de salida produce el conjunto de señales de control a partir de sus entradas y el estado actual.



Método
de la tabla
de estados

Grafo de estados



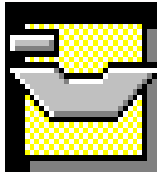


Método
de la tabla
de estados

Función de salida

Como instrucciones Aritmético-Lógicas se consideran únicamente: Add B, SUB B, ANA B y ORA B

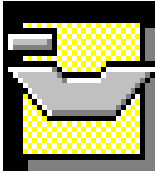
Función salida	Estados												
	0	1	2	3	4	5	6	7	8	9	10	11	12
Lri	1	0	0	0	0	0	0	0	0	0	0	0	0
SDir2,1	00	XX	XX	00	XX	XX	XX	XX	00	00	10	10	10
Lmem	1	0	0	1	0	0	0	0	1	1	1	0	0
Emem	0	0	0	0	0	0	0	0	0	0	0	1	0
Lpc	1	0	0	1	0	0	0	0	1	1	0	0	1
LspL	0	0	0	0	0	0	0	0	0	0	0	0	0
LspH	0	0	0	0	0	0	0	0	0	0	0	0	0
Isp	0	0	0	0	0	0	0	0	0	0	0	0	0
Dsp	0	0	0	0	0	0	0	0	0	0	0	0	0
LdiirL	0	0	0	0	0	0	0	0	1	0	0	0	0
LdirH	0	0	0	0	0	0	0	0	0	1	0	0	0
SELreg1,0	XX	XX	00	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX
Lreg	0	0	0	0	0	0	0	0	0	0	0	0	0
Sreg	0	0	1	0	0	0	0	0	0	0	0	0	0
Lac	0	0	0	0	1	1	1	1	0	0	1	0	0
Sac	0	0	0	0	0	0	0	0	0	0	0	1	0
Ltemp	0	0	1	1	0	0	0	0	0	0	0	0	0
ALU2,1,0	XXX	XXX	XXX	XXX	000	001	010	011	XXX	XXX	XXX	XXX	XXX
Salu	0	0	0	0	1	1	1	1	0	0	0	0	0
Sel0	X	X	X	X	0	0	0	0	X	X	X	X	X
SelC	X	X	X	X	0	0	0	0	X	X	X	X	X
SelZ	X	X	X	X	0	0	0	0	X	X	X	X	X
LF	0	0	0	0	1	1	1	1	0	0	0	0	0
SF	0	0	0	0	0	0	0	0	0	0	0	0	0



Método
de la tabla
de estados

Función de estado siguiente

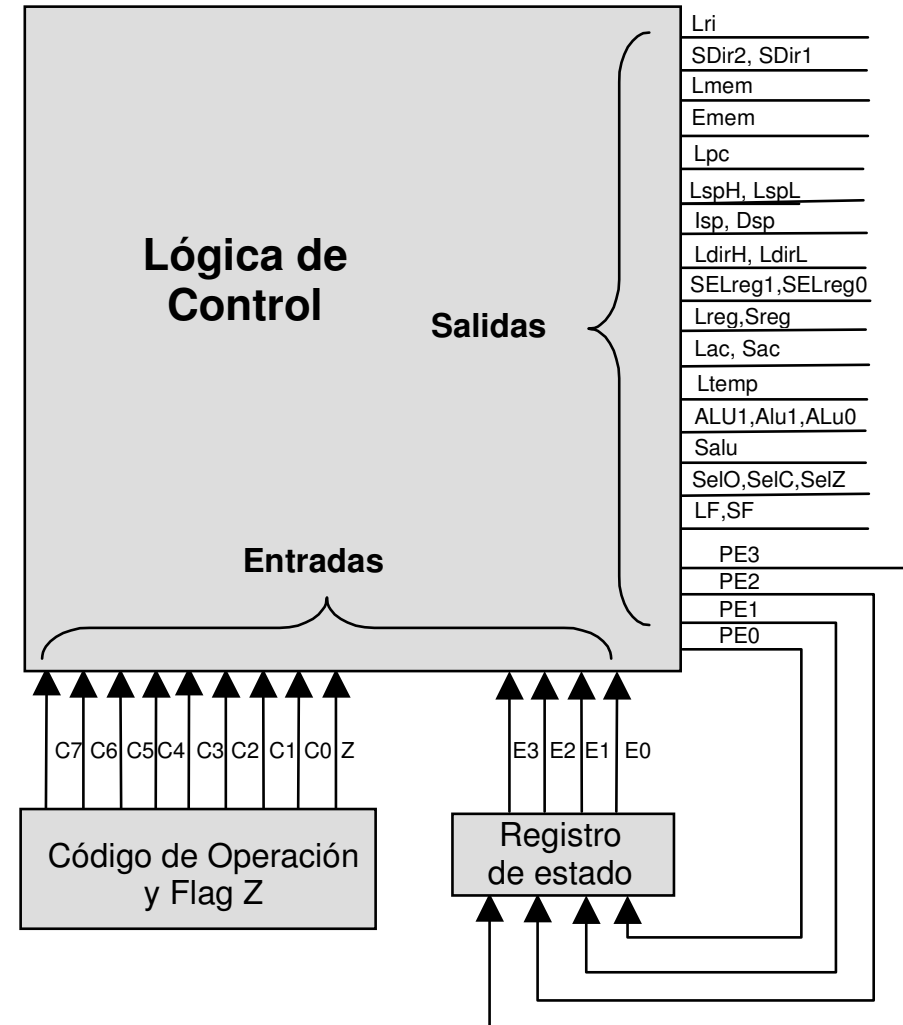
INSTRUCCIÓN	E3	E2	E1	E0	PE3	PE2	PE1	PE0
	0	0	0	0	0	0	0	1
72h,0 (JZ)	0	0	0	1	0	0	0	0
30h,18h,20h,24h,X (ADD,SUB,ANA,ORA)	0	0	0	1	0	0	1	0
35h,36h,68h,69h,X (ADI,SUI,ANI,ORI)	0	0	0	1	0	0	1	1
70h,71h,74h,X (LDA,STA,JMP);72h,1 (JZ)	0	0	0	1	1	0	0	0
30h,X (ADD)	0	0	1	0	0	1	0	0
18h,X (SUB)	0	0	1	0	0	1	0	1
20h,X (ANA)	0	0	1	0	0	1	1	0
24h,X (ORA)	0	0	1	0	0	1	1	1
35h,X (ADI)	0	0	1	1	0	1	0	0
36h,X (SUI)	0	0	1	1	0	1	0	1
68h,X (ANI)	0	0	1	1	0	1	1	0
69h,X (ORI)	0	0	1	1	0	1	1	1
	0	1	0	0	0	0	0	0
	0	1	0	1	0	0	0	0
	0	1	1	0	0	0	0	0
	0	1	1	1	0	0	0	0
	1	0	0	0	1	0	0	1
70h,X (LDA)	1	0	0	1	1	0	1	0
71h,X (STA)	1	0	0	1	1	0	1	1
72h,74h , X (JMP, JZ)	1	0	0	1	1	1	0	0
	1	0	1	0	0	0	0	0
	1	0	1	1	0	0	0	0
	1	1	0	0	0	0	0	0

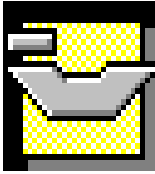


Método de la tabla de estados

Implementación de la Unidad de Control

- ◆ Unida de control construida como una máquina de estados finitos.
- ◆ El circuito combinacional podría implementarse mediante una ROM o una PLA.

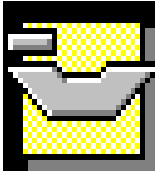




Método
de la tabla
de estados

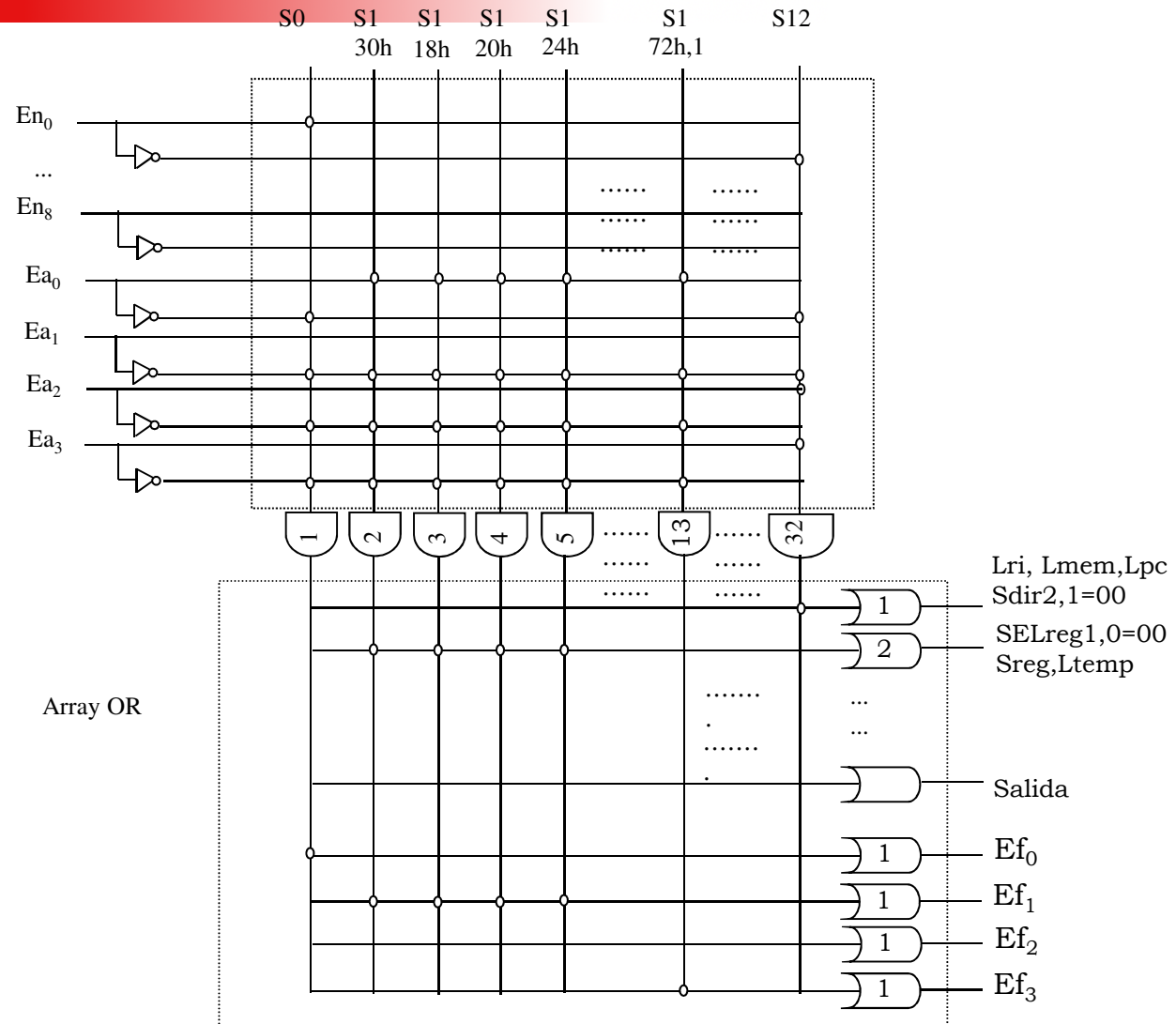
Implementación de la Unidad de Control

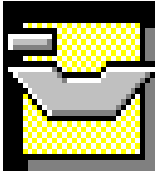
- ◆ Un PLA (array lógico programable) es un elemento lógico programable de propósito general para implementar cualquier función combinacional.
- ◆ Un PLA está formado por:
 - Un conjunto de entradas y salidas
 - Un conjunto de puertas AND que forma un conjunto de términos productos.
 - Un conjunto de puertas OR cada uno de los cuales forma una suma lógica de cualquiera de los términos producto.
 - Un conjunto de inversores para las entradas
 - Dos matrices de conexión modificables (AND y OR)



Método
de la tabla
de estados

Implementación de la Unidad de Control





Método
de la tabla
de estados

Implementación de la Unidad de Control

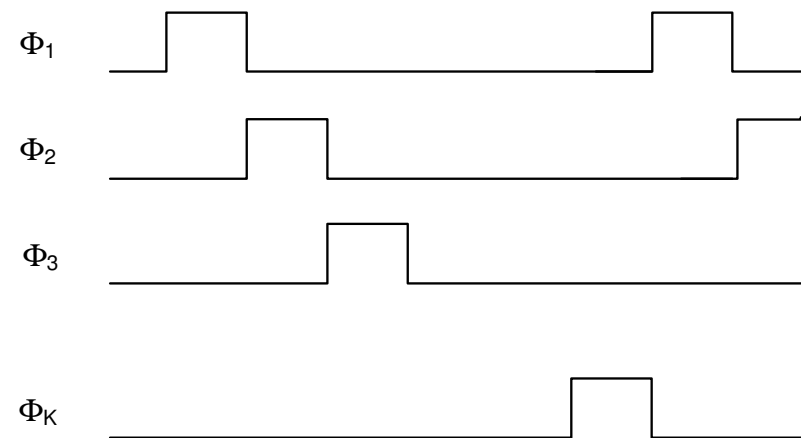
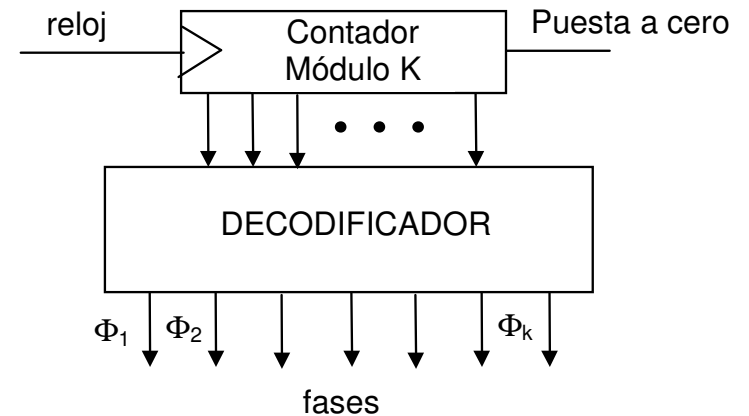
- ◆ El tamaño de un PLA es igual a la suma del tamaño del array de puertas AND y el tamaño del array de puertas OR.
- ◆ Para MaNoTaS = (13x número diferentes términos producto) + (32x número términos suma).
- ◆ Un PLA es más eficiente que una ROM ya que en lugar de almacenar la tabla de verdad completa, realiza una suma de productos mínima.
- ◆ La programación de un PLA es más difícil que una ROM.

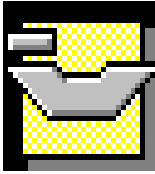


Unidad
de Control
Cableada

Método del contador de secuencia

- ◆ Basado en un contador módulo K cuyas salidas se conectan a un decodificador generándose una señal individual por cada fase.
- ◆ El contador atraviesa cíclicamente sus K estados.
- ◆ El decodificador genera K señales pulsadas $\{\Phi_j\}$ que son las señales de fase.
- ◆ Todas las fases tienen la misma duración igual a un ciclo de reloj.

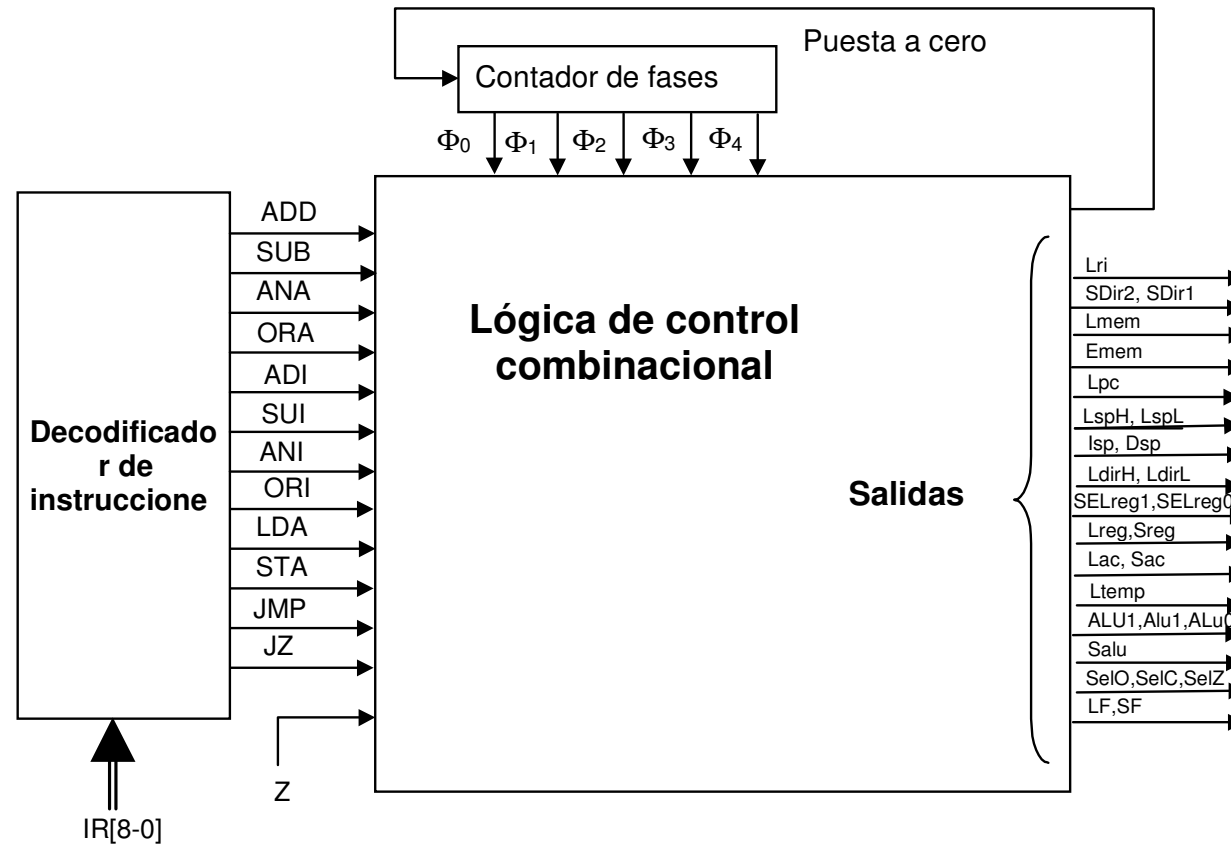


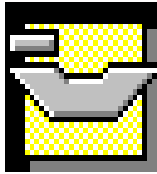


Método del Contador de Fases

Implementación de la Unidad de Control

- ◆ Cada señal de control se obtiene de:
$$C_i = \sum_j \Phi_j \sum_m I_m S_l$$



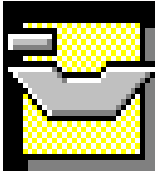


Método
del Contador
de Fases

Ejemplos de funciones lógicas

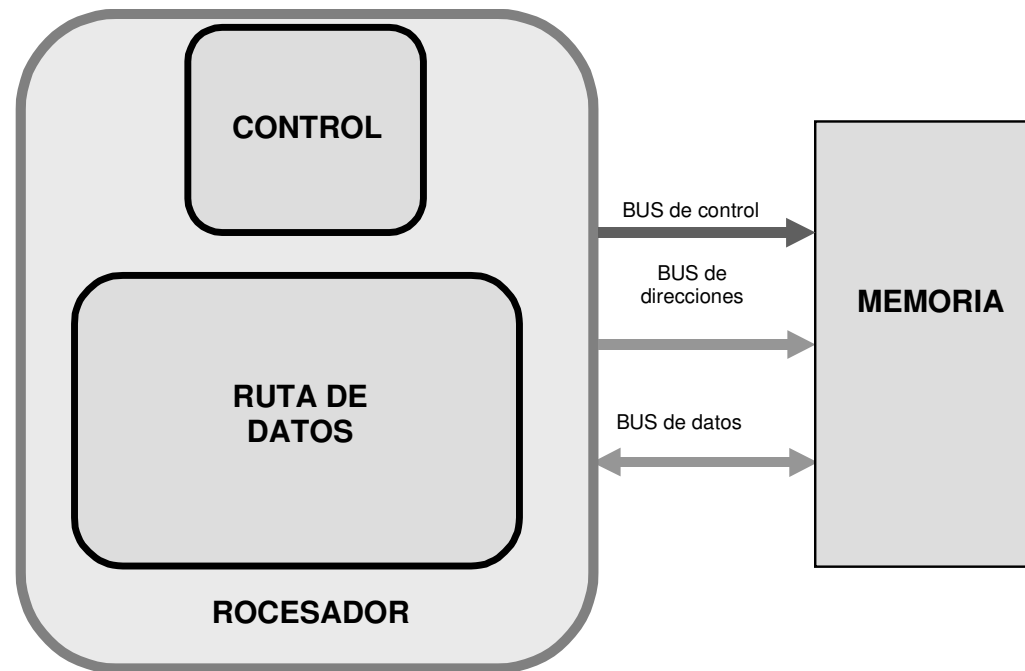
- ◆ Funciones lógicas de salida para algunas señales de control.

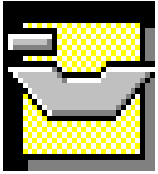
Señal	Función lógica de salida
Puesta a cero	$\Phi_{14} \cdot (JZ \cdot \bar{Z}) + \Phi_4 \cdot (\text{add} + \text{adi}) + \Phi_5 \cdot (\text{sub} + \text{sui}) + \Phi_6 \cdot (\text{ana} + \text{ani}) + \Phi_{10} \cdot \text{lda} + \Phi_{11} \cdot \text{sta} + \Phi_{12} \cdot (\text{jmp} + JZ \cdot Z)$
Lpc	$\Phi_0 + \Phi_2 \cdot (\text{adi} + \text{sui} + \text{ani} + \text{ori} + \text{lda} + \text{sta} + \text{jmp} + jz \cdot Z)$
ALU1	$\Phi_3 \cdot (\text{ana} + \text{ani} + \text{ora} + \text{ori})$
ALU0	$\Phi_3 \cdot (\text{sub} + \text{sui} + \text{ora} + \text{ori})$
Emem	$\Phi_4 \cdot \text{sta}$



MaNoTas

Abstracción de la máquina





Conclusiones

Conclusiones

- ◆ La estructura de la ruta de datos influye sobre el establecimiento de las fases.
- ◆ La estructura del repertorio de instrucciones influye sobre la ruta de datos.
- ◆ La Unidad de Control cableada es muy eficiente puesto que las señales de control se activan directamente por Hardware.
 - ◆ Es adecuada cuando el conjunto de instrucciones no es demasiado complejo.
 - ◆ Tiene la desventaja de ser poco flexible ante modificaciones posteriores.
- ◆ Los métodos de diseño de la UC estudiados son equivalentes, diferenciándose en la mayor o menor facilidad de obtener las funciones lógicas de las señales de control.