



CAPÍTULO 1: ARQUITECTURA GENERAL

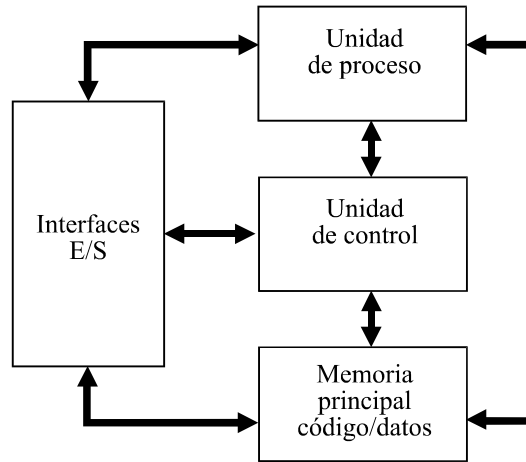


Fig.1.1. Arquitectura de un ordenador según Von Neumann.

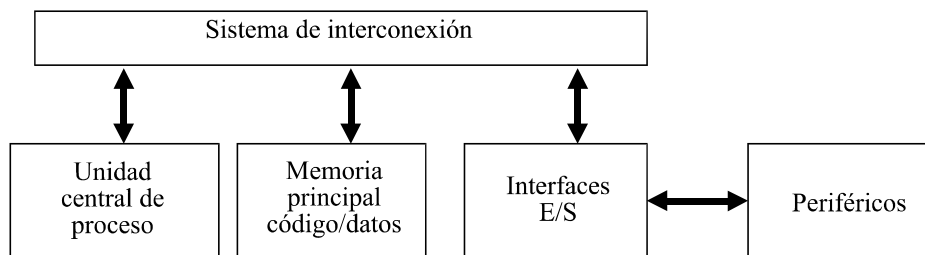


Fig.1.2. Organigrama de bloques de un ordenador, con interfaces a periféricos.

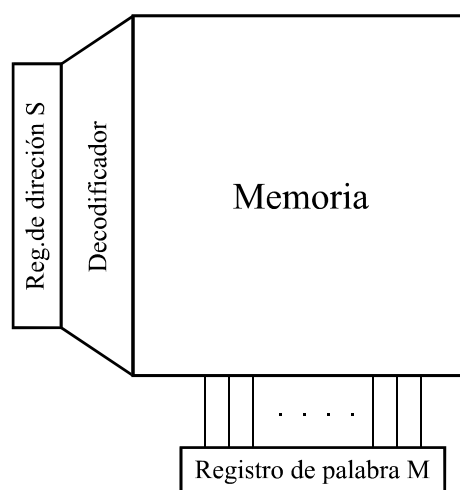


Fig.1.3. Esquema de una memoria principal de un ordenador.

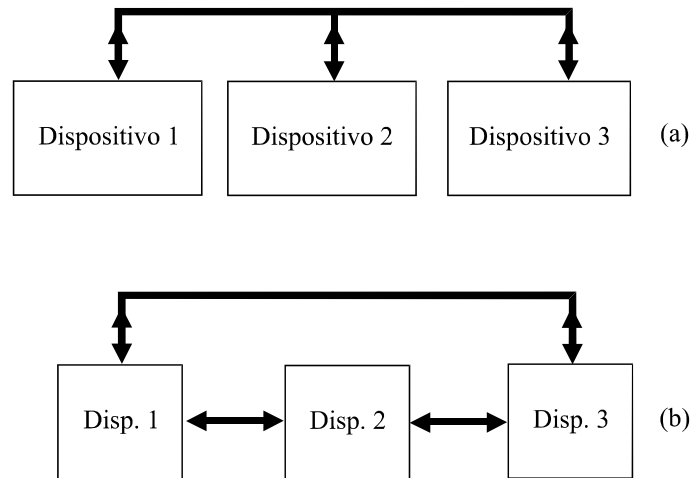


Fig.1.4. Configuraciones de buses.

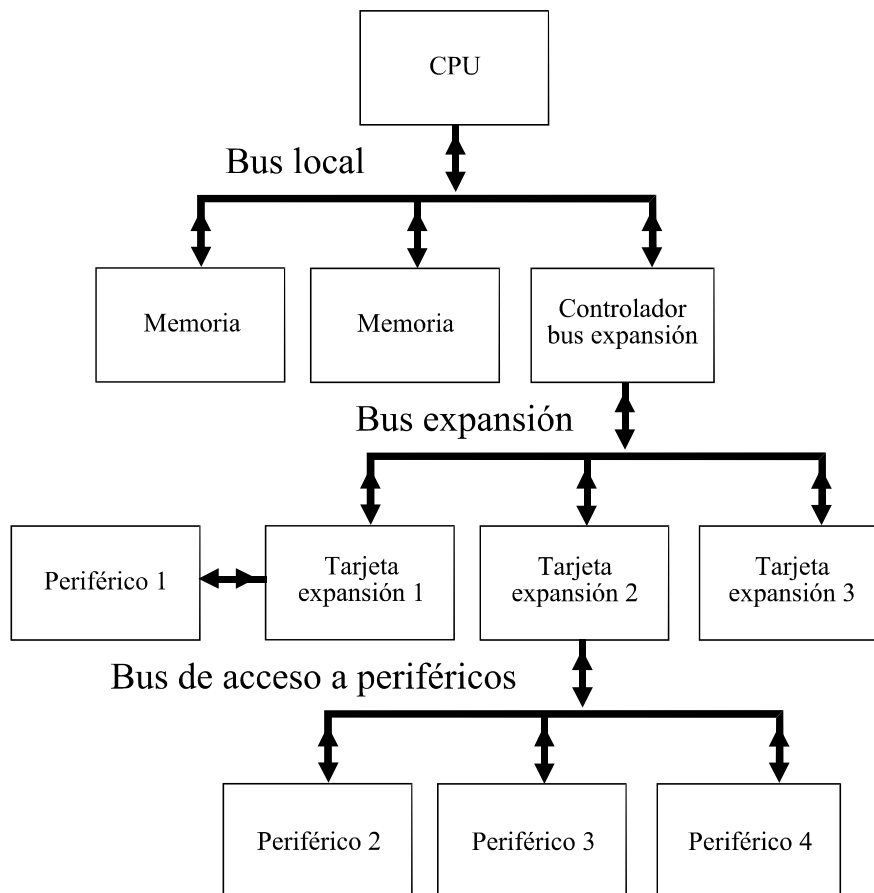


Fig.1.5. Configuración de varios buses jerarquizados.



CAPÍTULO 2: MEMORIAS

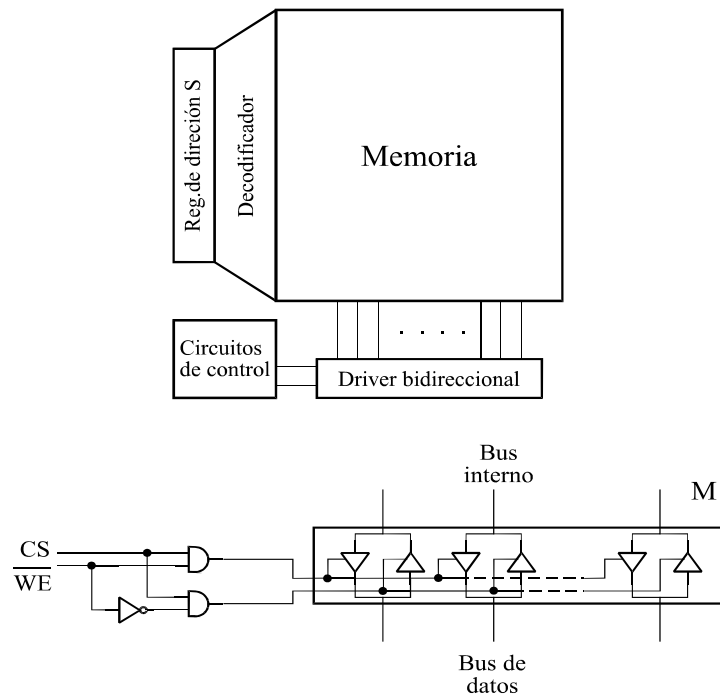


Fig.2.1. Memoria de acceso aleatorio con circuito de control y driver bidireccional.

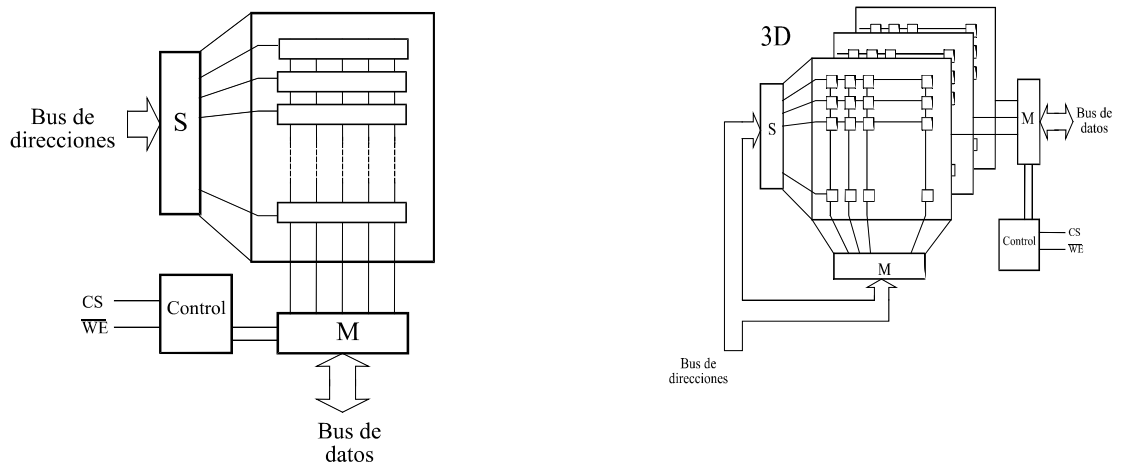


Fig.2.2. Estructura de memorias 2D y 3D.

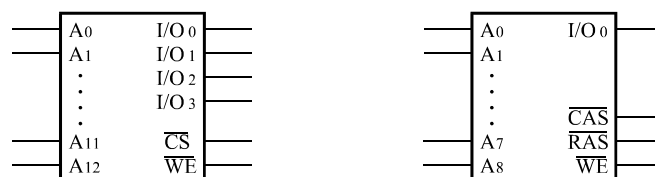


Fig.2.3. CIs de RAM estática y dinámica.

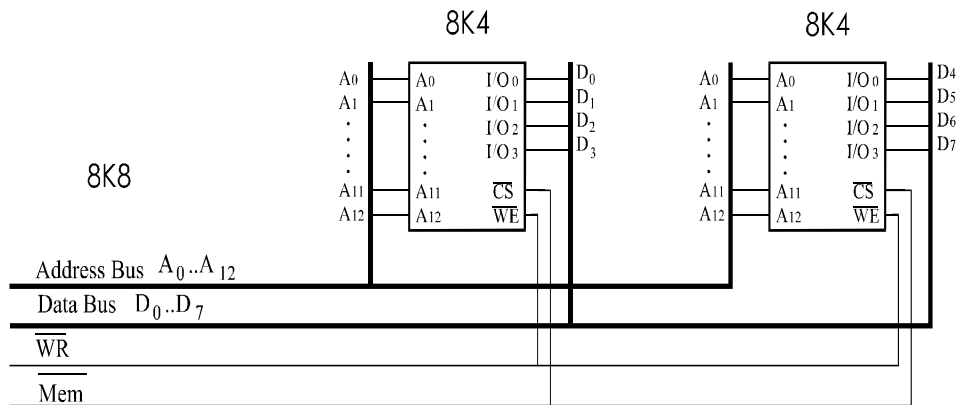


Fig.2.4. Incremento de la longitud de palabra.

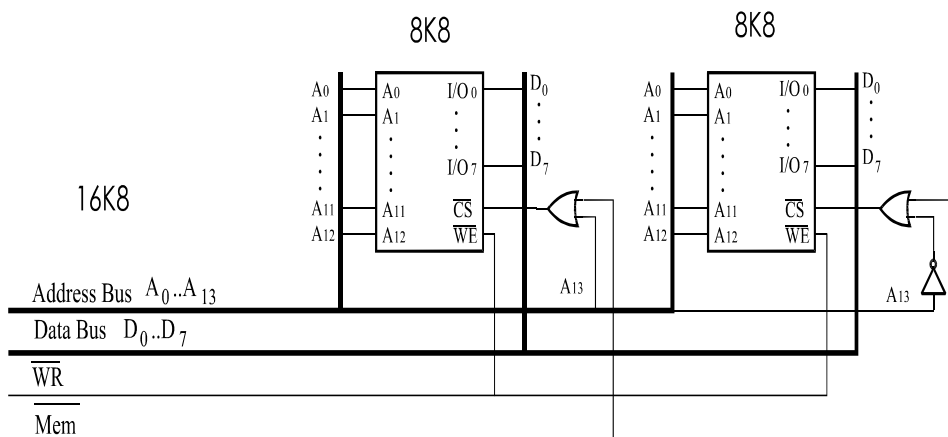


Fig.2.5. Incremento del número de palabras.

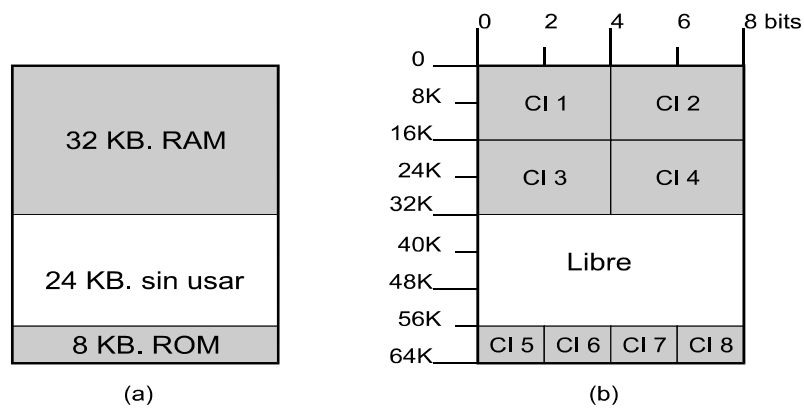
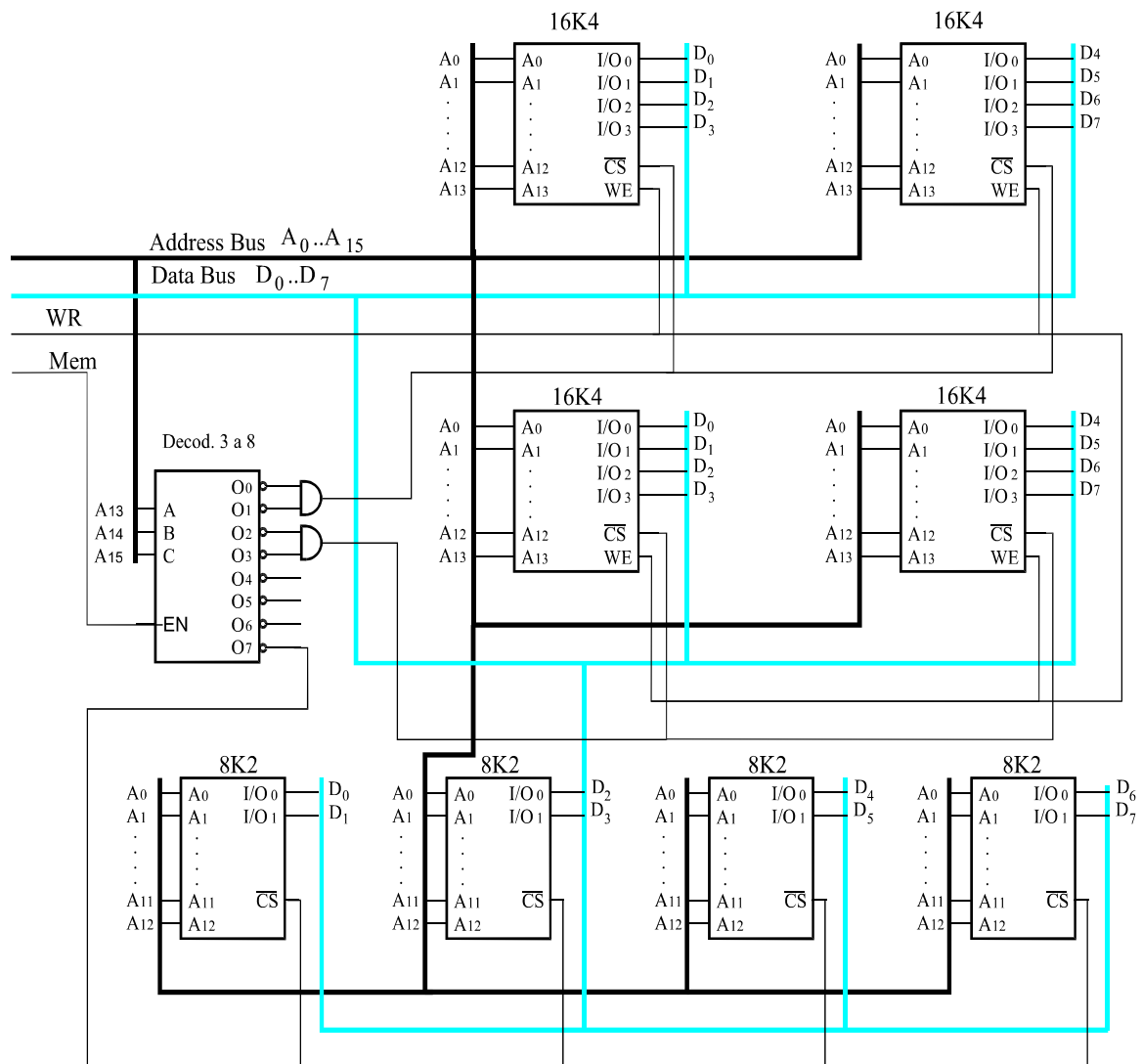


Fig.2.6. Mapa de memoria y distribución de CIs



Otra solución con un decod. 2 a 4.

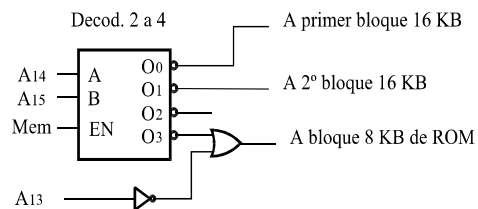


Fig.2.7. Incremento del número y tamaño de palabras.

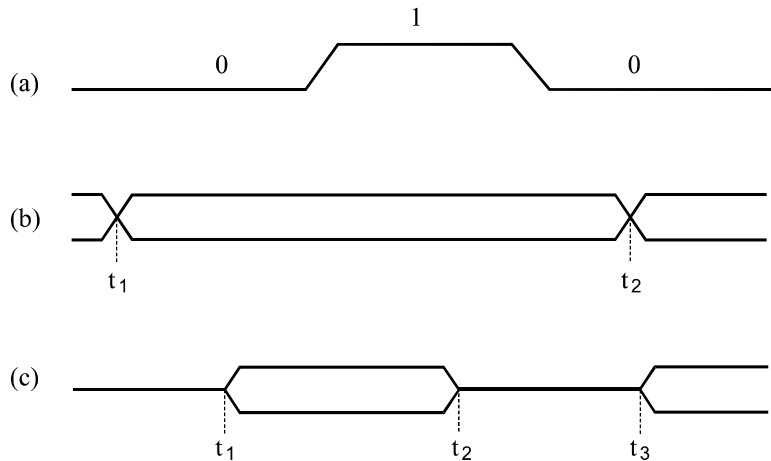


Fig.2.8. Signos convencionales de los cronogramas.

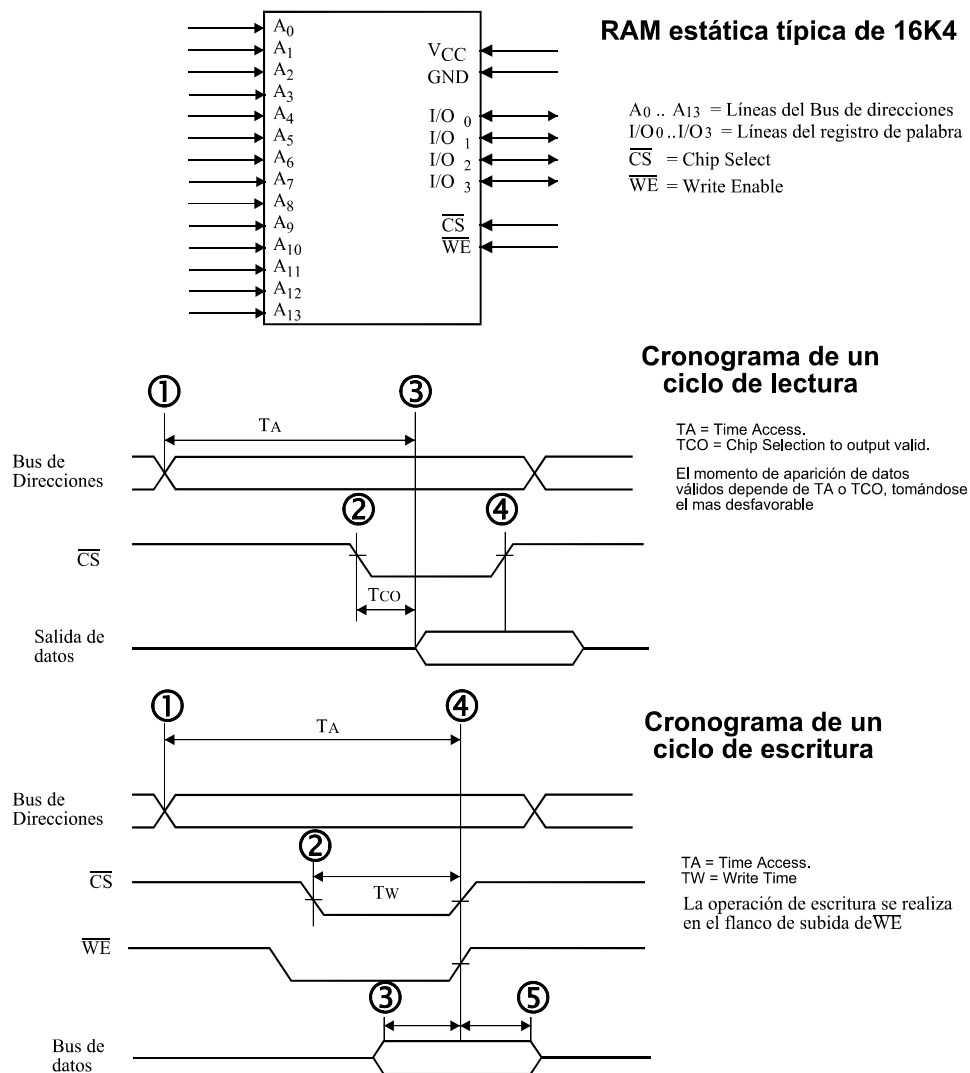


Fig.2.9. Cronograma de acceso a una memoria RAM estática.

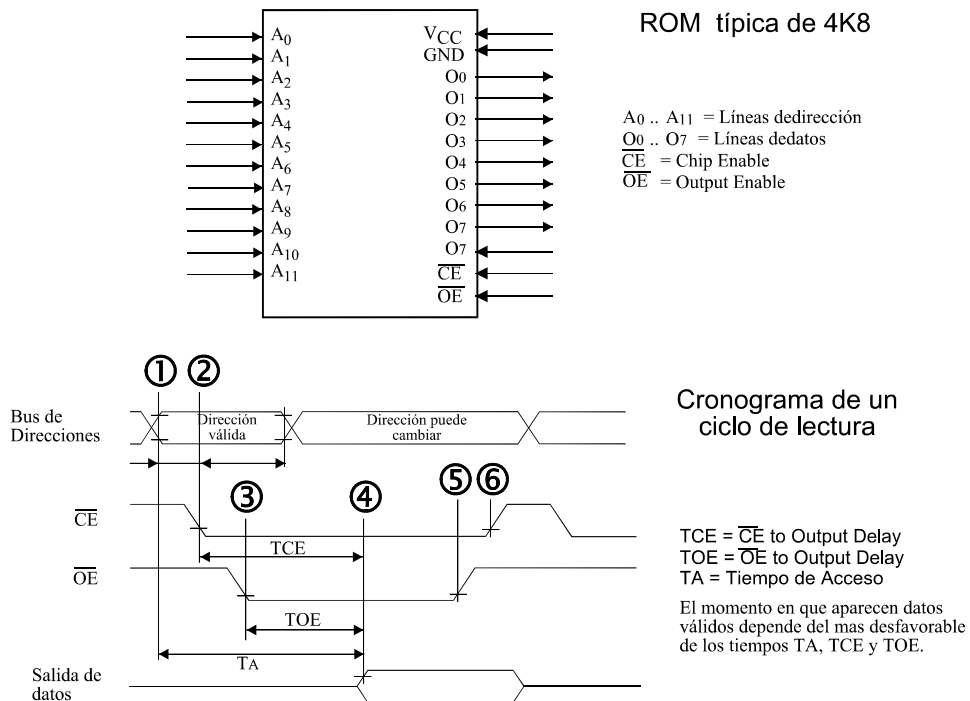


Fig.2.10. Cronograma de acceso a una memoria ROM

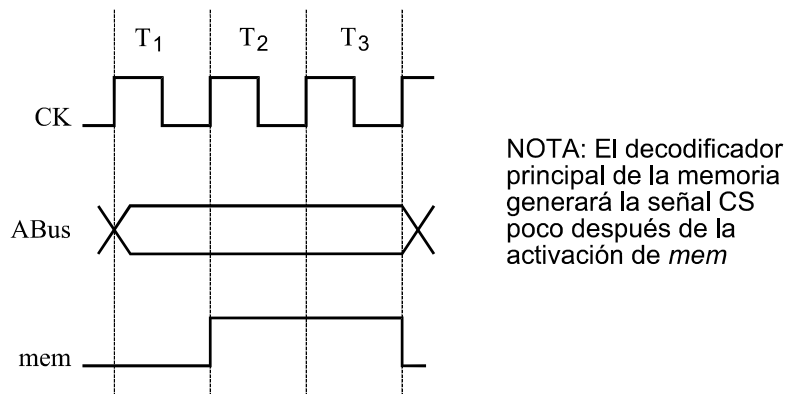


Fig.2.11. Señales generadas por la CPU para leer memoria.

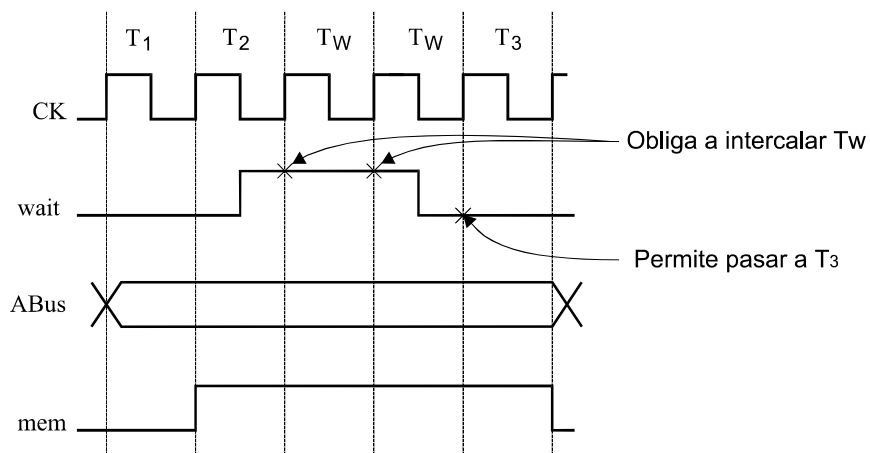
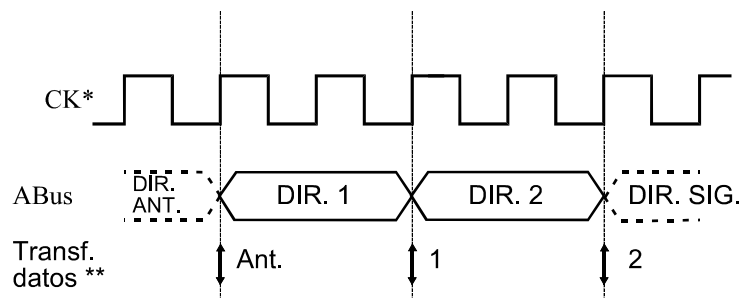
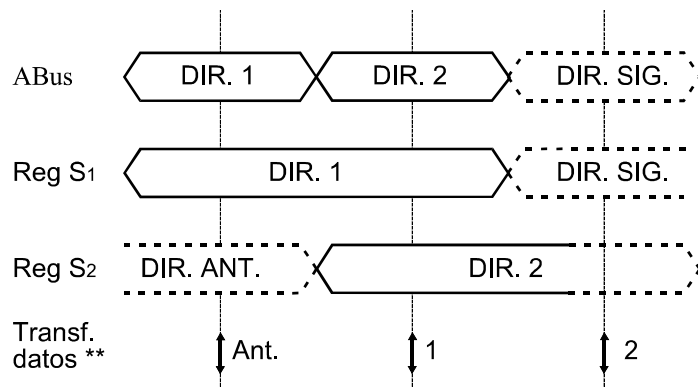


Fig.2.12. Acceso a la memoria con dos ciclos de espera.



(a)



(b)

* La señal CK es una referencia tanto para (a) como para (b)


** El símbolo  especifica el instante en el que se realiza la transferencia del dato

Fig.2.13. Anticipación de direcciones.

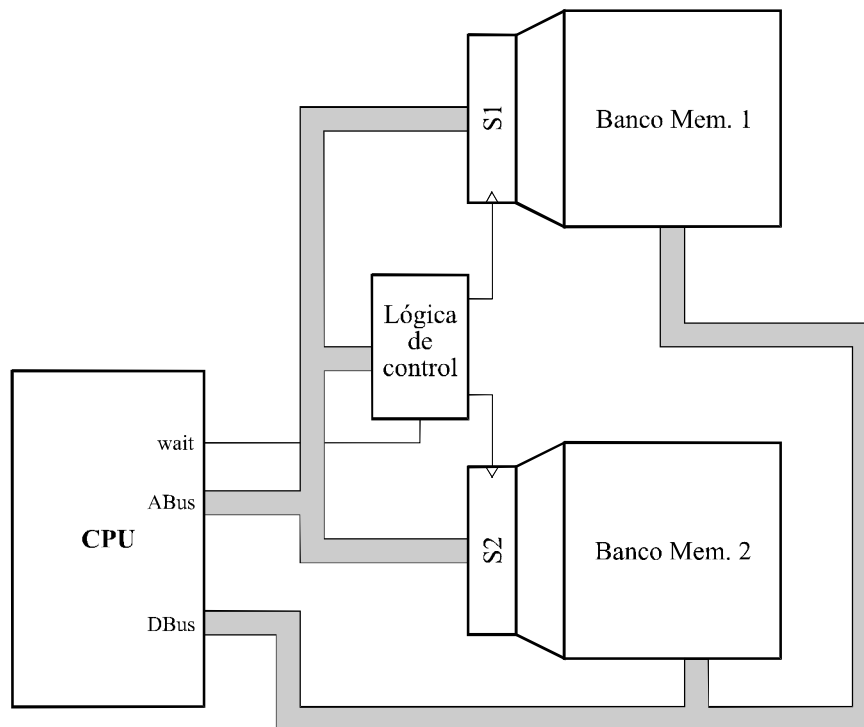


Fig.2.14. Configuración de memoria entrelazada.

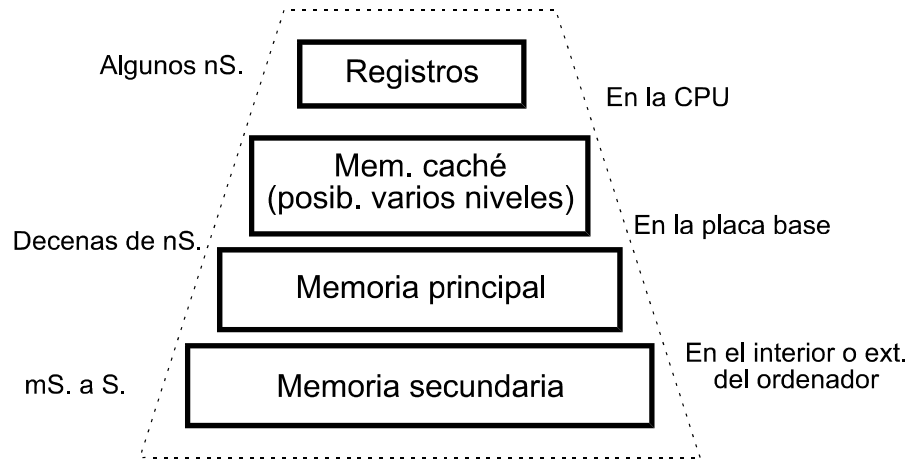


Fig.2.15. Jerarquía de memorias.



CAPÍTULO 3: UNIDAD DE PROCESO

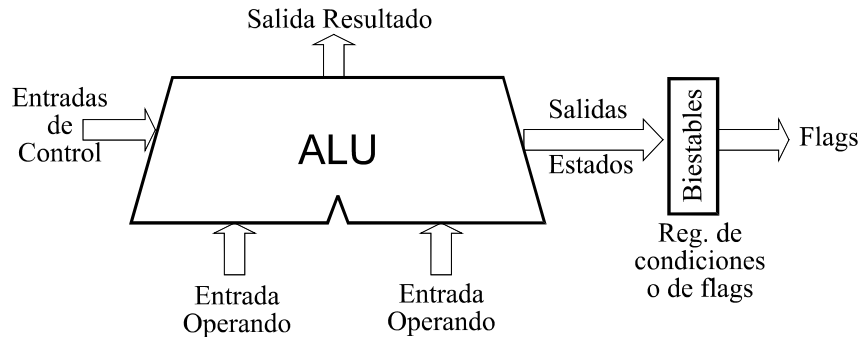
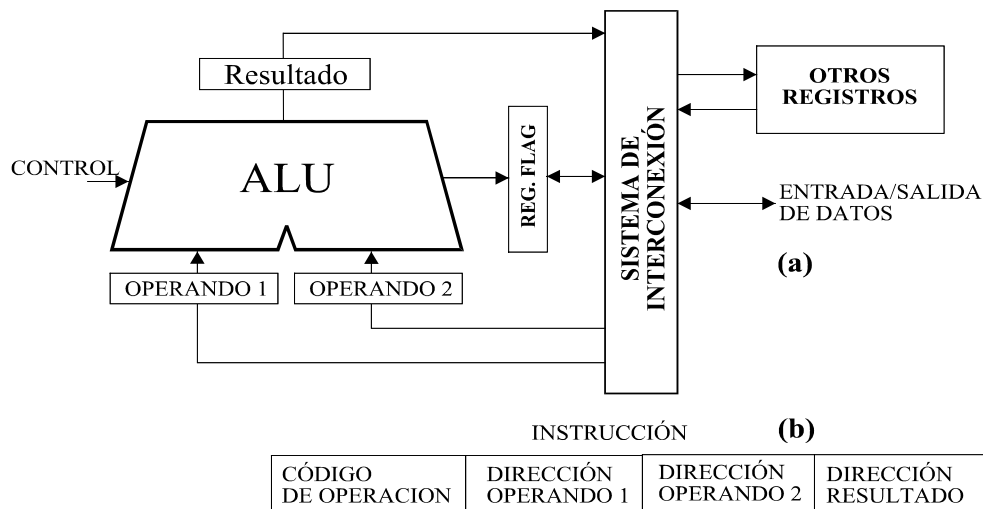
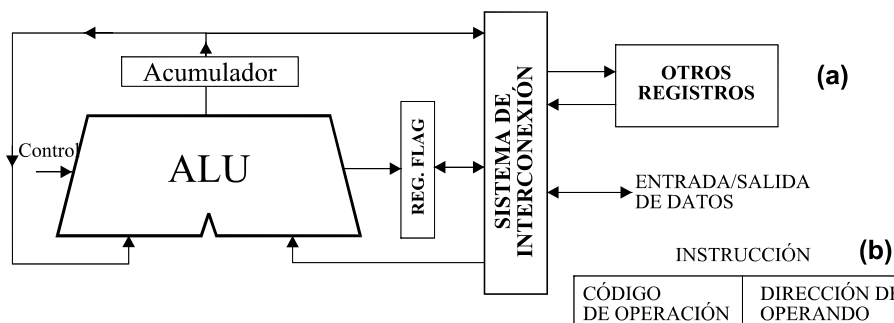


Fig.3.1. Esquema del bloque funcional de una ALU.



**Fig.3.2. (a) Unidad de Proceso de una máquina de tres direcciones.
(b) Formato de instrucción que utiliza.**



**Fig.3.3. (a) Unidad de Proceso de una máquina de una sola dirección.
(b) Formato de instrucción que utiliza.**

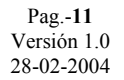


Fig.3.5.- ALU de 4 bits, con registro de flags.

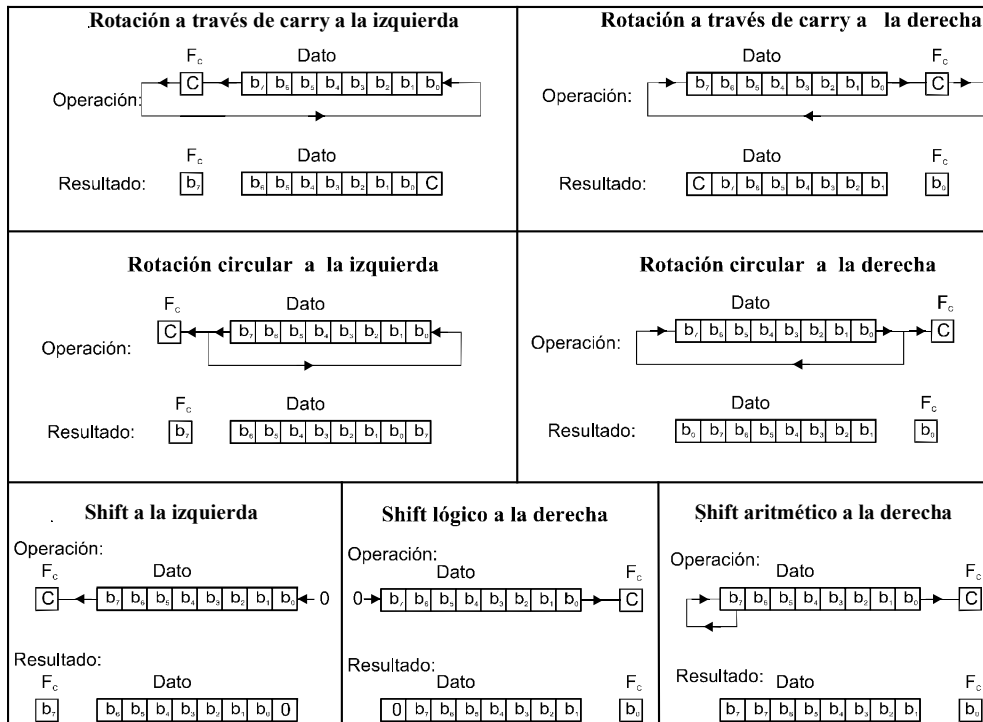


Fig.3.6. Shifts y rotaciones.

ESTADO DE LOS FLAGS		
F_Z	F_C	Significado
0	0	$A > B$
0	1	$A < B$
1	0	$A = B$

COMPROBACIÓN DE FLAGS	
Condición	Flags a comprobar
$A > B$	$F_C = 0$ y $F_Z = 0$
$A = B$	$F_Z = 1$
$A < B$	$F_C = 1$
$A \geq B$	$F_C = 0$
$A \geq B$	$F_Z = 0$
$A \leq B$	$F_C = 1$ ó $F_Z = 1$

Tabla 3.2.- Comparación o resta entre enteros sin signo (A-B).

ESTADO DE LOS FLAGS			
F_Z	F_S	F_O	Significado
0	0	0	$A > B$
0	0	1	$A < B$
0	1	0	$A < B$
0	1	1	$A > B$
1	0	0	$A = B$

COMPROBACIÓN DE FLAGS	
Condición	Flags a comprobar
$A > B$	$F_O = F_S$ y $F_Z = 0$
$A = B$	$F_Z = 1$
$A < B$	$F_O \neq F_S$
$A \geq B$	$F_O = F_S$
$A < B$	$F_Z = 0$
$A \leq B$	$F_Z = 1$ ó $F_O \neq F_S$

Tabla 3.3.- Comparación o resta entre enteros con signo (A-B).



CAPÍTULO 4: UNIDAD DE CONTROL

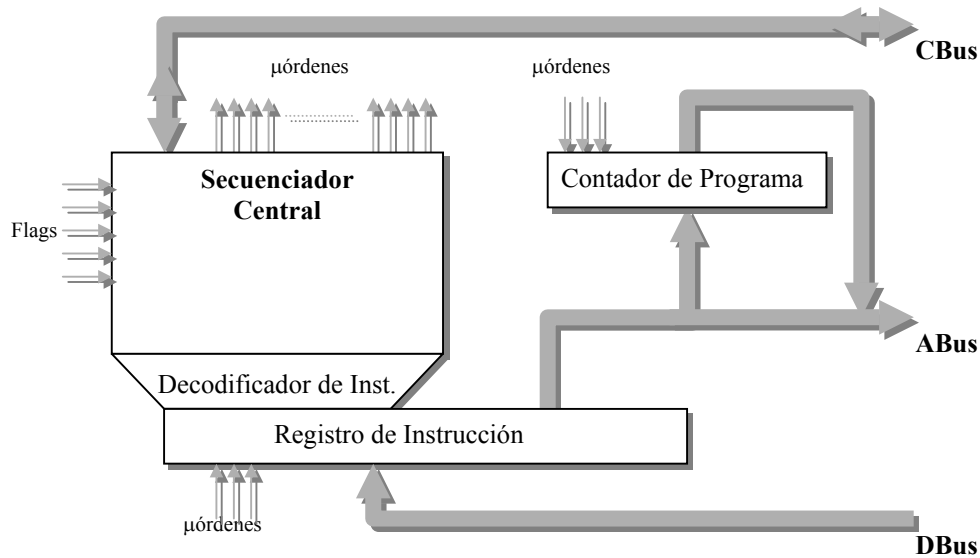


Fig.4.1. Unidad de Control.

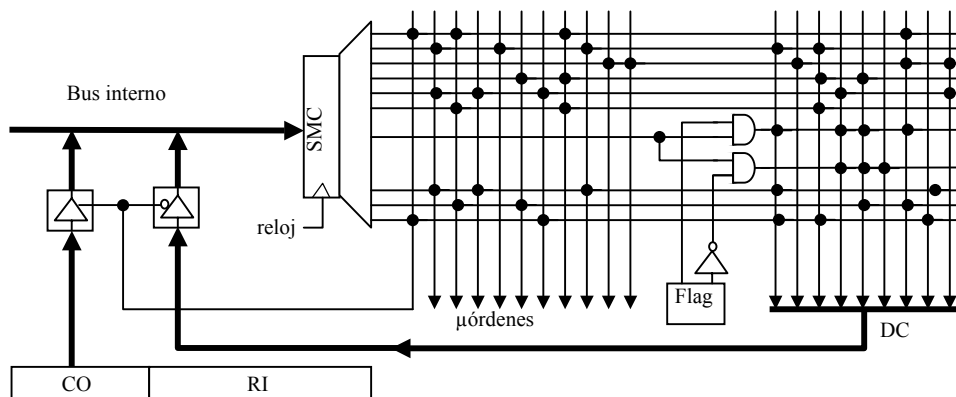
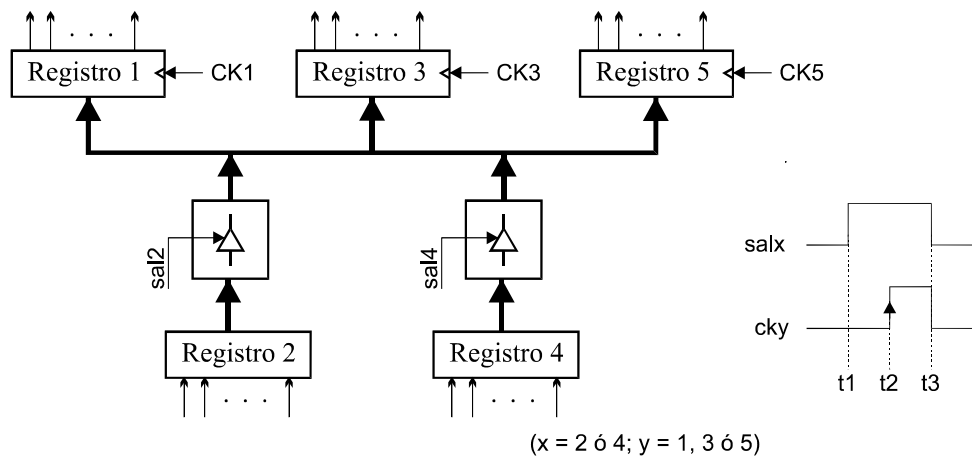


Fig.4.2. Unidad de control μprogramada según Wilkes 1.951



(x = 2 ó 4; y = 1, 3 ó 5)

Fig.4.3. Transferencia entre registros a través de un bus.

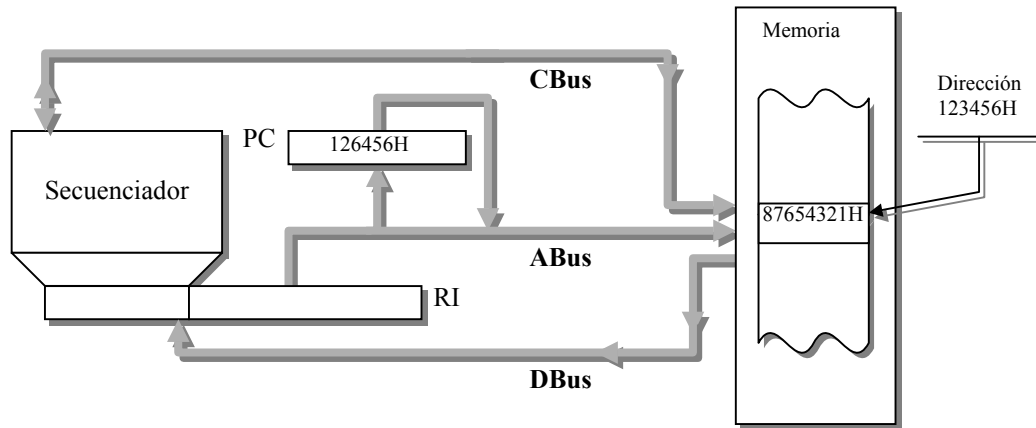


Fig.4.4. Unidad de control y memoria.

OPERACIÓN	μINSTRUCCIÓN	μÓRDENES
FETCH (PC)→RI; PC++	PC→ABus	Salida de PC
	(ABus)→DBus	Líneas de Cbus necesarias para leer la memoria
	DBus→RI	Carga paralelo de RI
	PC+1→PC	Incrementar PC
INTERNA RI [*] →PC	RI _l [*] →ABus	Salida de RI _l [*]
	ABus→PC	Carga paralelo de PC

* Denominaremos en esta tabla RI_l a los 24 bits menos significativos de RI, que son los que contienen la dirección a la que hay que saltar.

Tabla 4.1. Microinstrucciones necesarias para leer una instrucción.

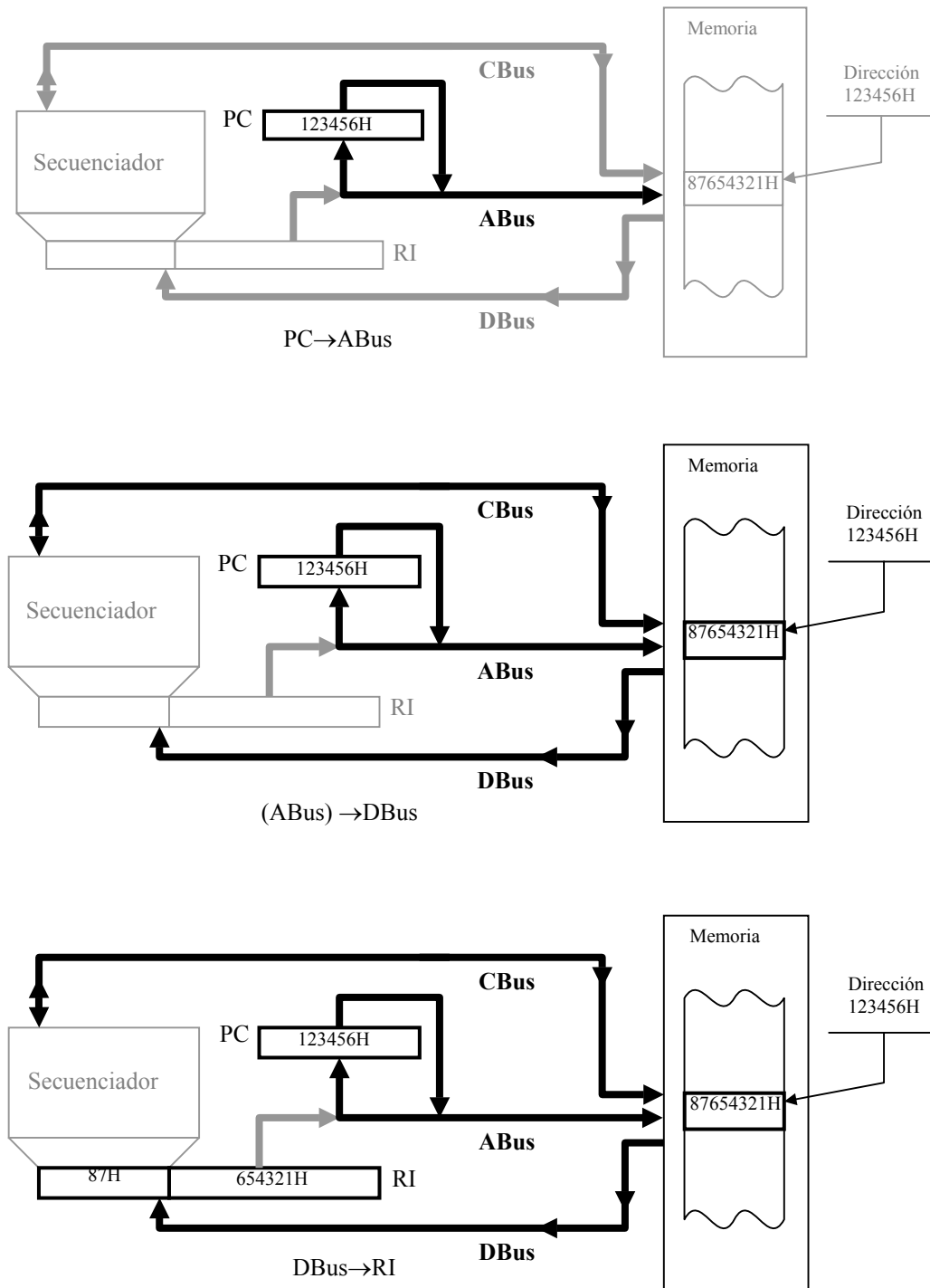


Fig.4.5. Las 3 primeras microórdenes.

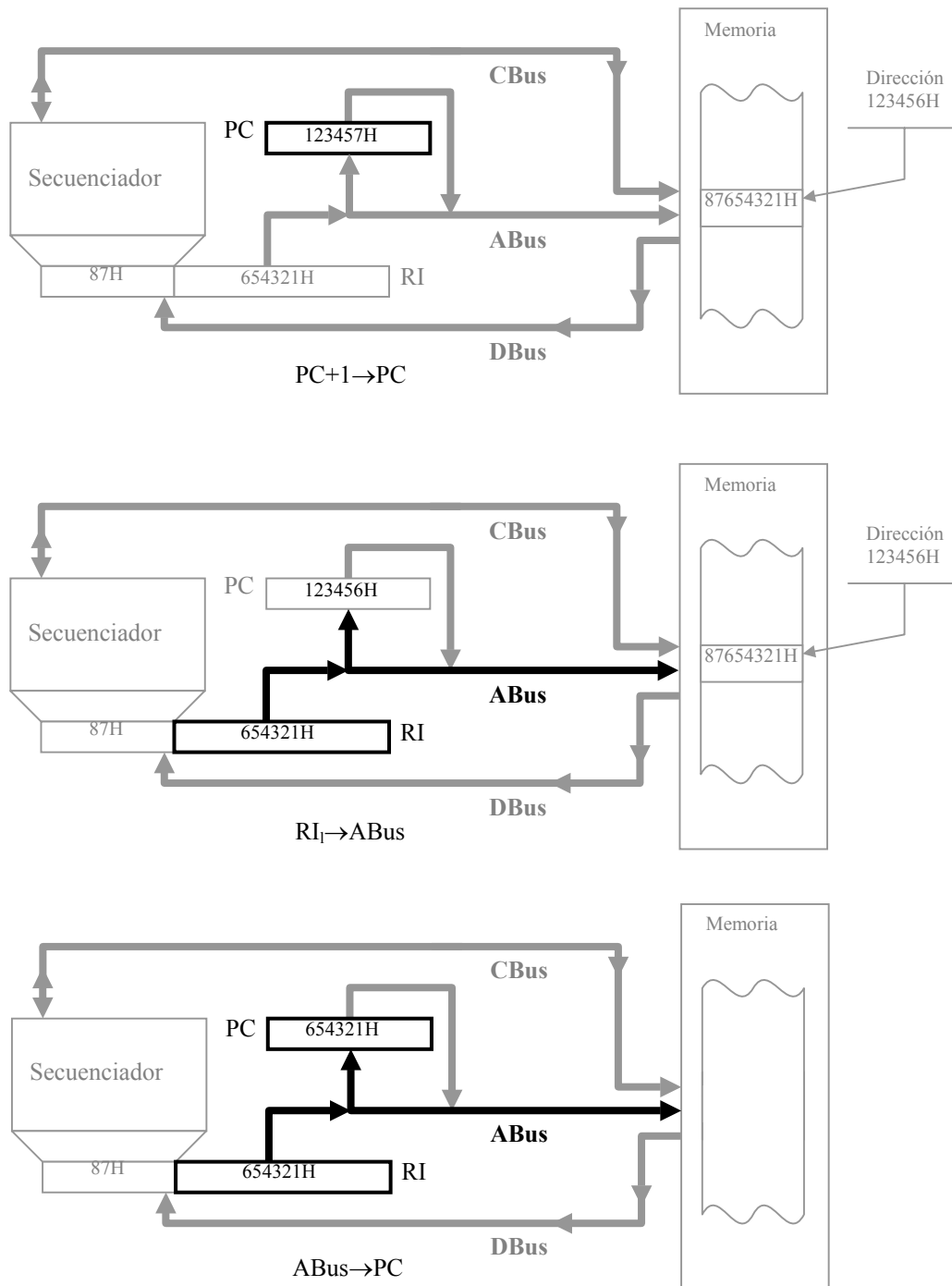


Fig.4.6. Las 3 últimas microórdenes.