U.N.J.U. – F.I.

INGENIERÍA INFORMÁTICA

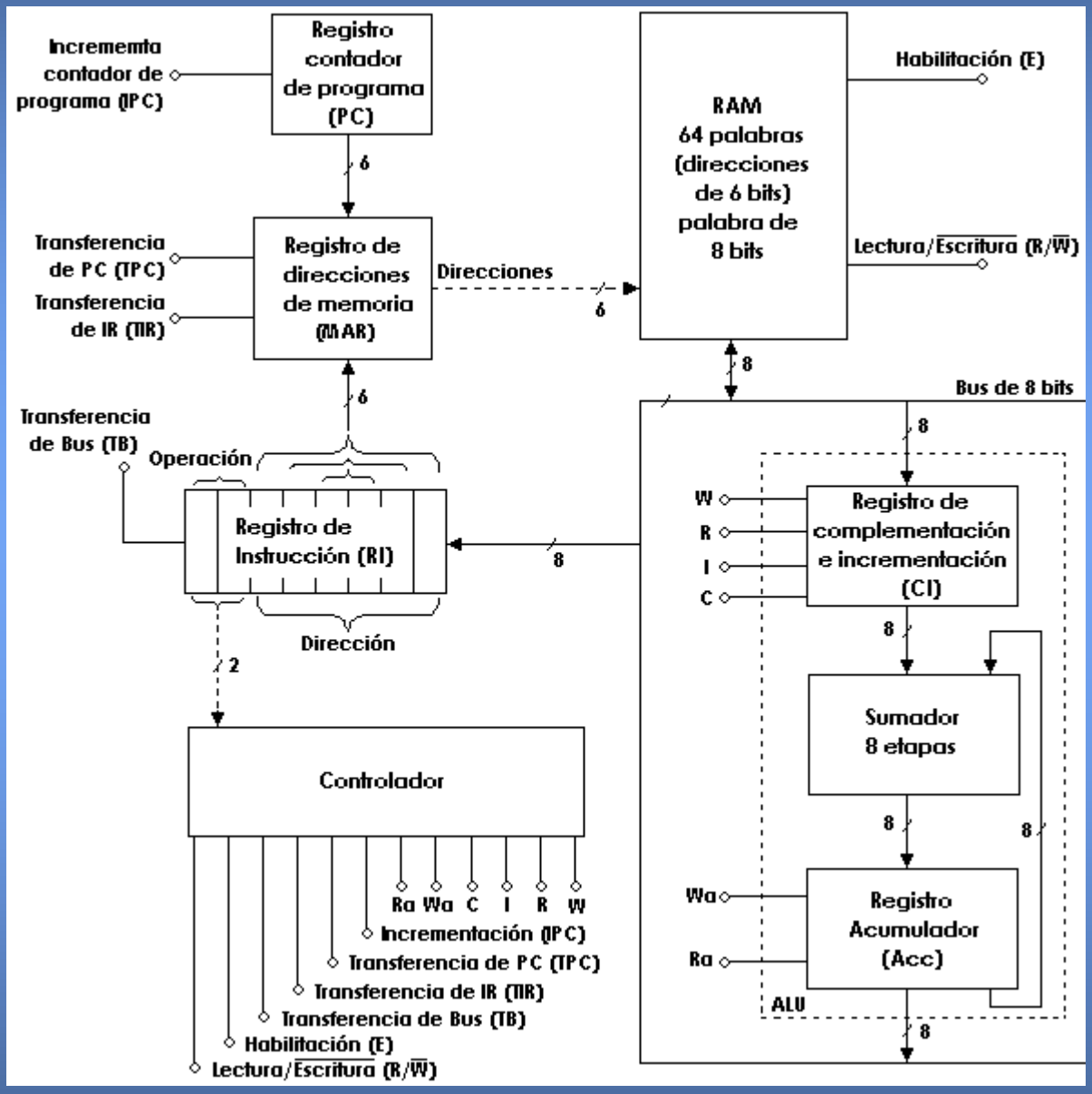
LABORATORIO DE COMPUTADORAS

ARQUITECTURA BÁSICA DE LAS COMPUTADORAS

TEMA: ARQUITECTURA Y MICROPROGRAMACIÓN

Ing. David Sánchez Rivero

Arquitectura Sencilla



Glosario

PC: registro contador de programa, contiene la dirección de la próxima instrucción a ejecutarse.

MAR: registro de direcciones de memoria, contiene una dirección de memoria que puede ser la de una instrucción o la de un dato.

RAM: memoria de lectura/escritura.

RI: registro de instrucción, contiene, básicamente el código de operación de la instrucción a ejecutarse y otro tipo de información.

ALU: unidad aritmética/lógica.

Acc: registro acumulador, es un registro de propósito general.

Controlador: circuito que controla la ejecución de las microoperaciones en el orden correcto, en el momento adecuado.

Ejemplo del contenido de la memoria

Dirección de Memoria	
000000	10111011
000001	01111100
000010	10111101
000011	01111110
000100	01111111
000101	11100111
000110	00xxxxxx
:	
:	
111011	00110001
111100	10110001
111101	11001100
111110	01111001
111111	01010010

Código	Instrucción
00	Alto
01	Suma al Acc
10	Resta del Acc
11	Transfiere el contenido del Acc a ...

Códigos de Instrucción

Dirección de Memoria	Explicación
0000000	Resta del Acc el contenido de la dirección de memoria 59_{10}
0000001	Suma al Acc el contenido de la dirección de memoria 60_{10}
0000010	Resta del Acc el contenido de la dirección de memoria 61_{10}
0000011	Suma al Acc el contenido de la dirección de memoria 62_{10}
0000100	Suma al Acc el contenido de la dirección de memoria 63_{10}
0000101	Transfiere el contenido del Acc a la dirección de memoria 39_{10}
0000110	Alto
. . .	
111011	49_{10}
111100	-79_{10}
111101	-52_{10}
111110	121_{10}
111111	82_{10}

Ciclo de Búsqueda

Ciclo de reloj	Descripción simbólica de la operación	Líneas de control a habilitar
1	PC → MAR	TPC
2	M → IR (M representa palabra de memoria direccionada) PC + 1→ PC	E, R/$\overline{W}$, TB IPC

Explicación:

1- Transfiere el contenido del contador de programa al registro de direcciones de memoria.

2- Transfiere la instrucción direccionada al registro de instrucción mediante: a) habilitación de la memoria al conectarla al bus; b) poniendo a R/W! a 1 al leer memoria, y c) transfiriendo la palabra del bus al registro de instrucción; y finalmente incrementando el contador de programa preparándolo para llamar a la siguiente instrucción cuando se haya completado la respuesta a la primera instrucción.

Ciclo de Ejecución

Operación: sustracción (por complementación)

Ciclo de reloj	Descripción simbólica de la operación	Líneas de control a habilitar
3	IR(AD) → MAR	TIR
4	M → BUS BUS → CI	E, R/$\overline{W}$, W
5	$\overline{CI}$ → CI	C
6	CI + 1→ CI	I
7	Sumador → Acc	R,Wa

Explicación:

3- Transfiere la parte dirección del registro RI al registro de direcciones de memoria.

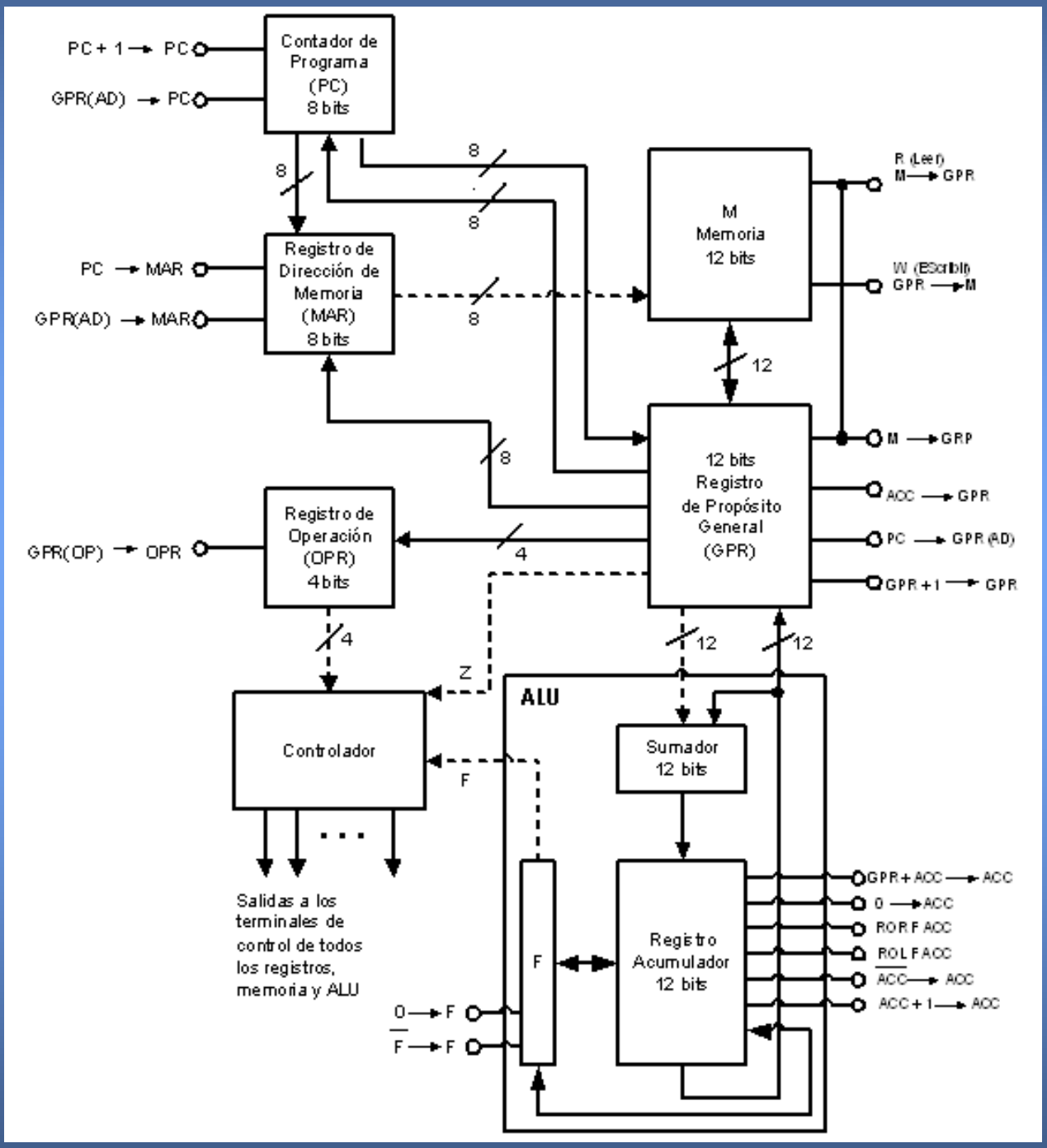
4- Transfiere la palabra direccionada de memoria al bus y de ahí al registro CI.

5- Complementa CI.

6- Incrementa CI.

7- Salida del registro sumador al registro acumulador.

Arquitectura Básica



Componentes y Operaciones del Sistema

Componente	Simbolismo de control	Explicación
Memoria	1. GPR → M	Escribe el contenido del GPR en el lugar de memoria direccionado
Contador de programa (PC)	2. PC + 1→ PC 3. GPR(AD) →PC	Incrementa el PC Transmite los bits de dirección del GPR al PC
Registro de direcciones de memoria (MAR)	4. PC → MAR 5. GPR(AD) →MAR	Transmite desde el PC al MAR Transmite los bits de dirección del GPR al MAR
Registro de Operación (OPR)	6. GPR(OP) →OPR	Transmite los bits de operación del GPR al OPR
Registro de Propósito General (GPR)	7. M→GPR 8. Acc→GPR 9. PC→GPR(AD) 10. GPR + 1 →GPR	Transmite palabra direccionada al GPR Transmite el contenido del Acc al GPR Transmite el contenido del PC a la parte dirección del GPR Incrementa el GPR

Componentes y Operaciones del Sistema

Componente	Simbolismo de control	Explicación
Unidad Aritmética Lógica (ALU)	11. GPR + Acc→Acc	Suma el número del GPR al número del Acc y deja el resultado en Acc
	12. 0 → Acc	Limpia Acc
	13. ROR F, Acc	Desplaza cíclicamente a la derecha el Acc junto con F
	14. ROL F, Acc	Desplaza cíclicamente a la izquierda el Acc junto con F
	15. 0 → F	Pone a "0" el flip-flop F
	16. $\bar{F}$ → F	Complementa el flip–flop F
	17. $\overline{Acc}$ → Acc	Complementa el Acc
	18. Acc + 1 →Acc	Incrementa el Acc

Operaciones ROR y ROL

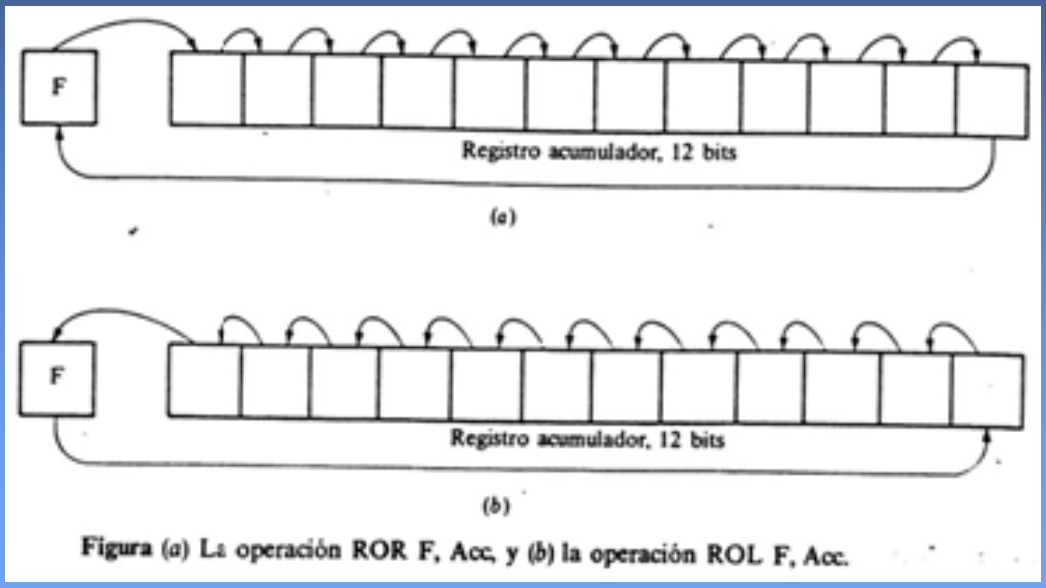


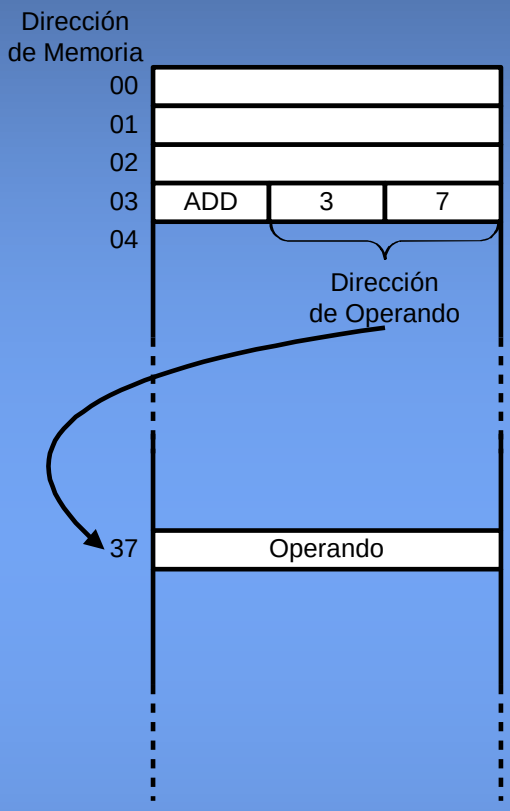
Figura (*a*) La operación ROR F, Acc, y (*b*) la operación ROL F, Acc.

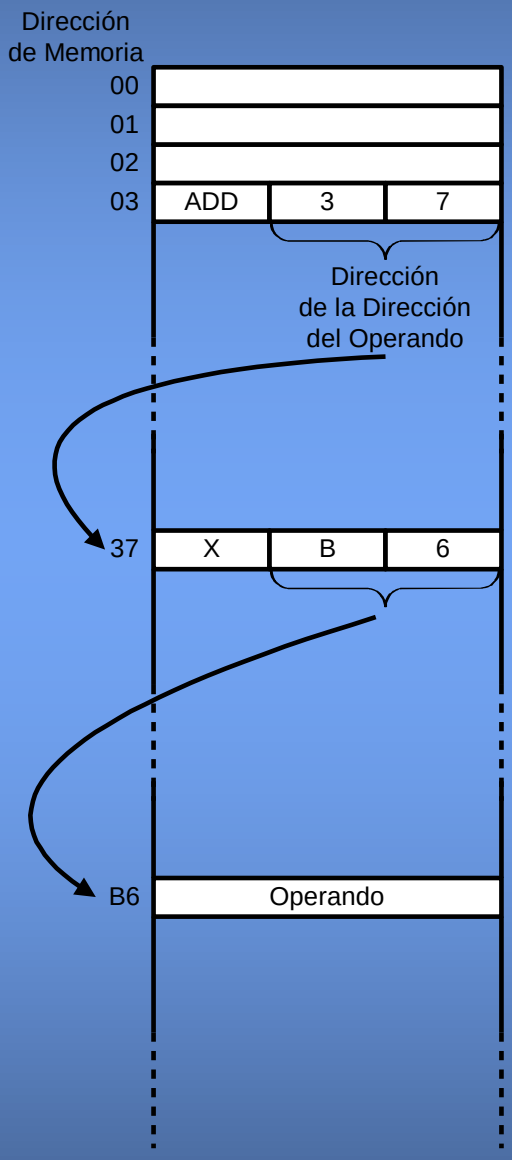
Como se puede apreciar, en la figura, la operación ROR desplaza a todos los bits del acumulador y el F.F. una posición a la derecha, sin perder ninguno. Por el contrario, la operación ROL realiza el desplazamiento, de cada bit, una posición hacia la izquierda. Como aplicación práctica un ROR divide por 2 y un ROL multiplica por 2, en ambos casos el F.F. debe estar en 0 previamente.

Ciclo de Búsqueda

Ciclo de reloj	Microoperación	Explicación
1	PC → MAR	Transmitir la posición de la instrucción desde el PC al MAR.
2	M → GPR PC + 1→ PC	Transmitir la palabra direccionada al GPR. Incrementar el PC.
3	GPR(OP) → OPR	Transmitir la parte operación de la instrucción al OPR.

Direccionamiento Directo e Indirecto





Ciclo de ejecución

Operación: ADD, dirección (direccionamiento directo)

Ciclo de reloj	Microoperación	Explicación
4	GPR(AD) → MAR	Transmitir la dirección del operando desde GPR(AD) a MAR.
5	M → GPR	Lee desde la memoria la palabra de la celda cuya dirección está en el MAR.
6	GPR + Acc → Acc	Suma el contenido del GPR al contenido del Acc, dejando la suma en el Acc.

Ciclo de ejecución

Operación: ADD, dirección (direccionamiento indirecto)

Ciclo de reloj	Microoperación	Explicación
4	GPR(AD) → MAR	Transmite la dirección desde GPR al MAR.
5	M → GPR	Transmite el contenido de la celda de memoria direccionada al GPR (el GPR tendrá entonces la dirección del operando).
6	GPR(AD) → MAR	Transmite la dirección del operando desde GPR(AD) al MAR.
7	M → GPR	Transmite el operando direccionado al GPR.
8	GPR + Acc → Acc	Suma el contenido del GPR al Acc.

Ciclo Fetch-Execute

EJEMPLO: Escribir el microprograma del ciclo de ejecución de la siguiente instrucción:

Mnemotécnico	Direccionamiento	Modo de Operación
DEC Q	Directo	M’<--M – 2*ACC

Analizando a partir del instante en que termina la instrucción anterior, especificar en notación hexadecimal el contenido de cada registro usado en la arquitectura básica de una computadora después de cada ciclo de reloj hasta que se completa la ejecución de la instrucción.

Suponer que:

PC = 40h, ACC = 02Ah, Q = 10h,

OP = 2h, (Q) = 060h

Resolución

CICLO	CÓDIGO	PC (40h)	MAR (?)	GPR (?)	OPR (?)	M (?)	ACC (02Ah)	F (?)	(40h)	Q	(Q)
BÚSQUEDA	PC → MAR	40h	40h	?	?	210h	02Ah	?	210h	10h	060h
	M → GPR	40h	40h	210h	?	210h	02Ah	?	210h	10h	060h
	PC+1 → PC	41h	40h	210h	?	210h	02Ah	?	210h	10h	060h
	GPR(OP) → OPR	41h	40h	210h	2h	210h	02Ah	?	210h	10h	060h
EJECUCIÓN	GPR(AD) → MAR	41h	10h	210h	2h	060h	02Ah	?	210h	10h	060h
	M → GPR	41h	10h	060h	2h	060h	02Ah	?	210h	10h	060h
	0 → F	41h	10h	060h	2h	060h	02Ah	0	210h	10h	060h
	ROL F, ACC	41h	10h	060h	2h	060h	054h	0	210h	10h	060h
	ACC! → ACC	41h	10h	060h	2h	060h	FABh	0	210h	10h	060h
	ACC+1 → ACC	41h	10h	060h	2h	060h	FACh	0	210h	10h	060h
	GPR+ACC → ACC	41h	10h	060h	2h	060h	00Ch	0	210h	10h	060h
	ACC → GPR	41h	10h	00Ch	2h	060h	00Ch	0	210h	10h	060h
	GPR → M	41h	10h	00Ch	2h	00Ch	00Ch	0	210h	10h	00Ch

Como puede apreciarse, el contenido de la posición de memoria Q es 060h y el contenido inicial del ACC es 02Ah. Y el resultado de la instrucción será 00Ch y queda almacenado en la posición Q de memoria.

Sumario de Instrucciones

N°	Mnemotécnico	Descripción
1	CRA	Borrar el acumulador
2	CTA	Complementar el acumulador
3	ITA	Incrementar el acumulador
4	CRF	Borrar el flip-flop F
5	CTF	Complementar el flip-flop F
6	SFZ	Saltar a la siguiente instrucción si F = 0
7	ROR	Desplazar cíclicamente a la derecha
8	ROL	Desplazar cíclicamente a la izquierda
9	ADD	Sumar al acumulador
10	ADDI	Sumar indirecto al acumulador
11	STA	Almacenar en memoria el contenido del acumulador
12	JMP	Bifurcar
13	JMPI	Bifurcar indirecto
14	CSR	Llamada a subrutina
15	ISZ	Incrementar y saltar si Z = 0
16	HLT	Alto

Con este sumario de instrucciones se pueden desarrollar pequeños programas para resolver problemas de aplicación.

Instrucciones: STA Q , ISZ Q y JMPI Q

STA, dirección

Almacena el contenido del acumulador en la dirección de memoria especificada.

La secuencia de microoperaciones es:

Ciclo de reloj	Microoperación	Explicación
1	GPR(AD)→MAR	Transmite la dirección desde el GPR al MAR
2	Acc→GPR	Transmite el contenido del Acc al GPR
3	GPR→M	Escribe el contenido del GPR en la dirección de memoria retenida en el MAR

ISZ, direccion

Incrementar y saltar si cero (Increment and skip if zero). Leer el número de la celda especificada de memoria, incrementarlo y devolverlo a su celda original. Si después de incrementarlo, el número es cero, saltar a la siguiente instrucción.

Las microoperaciones que requiere esta instrucción ISZ son:

Ciclo de reloj	Microoperación	Explicación
1	GPR(AD)→MAR	Transmitir al MAR la dirección donde está alumbrado el número que ha de ser incrementado
2	M→GPR	Leer de la memoria el número
3	GPR+1→GPR	Incrementar el número
4	GPR→M	Devolver el número a la memoria
5	PC+1→PC (si GPR=0)	Saltar a la siguiente instrucción si GPR=0

JMPI, direccion:

Bifurca a la instrucción cuya dirección está almaecenada en la dirección de memoria especificada en la instrucción.

Esta instrucción es la forma *indirecta* de la instrucción JMP dada antes. La parte dirección de la instrucción JMPI es la dirección de la celda de memoria que retiene la *dirección de la siguiente* instrucción. La secuencia de microoperaciones es:

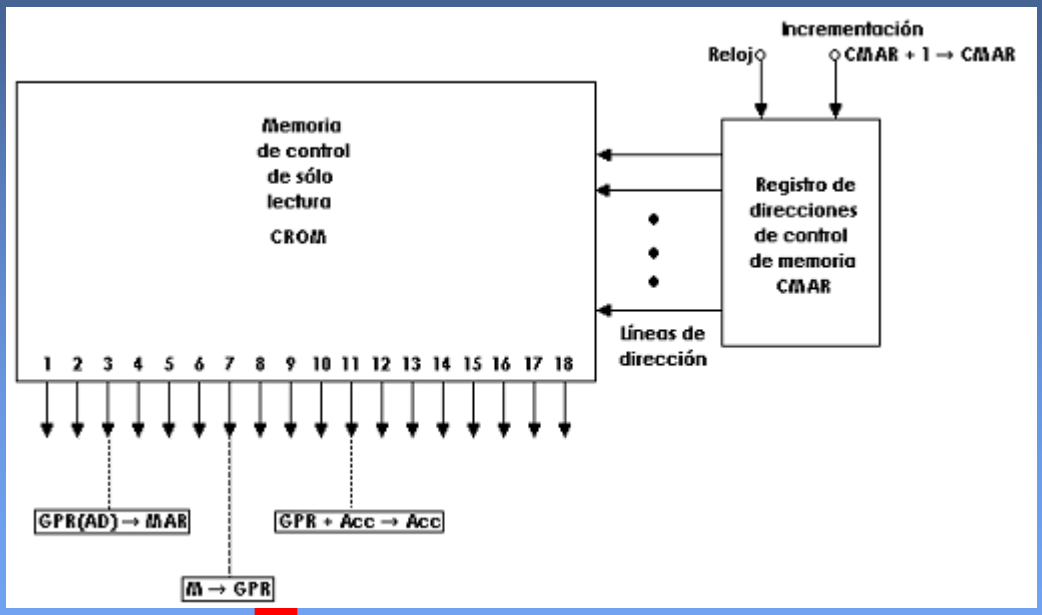
Ciclo de reloj	Microoperación	Explicación
1	GPR(AD)→MAR	Transmitir la dirección al MAR
2	M→GPR(AD)	Lee de la memoria la dirección de la instrucción siguiente
3	GPR(AD)→PC	Transmite la dirección de la instrucción siguiente desde el GPR al PC

Ejemplo de Programa

Aplicando el sumario anterior, desarrollar un programa que sume una serie de 100 sumandos que se encuentran almacenados a partir de la dirección FADh.

Rótulo	Contenido	Comentario
	CRA	Limpia el acumulador
LOOP	ADDI ANA	Suma al acumulador el número cuya dirección está en la celda ANA (dirección del siguiente sumando)
	ISZ ANA	Incrementa la dirección del sumando
	ISZ CTR	Incrementa el número en la celda CTR y se salta la siguiente instrucción si el incremento hace cero el contenido de CTR
	JMP LOOP	Bifurca a la instrucción rotulada LOOP
	STA RES	Almacena el resultado en la celda RES
	HLT	Para
RES	XXX	Aquí va a ser almacenado el resultado
ANA	FAD	Esta celda contiene la dirección del siguiente sumando
CTR	F9C	Esta celda contiene el número de sumas que han de hacerse (representación por complemento a dos de −100)
FAD	ADD(1)	100 sumandos que han de sumarse
	ADD(2)	
	.	
	.	
	.	
	ADD(100)	

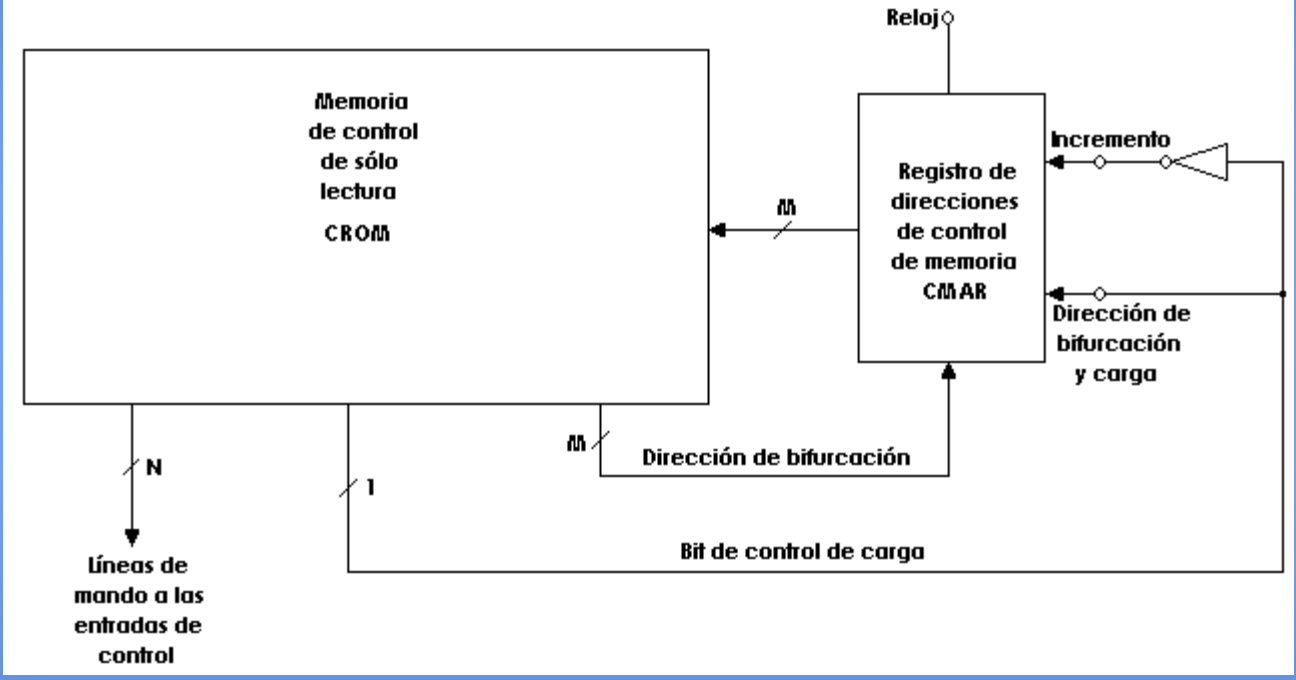
Microprogramación



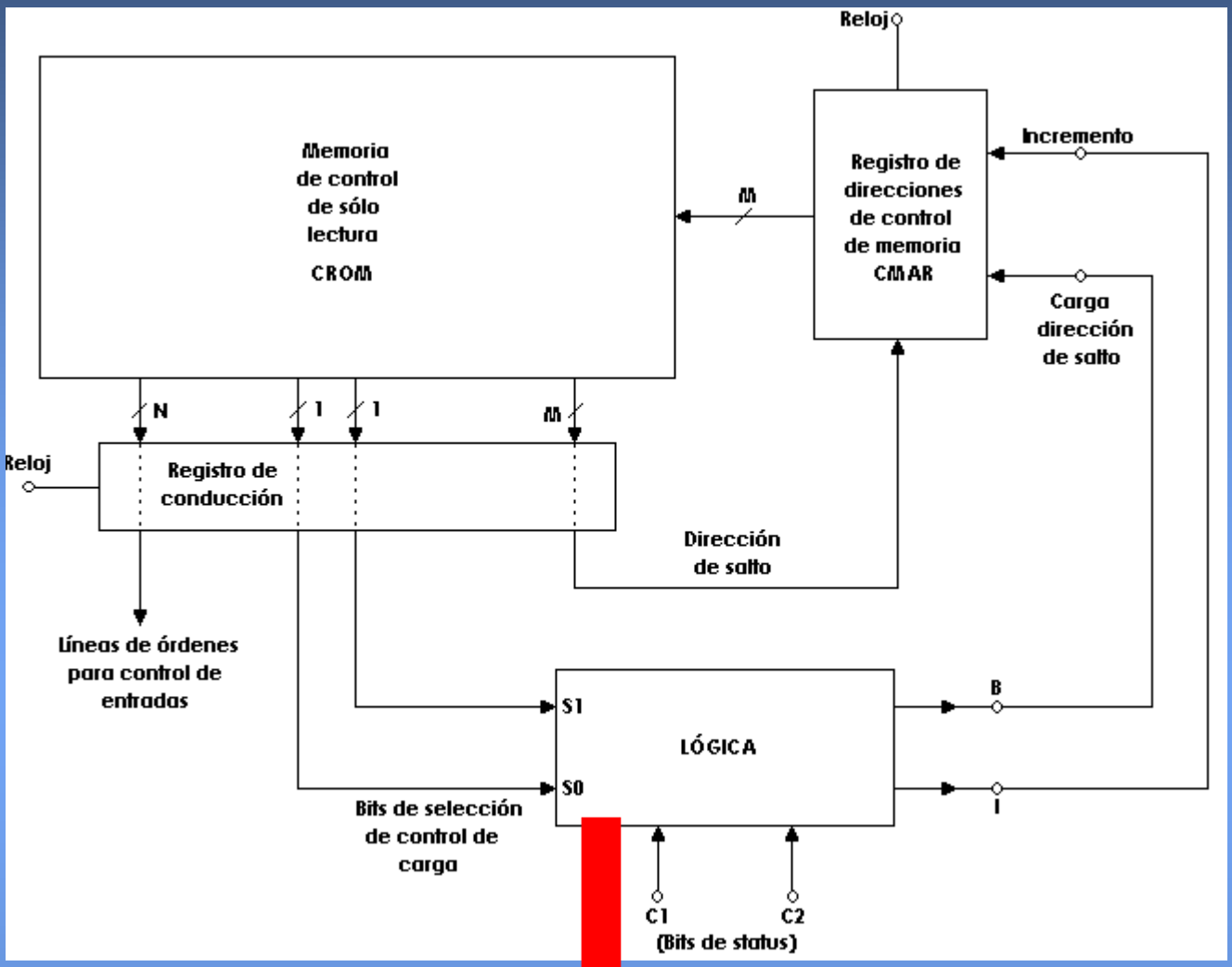
Ejemplo: ADD, dirección (Dir. Indirecto)

	Número de bit																	
Palabra	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bifurcación en Microprograma

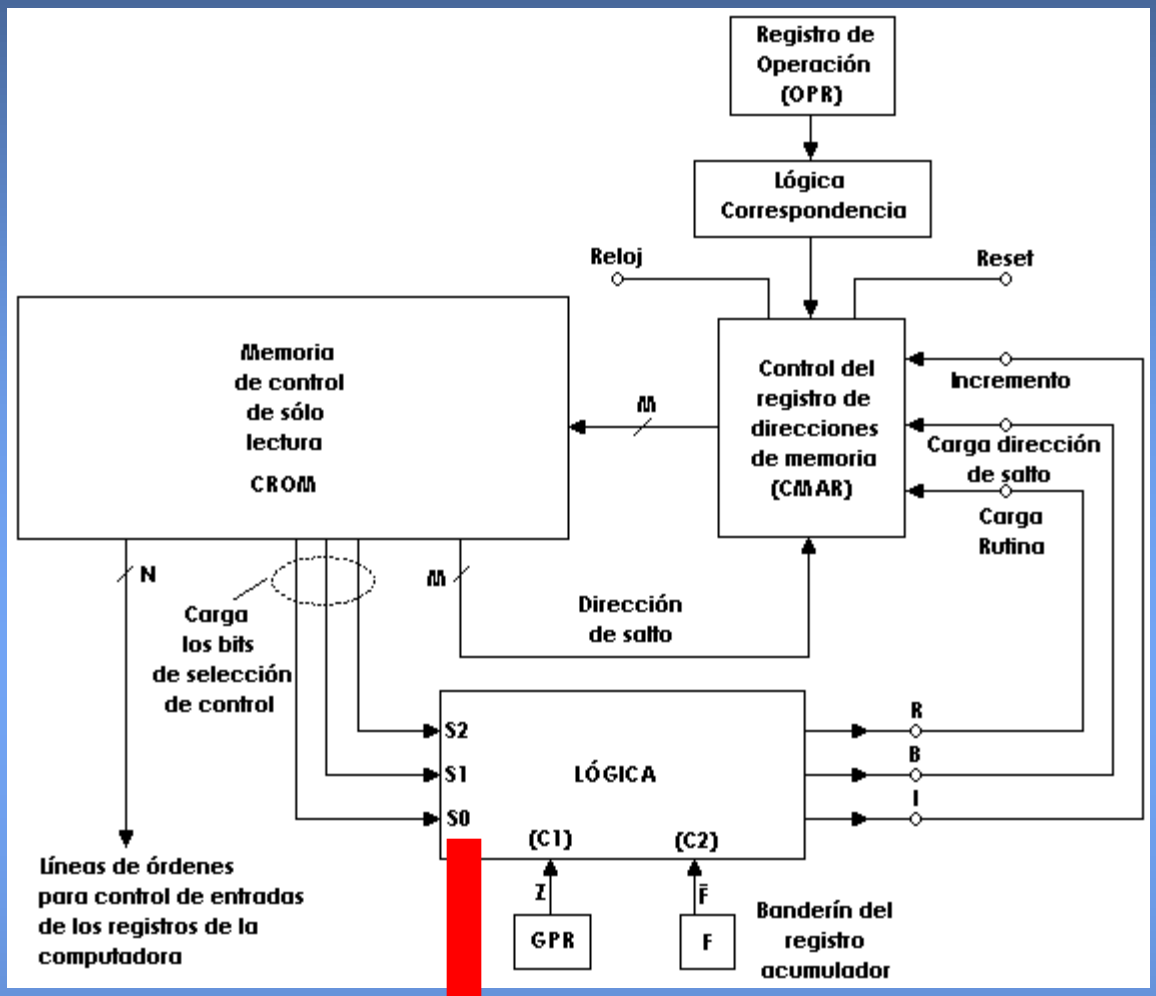


Bifurcación Condicional



S_1	S_0	C_1	C_2	B	I	
0	0	X	X	0	1	Incrementa. Ignora los bits de status.
0	1	X	X	1	0	Bifurca. Ignora los bits de status.
1	0	0	X	0	1	Responde sólo a C_1: Salta si $C_1 = 1$,
1	0	1	X	1	0	Incrementa si $C_1 = 0$
1	1	X	0	0	1	Responde sólo a C_2: Salta si $C_2 = 1$,
1	1	X	1	1	0	Incrementa si $C_2 = 0$

Controlador Microprogramado



S_2	S_1	S_0	C_1	C_2	B	I	R	
0	0	0	X	X	0	1	0	Incrementa. Ignora los bits de status.
0	0	1	X	X	1	0	0	Salta. Ignora los bits de status.
0	1	0	0	X	1	0	0	Responde sólo a C_1: Salta si $C_1 = 0$,
0	1	0	1	X	0	1	0	Incrementa si $C_1 = 1$
0	1	1	X	0	1	0	0	Responde sólo a C_2: Salta si $C_2 = 0$,
0	1	1	X	1	0	1	0	Incrementa si $C_2 = 1$
1	X	X	X	X	0	0	1	Carga Rutina