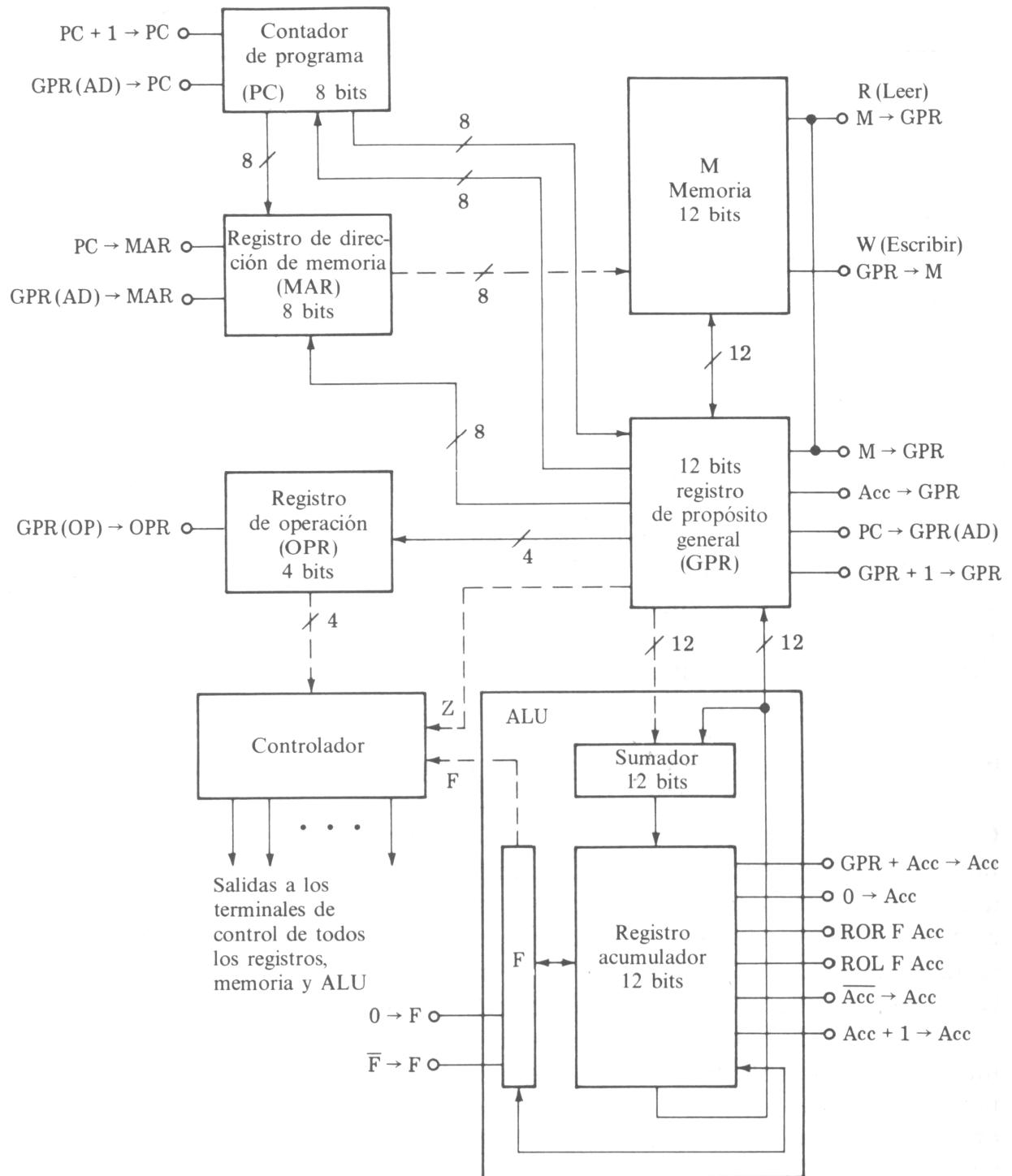
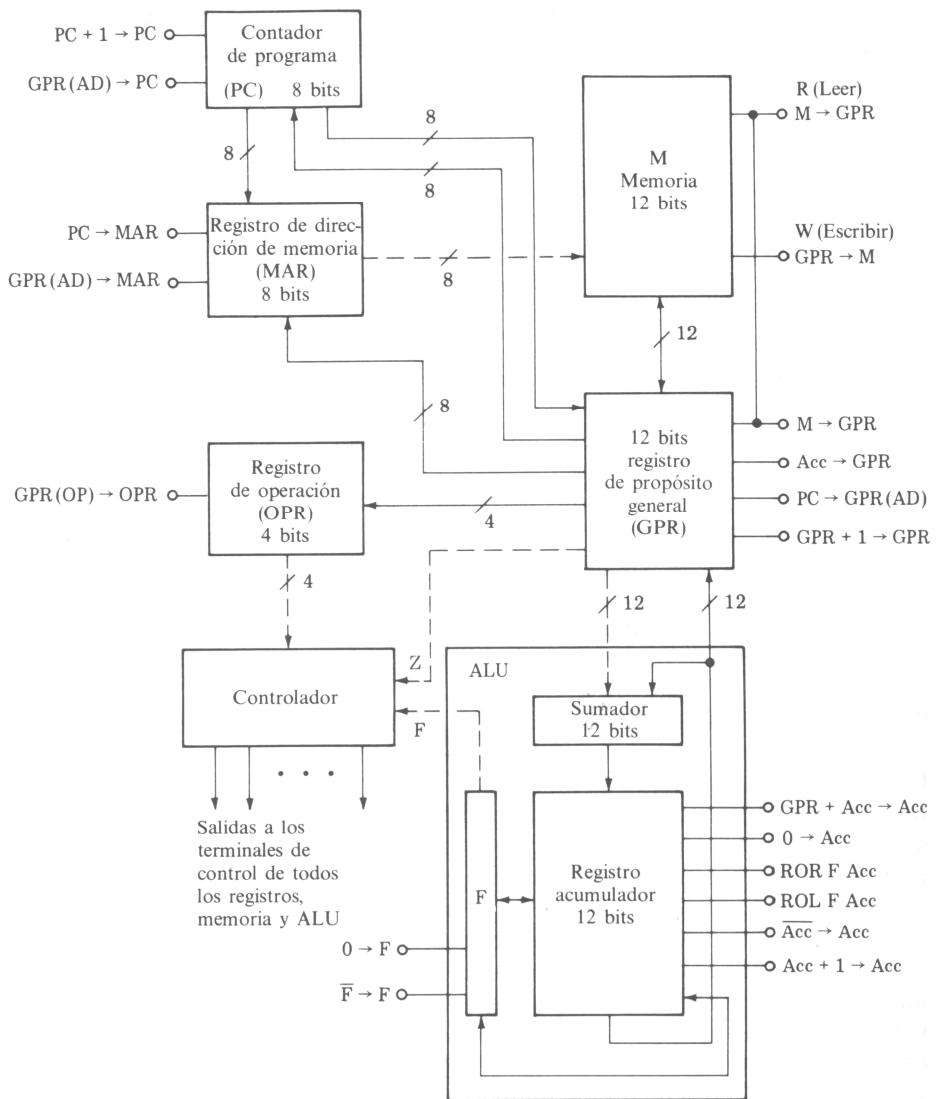


COMPUTADORA MEJORADA

Computadora mejorada

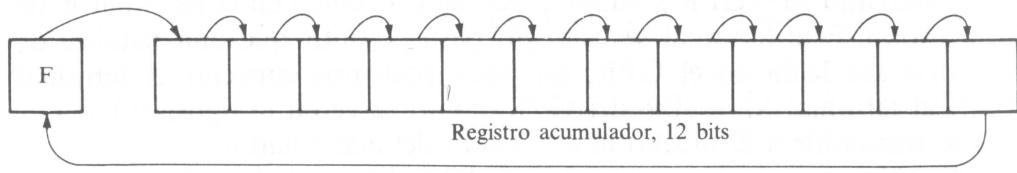


Computadora mejorada

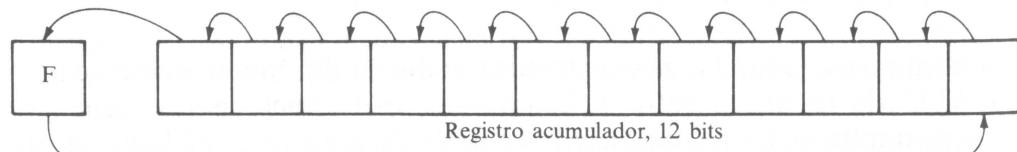


Conjunto de instrucciones

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 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



(a)



(b)

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

Componentes y operaciones de control 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 (PR)	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 MAR Incrementa el GPR
Unidad aritmética lógica (ALU)	11. GPR + Acc → Acc 12. 0 → Acc 13. ROR F, Acc 14. ROL F, Acc 15. 0 → F 16. F → F 17. Acc → Acc 18. Acc + 1 → Acc	Suma el número del GPR al número del Acc y deja el resultado en Acc Limpia Acc Desplaza cíclicamente a la derecha el Acc junto con F Desplaza cíclicamente a la izquierda el Acc junto con F Pone a «0» el flip-flop F Complementa el flip-flop F Complementa el Acc Incrementa el Acc

Ciclo de búsqueda

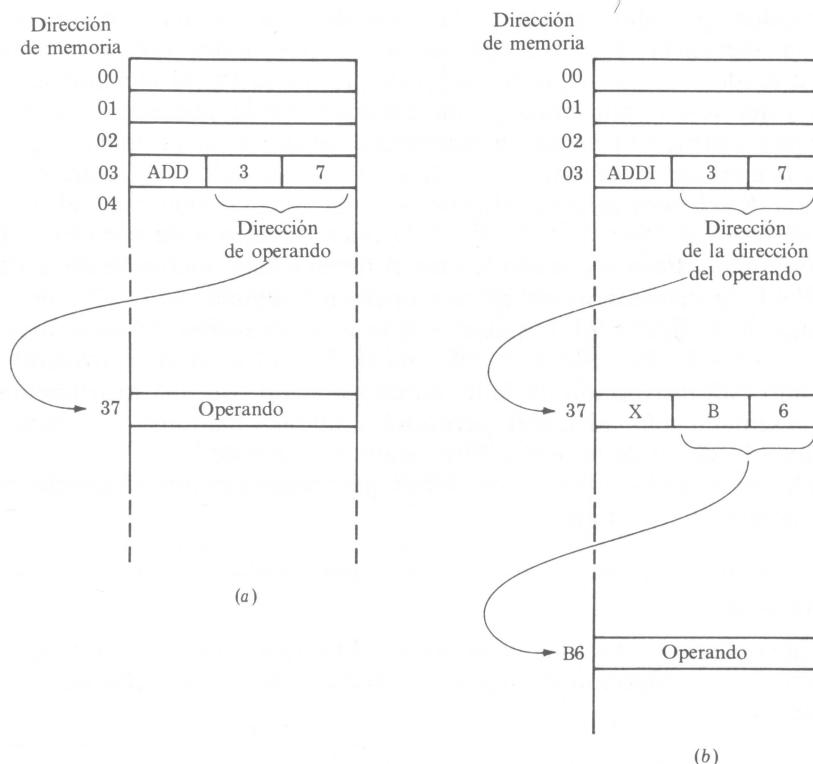
Ciclo de reloj	Microoperación	Explicación
1	$PC \rightarrow MAR$	Transmitir la posición de la instrucción desde el PC al MAR
2	$M \rightarrow GPR$ $PC + 1 \rightarrow PC$	Transmitir la palabra direccionada al GPR; incrementar el PC
3	$GPR(OP) \rightarrow OPR$	Transmitir la parte operación de la instrucción al OPR

Instrucciones que se componen de una sola micro-operación

Instrucción (microoperación)	Explicación	Nemotécnico
$0 \rightarrow Acc$	Limpiar acumulador	CRA
$Acc \rightarrow Acc$	Complementar acumulador	CTA
$Acc + 1 \rightarrow Acc$	Incrementa acumulador	ITA
$0 \rightarrow F$	Limpia flip-flop F	CRF
$\bar{F} \rightarrow F$	Complementa flip-flop F	CTF
$PC + 1 \rightarrow PC$	Salta a la siguiente instrucción si F es cero	SFZ
Desplazamiento cíclico a la derecha	Desplaza cíclicamente a la derecha el acumulador junto con F	ROR
Desplazamiento cíclico a la izquierda	Desplaza cíclicamente a la izquierda el acumulador junto con F y Acc	ROL

ADD dirección

Ciclo de reloj	Microoperación	Explicación
1	GPR(AD)→MAR	Transmite la dirección del operando desde GPR(AD) a MAR
2	M→GPR	Lee desde la memoria la palabra de la celda cuya dirección está en el MAR
3	GPR + Acc→Acc	Suma el contenido del GPR al contenido del Acc, dejando la suma en Acc



Diferencia entre (a) instrucción de direccionamiento directo y (b) instrucción indirecta.

ADDI dirección

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

STA dirección

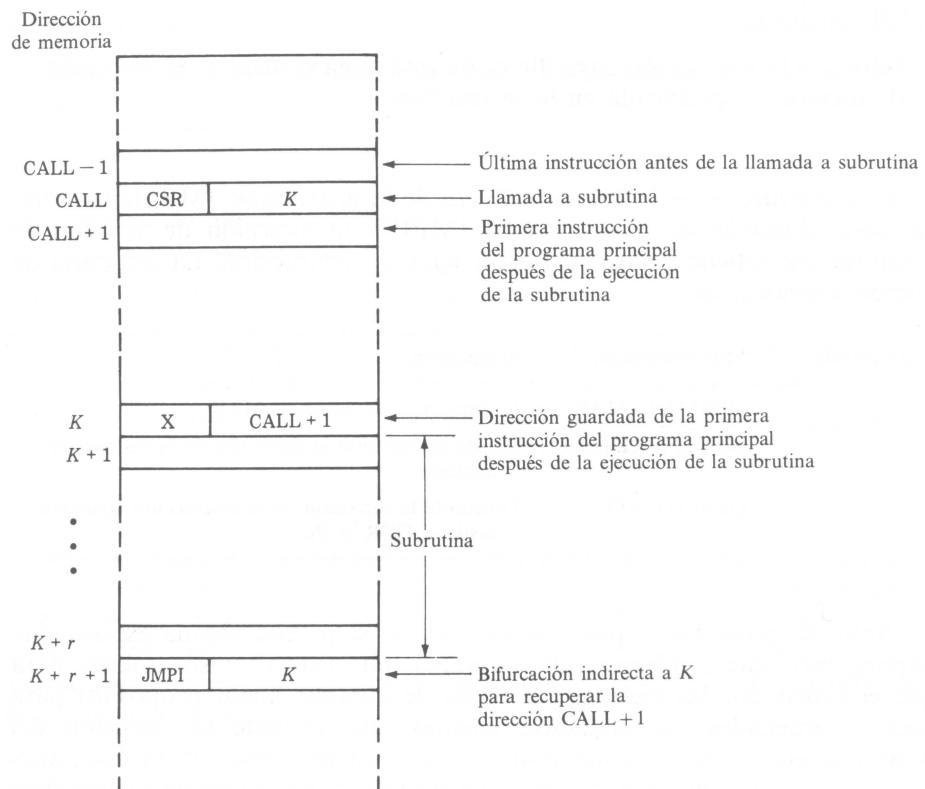
Ciclo de reloj	Microoperación	Explicación
1	GPR(AD)→MAR	Transmite la dirección desde el GPR al MAR
2	Acc→GPR	Transmite el contenido del Acc al GPR
3	GPR→M	Escribe el contenido del GPR en la dirección de memoria retenida en el MAR

JMP dirección

Ciclo de reloj	Microoperación	Explicación
1	GPR(AD)→PC	Transmite la dirección de la siguiente instrucción desde el GPR al PC

JMPI Dirección

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



Manipulaciones involucradas en una llamada a subrutina.

CSR dirección

Ciclo de reloj	Microoperación	Explicación
1	GPR(AD)→MAR	Transmite al MAR la dirección donde ha de almacenarse la dirección de vuelta; la dirección de almacenamiento es K
2	GPR(AD)→PC PC→GPR(AD)	Intercambia los contenidos del PC y del GPR [después del intercambio el PC contiene la dirección K y GPR(AD) contiene CALL +1 puesto que el PC fue incrementado desde el valor CALL A CALL +1 durante el ciclo de búsqueda]
3	GPR(AD)→M	Transmite el contenido de GPR(AD) a la memoria [el resultado es que la dirección CALL +1 será escrita en la celda de memoria K]
4	PC + 1 → PC	Incrementa el PC [el PC contendrá entonces la dirección $K + 1$; la siguiente instrucción será tomada entonces de la celda $K + 1$, donde está la primera instrucción de la subrutina]

ISZ Dirección

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

Ejemplos de programas

Celda
de memoria

00	CRA	X	X
01	ADD	0	6
02	ADD	0	7
03	ADD	0	8
04	STA	0	9
05	HLT	X	X
06	0	1	7
07	0	0	B
08	0	1	C
09			

Explicación

- Limpia el acumulador
- Suma el contenido de 06 al acumulador
- Suma el contenido de 07 al acumulador
- Suma el contenido de 08 al acumulador
- Almacena el contenido del acumulador en la celda 09
- Para
- Operando en 06
- Operando en 07
- Operando en 08
- Al final del programa 03B será almacenado aquí

Memoria con un programa que calcula la suma de tres números.

El programa escrito asignando direcciones simbólicas a las posiciones de memoria a las que debe hacerse referencia

Celda simbólica	Contenido	Comentario
	CRA	Limpia el acumulador
	ADD W	Suma el contenido de W al acumulador
	ADD X	Suma el contenido de X al acumulador
	ADD Y	Suma el contenido de Y al acumulador
	STA Z	Almacena el contenido del acumulador en la celda Z
	HLT	Para
W	017	Operando en W
X	00B	Operando en X
Y	01C	Operando en Y
Z	XXX	Celda para almacenar el resultado

Programa para la resta

Rótulo	Contenido	Comentario	Código	Comentario	Código
	CRA	Limpia el acumulador		CRA	
	ADD SUB	Suma el sustraendo al acumulador		ADD SUB	
	CTA	Complementa el acumulador		CTA	
	ITA	Incrementa el acumulador		ITA	
	ADD MIN	Suma el minuendo al acumulador		ADD MIN	
	STA DIF	Almacena en la celda rotulada «DIF»		STA DIF	
	HLT	Para		HLT	
SUB	09C	Sustraendo en la celda rotulada «SUB»			
MIN	0B7	Minuendo en la celda rotulada «MIN»			
DIF	XXX	Celda para almacenar la diferencia (la diferencia será $0B7 - 09C = 01B$)			

Programa para la resta involucrando dos sustraendos

Rótulo	Contenido	Comentario	Código	Comentario	Código
	CRA	Limpia el acumulador		CRA	
	ADD SUB(1)	Suma al acumulador el contenido de la celda SUB(1) (sustraendo 1)		ADD SUB(1)	
	CTA	Complementa el acumulador		CTA	
	ITA	Incrementa el acumulador		ITA	
	STA NSUB(1)	Almacena en la celda NSUB(1) (-sustraendo 1)		STA NSUB(1)	
	CRA	Limpia el acumulador		CRA	
	ADD SUB(2)	Suma sustraendo (2) al acumulador		ADD SUB(2)	
	CTA	Complementa el acumulador		CTA	
	ITC	Incrementa el acumulador		ITC	
	ADD MIN	Suma al acumulador el contenido de la celda MIN (minuendo)		ADD MIN	
	ADD NSUB(1)	Suma el contenido de la celda NSUB(1)		ADD NSUB(1)	
	STA RES	Almacena en la celda RES (resultado)		STA RES	
	HLT	Para		HLT	
MIN	0B7	Minuendo en la celda MIN			
SUB(1)	09C	Sustraendo 1 en la celda SUB(1)			
SUB(2)	005	Sustraendo 2 en la celda SUB(2)			
NSUB(1)	XXX	Celda para almacenar el negativo de SUB(1)			
RES	XXX	Celda para almacenar el resultado			

Programa para sumar un gran número de sumandos

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)	

