

UNIDAD DE PROCESAMIENTO DE CS1.

Microoperaciones RT.

- En la tabla se resumen microoperaciones de transferencia que la UP puede satisfacer, y que son suficientes para realizar el conjunto de las cuatro instrucciones del computador simple.
- En la fila (*) se muestra una transferencia múltiple de dos transferencias simples que pueden realizarse en paralelo en un mismo ciclo máquina.
- En color amarillo se indican las señales de control a activar al valor "1" para ejecutar cada microoperación

Microoperacion es RT	Señales de control											
	PC		MAR		IR	RT	AC		ALU		Memoria Principal	
	CLPC	IPC	TIR	TPC	TB	WT	WAC	RAC	sumar	restar	R	W
PC=0	1	0	0	0	0	0	0	0	-	-	0	0
(*) PC←PC+1 IR←M[MAR]	0	1	0	0	1	0	0	0	-	-	1	0
MAR←PC	0	0	0	1	0	0	0	0	-	-	0	0
MAR←IR5-0	0	0	1	0	0	0	0	0	-	-	0	0
RT←M[MAR]	0	0	0	0	0	1	0	0	-	-	1	0
M[MAR]←AC	0	0	0	0	0	0	0	1	-	-	0	1
AC←AC+RT	0	0	0	0	0	0	1	0	1	0	0	0
AC←AC-RT	0	0	0	0	0	0	1	0	0	1	0	0

UNIDAD DE PROCESAMIENTO DE CS1.

CS1: MICROOPERACIONES

Búsqueda

1	MAR ← PC (TPC)
2	IR ← RAM; PC ← PC + 1 (R, TB, IPC)

NOTA: Basado en [DIA09]

Ejecución

	STOP IR _{7,6} =00	ADD A IR _{7,6} =01	SUB A IR _{7,6} =10	STA A IR _{7,6} =11
3	NOP	MAR ← IR (TIR)		
4	-	RT ← RAM (R, WT)		RAM ← AC (RAC, W)
5	-	AC ← AC + RT (s, WAC)	AC ← AC - RT (r, WAC)	-

Microoperacion es RT	Señales de control										
	PC		MAR		IR	RT	AC		ALU		Memoria Principal
	CLPC	IPC	TIR	TPC	TB	WT	WAC	RAC	sumar	restar	R W
PC=0	1	0	0	0	0	0	0	0	-	-	0 0
(*) PC←PC+1 IR←M[MAR]	0	1	0	0	1	0	0	0	-	-	1 0
MAR←PC	0	0	0	1	0	0	0	0	-	-	0 0
MAR←IR5-0	0	0	1	0	0	0	0	0	-	-	0 0
RT←M[MAR]	0	0	0	0	0	1	0	0	-	-	1 0
M[MAR]←AC	0	0	0	0	0	0	0	1	-	-	0 1
AC←AC+RT	0	0	0	0	0	0	1	0	1	0	0 0
AC←AC-RT	0	0	0	0	0	0	1	0	0	1	0 0

EJEMPLO DE PROGRAMA CS1.

Programa en ensamblador (\$DirDato en hexadecimal)	Descripción RT del programa ProgSUMA2	Instrucción en binario		Instrucción en hexadecimal
		CO 2 bits	Dirección del dato en binario con 6 bits	
STA \$28	$M(\$DirDato \text{ de } datAux=28) \leftarrow AC$	11	10 1000	E8
SUB \$28	$AC \leftarrow AC - M(\$DirDato \text{ de } datAux=\$28)$	10	10 1000	A8
ADD \$30	$AC \leftarrow AC + M(\$DirDato \text{ de } dat1=\$30)$	01	11 0000	70
ADD \$31	$AC \leftarrow AC + M(\$DirDato \text{ de } dat2=\$31)$	01	11 0001	71
STA \$32	$M(\$DirResultado = 32) \leftarrow AC$	11	11 0010	F2
STOP	Fin ejecución	00	-----	00

Programa **ProgSUMA2**.

Ensamblador (\$DirDato en hexadecimal)	Descripción RT	Formato de la Instrucción en binario	
		CO	Dirección del Dato en binario
STOP	Fin ejecución	00	X X X X X
ADD \$DirDato	$AC \leftarrow AC + M(\$DirDato)$	01	$A_5 A_4 A_3 A_2 A_1 A_0$
SUB \$DirDato	$AC \leftarrow AC - M(\$DirDato)$	10	$A_5 A_4 A_3 A_2 A_1 A_0$
STA \$DirDato	$M(\$DirDato) \leftarrow AC$	11	$A_5 A_4 A_3 A_2 A_1 A_0$

Ciclo	Num. Inst. (0-6)	Fase (CAP/EJ)	PC	IR	MAR	AC	RT	M(\$28)	M(\$32)	INSTRUCCIÓN
1	0	CAP	00	00	00	B5	00	00	00	STA \$28
2	0	CAP	01	E8	00	B5	00	00	00	STA \$28
3	0	EJ	01	E8	28	B5	00	00	00	STA \$28
4	0	EJ	01	E8	28	B5	00	B5	00	STA \$28
5	1	CAP	01	E8	01	B5	00	B5	00	SUB \$28
6	1	CAP	02	A8	01	B5	00	B5	00	SUB \$28
7	1	EJ	02	A8	28	B5	00	B5	00	SUB \$28
8	1	EJ	02	A8	28	B5	B5	B5	00	SUB \$28
9	1	EJ	02	A8	28	00	B5	B5	00	SUB \$28
10	2	CAP	02	A8	02	00	B5	B5	00	ADD \$30
11	2	CAP	03	70	02	00	B5	B5	00	ADD \$30
12	2	EJ	03	70	30	00	B5	B5	00	ADD \$30
13	2	EJ	03	70	30	00	A5	B5	00	ADD \$30
14	2	EJ	03	70	30	A5	A5	B5	00	ADD \$30
15	3	CAP	03	70	03	A5	A5	B5	00	ADD \$31
16	3	CAP	04	71	03	A5	A5	B5	00	ADD \$31
17	3	EJ	04	71	31	A5	A5	B5	00	ADD \$31
18	3	EJ	04	71	31	A5	02	B5	00	ADD \$31
19	3	EJ	04	71	31	A7	02	B5	00	ADD \$31
20	4	CAP	04	71	04	A7	02	B5	00	STA \$32
21	4	CAP	05	F2	04	A7	02	B5	00	STA \$32
22	4	EJ	05	F2	32	A7	02	B5	00	STA \$32
23	4	EJ	05	F2	32	A7	02	B5	A7	STA \$32
24	6	CAP	05	F2	05	A7	02	B5	A7	STOP
25	6	CAP	06	00	05	A7	02	B5	A7	STOP
26	6	EJ	06	00	05	A7	02	B5	A7	STOP

	SUPONIENDO INICIALMENTE		AC = B5							
			RT = 00							
			M(30) = A5							
			M(31) = 02							