

Programas usando el lenguaje ensamblador.

Ejemplo1. Vamos a realizar un programa que genere un contador a partir del numero 7 y se incremente de uno en uno. Observemos el diagrama de flujo de la ilustración 1.

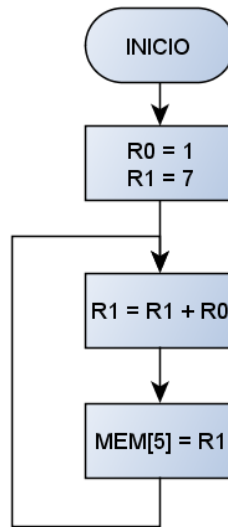


Ilustración 1 Algoritmo de un contador

El programa que implementa el diagrama de flujo de la ilustración 1 se muestra en la tabla 1.

Instrucciones	Significado	Dirección	24...20	19...16	15...12	11...8	7...4	3...0	T
LI R0, #1	R0 = 1	0	00001	0000	0000	0000	0000	0001	I
LI R1, #7	R1 = 7	1	00001	0001	0000	0000	0000	0111	I
CICLO: ADD R1, R1, R0	R1 = R1 + R0	2	00000	0001	0001	0000	xxxx	0000	R
SWI R1, 5	Mem[5] = R1	3	00011	0001	0000	0000	0000	0101	I
B CICLO	goto CICLO (PCx = 2)	4	10011	xxxx	0000	0000	0000	0010	J

Tabla 1 Programa del contador

Ejemplo 2. Vamos a realizar un programa que obtenga los primeros 12 términos de la serie de Fibonacci. Los valores iniciales de la serie son 0 y 1 y se colocan $R0 = 0$ y $R1 = 1$. Cada término de la serie se obtiene sumando los 2 números anteriores. Además, el término de la serie calculado debe colocarse en la dirección de memoria 72. Observemos la solución mostrada en el diagrama de flujo de la ilustración 2.

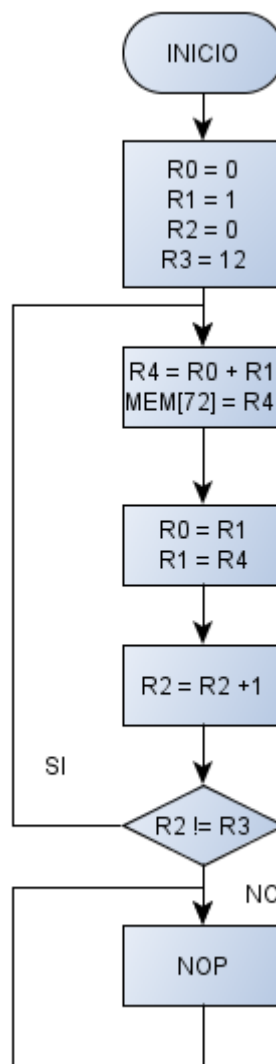


Ilustración 2 Algoritmo de la serie de Fibonacci

El programa que implementa el diagrama de flujo de la ilustración 2 se muestra en la tabla 2.

Instrucciones	Significado	Dir	24...20	19...16	15...12	11...8	7...4	3...0	T
LI R0, #0	R0 = 0	0	00001	0000	0000	0000	0000	0000	I
LI R1, #1	R1 = 1	1	00001	0001	0000	0000	0000	0001	I
LI R2, #0	R2 = 0	2	00001	0010	0000	0000	0000	0000	I
LI R3, #10	R3 = 10	3	00001	0011	0000	0000	0000	1010	I
SERIE: ADD R4, R0, R1	R4 = R0 + R1	4	00000	0100	0000	0001	xxxx	0000	R
SWI R4, 72	MEM[72] = R4	5	00011	0100	0000	0000	0100	1000	I
ADDI R0, R1,#0	R0 = R1 + 0	6	00101	0000	0001	0000	0000	0000	I
ADDI R1, R4,#0	R1 = R4 + 0	7	00101	0001	0100	0000	0000	0000	I
ADDI R2, R2,#1	R2 = R2 + 1	8	00101	0010	0010	0000	0000	0001	I
BNEI R2, R3, SERIE	If(R2 != R3) goto SERIE (PCx = PCx + -5, = 9 + -5 = 4)	9	01110	0010	0011	1111	1111	1011	I
CICLO: NOP		A	10110	xxxx	xxxx	xxxx	xxxx	xxxx	
B CICLO	goto CICLO (PCx = A)	B	10011	xxxx	0000	0000	0000	1010	J

Tabla 2 Programa de la serie de Fibonacci.

Ejemplo 3. Vamos a realizar un programa que compare 3 números colocados en los registros R0, R1 y R2 y guarde el mayor de los 3 en la dirección de memoria 0x20. Considere los 3 números diferentes. Observemos la solución mostrada en el diagrama de flujo de la ilustración 3.

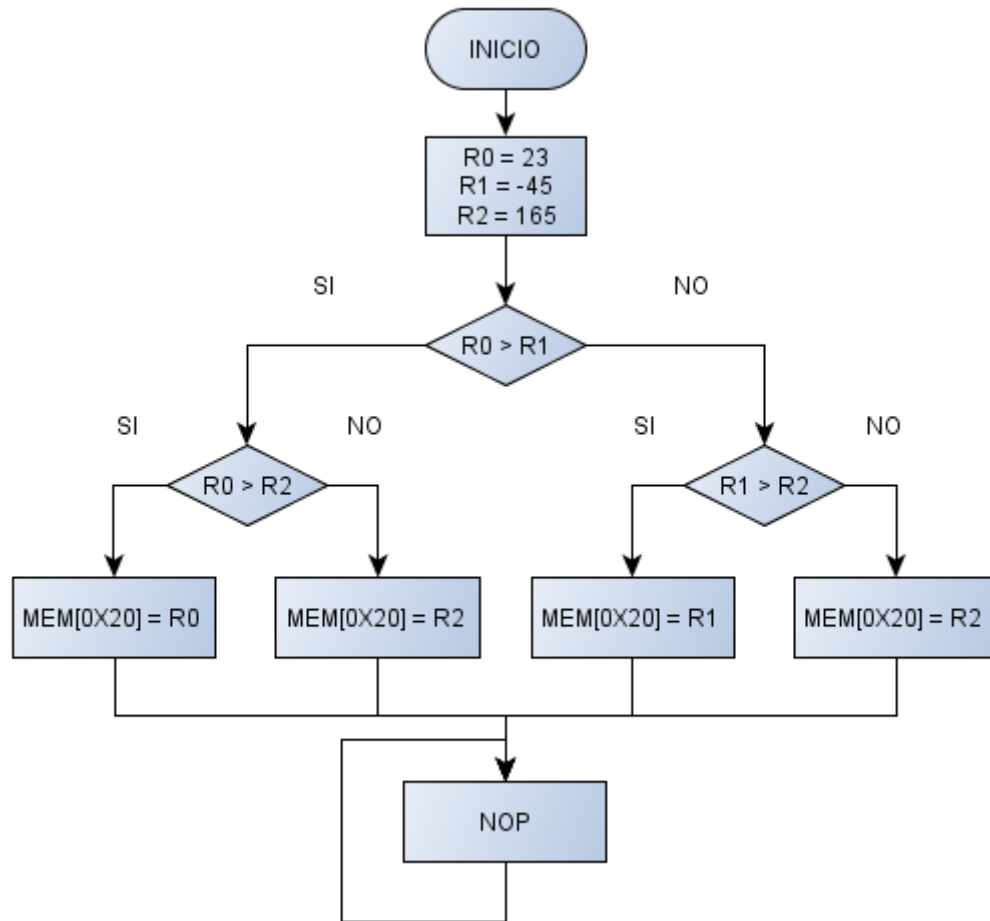


Ilustración 3 Algoritmo 1 de comparación.

En esta solución existen 3 comparaciones en el programa, pero los valores originales de los registros permanecen sin modificarse. El programa que implementa el diagrama de flujo de la ilustración 3 se muestra en la tabla 3.

Instrucciones	Significado	Dir	24...20	19...16	15...12	11...8	7...4	3...0	T
LI R0, #23	R0 = 23	0	00001	0000	0000	0000	0001	0111	I
LI R1, #-45	R1 = -45	1	00001	0001	1111	1111	1101	0011	I
LI R2, #165	R2 = 165	2	00001	0010	0000	0000	1010	0101	I
BGTI R1, R0, CR0R2	if(R0>R1) goto CR0R2 (PCx = PCx + 4, = 3 + 4 = 7)	3	10001	0001	0000	0000	0000	0100	I
BLTI R2, R1, R2MAY	If(R1 < R2) goto R2MAY (PCx = PCx + 6, = 4 + 6 = A)	4	01111	0010	0001	0000	0000	0110	I
SWI R1, 0X20	Mem[0x20] = R1	5	00011	0001	0000	0000	0010	0000	I
B CICLO	goto CICLO (PCx = B)	6	10011	xxxx	0000	0000	0000	1011	J
CR0R2: BGTI R2, R0, R0MAY	If(R0 > R2) goto R0MAY (PCx = PCx + 3, = 7 + 3 = A)	7	10001	0010	0000	0000	0000	0011	I
SWI R2, 0X20	Mem[0x20] = R2	8	00011	0010	0000	0000	0010	0000	I
B CICLO	goto CICLO (PCx = B)	9	10011	xxxx	0000	0000	0000	1011	J
R0MAY: SWI R0, 0X20	Mem[0x20] = R0	A	00011	0000	0000	0000	0010	0000	I
CICLO: NOP		B	10110	xxxx	xxxx	xxxx	xxxx	xxxx	
B CICLO	goto CICLO (PCx = B)	C	10011	xxxx	0000	0000	0000	1011	J

Tabla 3 Programa 1 de comparaciones.

Otra solución consiste en un algoritmo que solo realiza dos comparaciones, pero en este algoritmo se pierde el valor de uno de los registros, por lo que si existen instrucciones que requieren el valor de este registro no se podrán ejecutar. Observemos la solución mostrada en el diagrama de flujo de la ilustración 4.

El programa que implementa el diagrama de flujo de la ilustración 4 se muestra en la tabla 4.

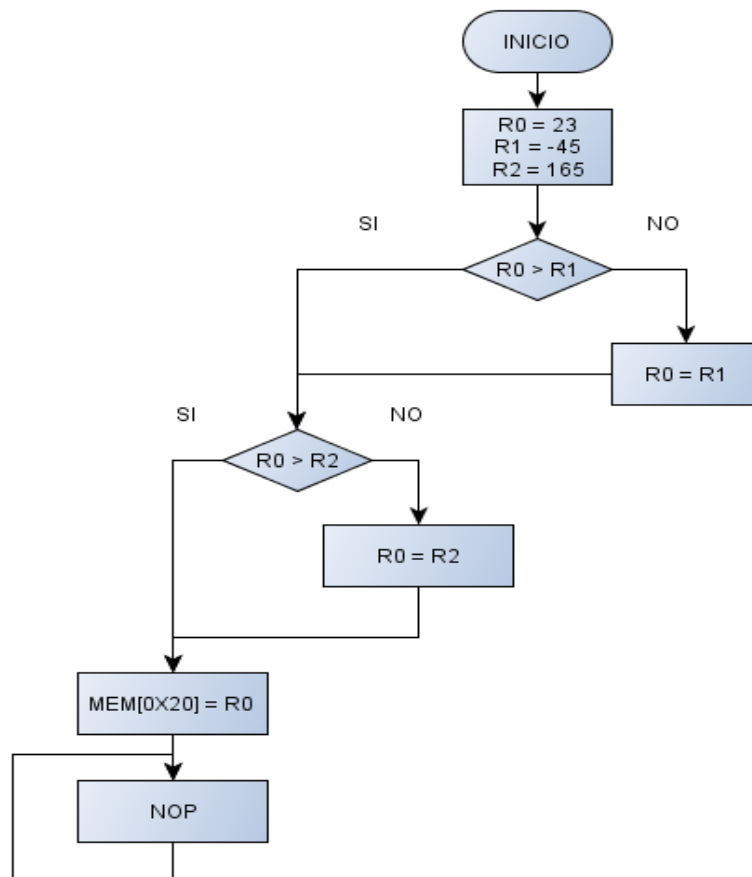


Ilustración 4 Algoritmo 2 de comparación.

Instrucciones	Significado	Dir	24...20	19...16	15...12	11...8	7...4	3...0	T
LI R0, #23									
LI R1, #-45									
LI R2, #165									
BGTI R0, R1, CR0R2									
ADD R0, R1, #0									
CR0R2: BGTI R0, R2, R0MAY									
ADD R0, R2, #0									
R0MAY: SWI R0, 0X20									
CICLO: NOP									
B CICLO									

Tabla 4 Programa 2 de comparaciones.

Ejemplo 4. Vamos a realizar un programa que permita calcular el área de un triángulo usando funciones. Para realizar este programa necesitamos implementar la operación de multiplicación usando sumas sucesivas. Observemos el diagrama de flujo de la ilustración 5.

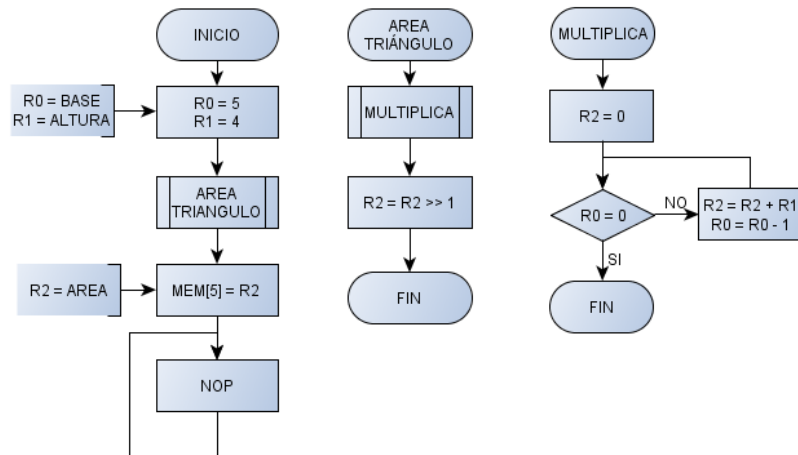


Ilustración 5 Cálculo de área de un triángulo.

El programa que implementa el diagrama de flujo de la ilustración 5 se muestra en la tabla 5.

Instrucciones	Significado	Dirección	24...20	19...16	15...12	11...8	7...4	3...0	T
LI R0, #0	R0 = 0	0	00001	0000	0000	0000	0000	0000	I
LI R1, #0	R1 = 0	1	00001	0001	0000	0000	0000	0000	I
CALL SUMA	SP++ PC(SP) = 6	2	10100	xxxx	0000	0000	0000	0110	J
SWI R1, 5	Mem[5] = R1	3	00011	0001	0000	0000	0000	0101	I
CICLO:	NOP	4	10110	xxxx	xxxx	xxxx	xxxx	xxxx	
	B CICLO	5	10011	xxxx	0000	0000	0000	0100	J
SUMA: SWI R1, 5	Mem[5] = R1	6	00011	0001	0000	0000	0000	0101	I
CALL DATOS	SP++ PC(SP) = 10	7	10100	xxxx	0000	0000	0000	1010	J
ADD R1, R1, R0	R1 = R1 + R0	8	00000	0001	0001	0000	xxxx	0000	R
RET	SP -- PC(SP) ++	9	10101	xxxx	xxxx	xxxx	xxxx	xxxx	
DATOS: LI R0, #1	R0 = 1	10	00001	0000	0000	0000	0000	0001	I
LI R1, #7	R1 = 7	11	00001	0001	0000	0000	0000	0111	I
RET	SP -- PC(SP) ++	12	10101	xxxx	xxxx	xxxx	xxxx	xxxx	

Tabla 5 Programa con función.