

1

PT3. CS
CURSO DE PROGRAMACION DEL COMPU-
TADOR DIGITAL "E R -56"
Publicación Técnica Interna
Sección Computadores y Servomecanismos
I. I. E. E.
UNIVERSIDAD DE CHILE

GUILLERMO GONZALEZ R.
Casilla 2777
Santiago - Chile

Ed. 5
120 Ejs.
15.11.61

P R E F A C I O

El presente curso tiene por finalidad el dar a conocer tanto la programación como la estructura general del Computador ER-56 que ha sido adquirido por la Universidad de Chile.

Las personas que asistan regularmente a clases (especialmente a las primeras ocho) quedarán en condiciones de hacer programas para resolver muchos de los problemas que se presentan en sus respectivas actividades, y que pueden ser resueltos con grandes ventajas mediante el uso del Computador.

A medida que avance el curso se irán repartiendo apuntes que podrán agregarse a esta publicación. De todas maneras, es necesario contar con las siguientes publicaciones del "IIEE" o publicaciones internas de la Sección Computadores y Servomecanismos del I.I.E.E.

PT7.CS "Código y Descripción Resumida del Computador Digital ER-56"

PT2.CS "Programas y Subrutinas como Operadores Matemáticos"
IIEE-9-60 "Circuitos de Computación"

Como complemento o ejemplo de programas, es conveniente tener las siguientes publicaciones:

IIEE-4-60 "Programación Automática"

IT1. CS "Programa para Solución de una Red Eléctrica por medios Topológicos"

IT3. CS "Programa para obtener la respuesta de frecuencia a partir de la respuesta transiente"

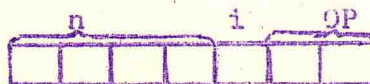
GUILLERMO GONZALEZ REES
Ing. Jefe
Sección

Computadores y Servomecanismos

EXPLICACION Y TRADUCCION DE TERMINOS USADOS CON FRECUENCIA EN EL CAMPO

DE LOS COMPUTADORES DIGITALES

Palabra (word)	Conjunto de digitos que representan un dato o una instrucción. En el ER-56 las palabras tienen 7 cifras decimales. En algunos casos se trabaja con palabras dobles de 14 cifras que se forman por juxtaposición de dos palabras de 7 cifras.
Celda (cell)	Elemento de la memoria en que se guarda la palabra.
Memoria (Memory, storage)	lugar donde se almacena la información.
Dirección (address)	número asignado a una celda. En el ER-56 la dirección se designa por "n" y el contenido de esa celda por "(n)".
Unidad aritmética (arithmetic unit)	Sector del computador donde se realizan las operaciones principales.
Registro (register)	unidad capaz de guardar un conjunto de cifras.
Acumulador (accumulator)	registro de la unidad aritmética en donde aparece el resultado de las operaciones. En el ER-56 el acumulador tiene una capacidad de 2 palabras, o sea, de 14 cifras decimales.
Instrucción (instruction)	la palabra que contiene los datos necesarios para que realice cierta operación. En el ER-56 la palabra de instrucción tiene el aspecto siguiente:



n : 4 primeras cifras que, en general, indican la dirección a la cual se refiere la instrucción. Si no indican dirección se designa por p.

i : número de registro de índice modificador de esta instrucción. Si $i = 0$ no hay modificación.

OP : Operación.

Por ejemplo : 0382 0 35 significa :

Sumar al Acumulador el contenido de la celda 0382 y colocar el resultado en el Acumulador. Simbólicamente esto se expresa como sigue :

$$(A) + (n) \longrightarrow A$$

$$(A) + (0382) \longrightarrow A$$

El contenido de A, se reemplaza por el contenido de A + el contenido de n.

Registro de índice : registro cuyo contenido modifica la parte "n" o "p" (index register) de una instrucción. Se designan por Ji y su acción es tal que

$$n + (Ji) \longrightarrow n$$

Por ejemplo : sea (J4) = 0002 y la instrucción 0382435, que significa

$$(A) + (n) \longrightarrow A$$

como i = 4, se tendrá :

$$n + (J4) \longrightarrow n$$

$$0382 + 0002 = 0384 \longrightarrow n$$

La operación efectiva es, entonces,

$$(A) + (0384) \longrightarrow A$$

Si i = 0, n no se modifica

Registro contador (counter)

Registro de índice especial que contiene la dirección de la celda en que está contenida la instrucción que se va a ejecutar.

En el ER-56 se designa por J9. El contador avanza automáticamente en 1 durante la realización de la instrucción.

Datos
(data)

palabras con las cuales se efectúan las operaciones. Pueden, por ejemplo, representar números o, letras o información estadística codificada. Para ver cómo se representa la información en el EN-56, ver PT7. CS, pág. 2.-

Guardar
(store)

colocar el contenido de una celda o registro en otra celda o registro. Tanto en el EN-56 como en la mayoría de los computadores, el contenido anterior del elemento desaparece. El contenido del elemento que entrega la información no se altera.

Por ejemplo: si $A \longrightarrow n$, el (A) no se altera pero (n) desaparece y es substituido por (A). La flecha \longrightarrow se usa para indicar este reemplazo.

Leer
(read)

Tomar el contenido de un elemento, en general, de una celda de la memoria y colocarlo en algún registro o celda. Para esta operación siguen las consideraciones anteriores sobre permanencia de la información.

Programa
(program)

conjunto de instrucciones que se ejecutan en forma sucesiva.

Subrutina
(subroutine)

subprograma al cual se salta desde el programa principal para calcular alguna función. Por ejemplo, si en el transcurso del programa principal se llega a una etapa donde hay que calcular \sqrt{z} , se pone z en el Acumulador y enseguida se ordena un salto a la celda donde comienza la subrutina.

Esta calcula \sqrt{z} , pone el resultado en A y salta automáticamente al programa principal. (Ver publicación PT2. CS: "Programas y Subrutinas como Operadores Matemáticos")

ETAPAS EN LA REALIZACION DE UNA INSTRUCCION

Para analizar los pasos que sigue el computador en la interpretación y ejecución de una instrucción del programa, estudiemos el caso particular de la instrucción:

n	i	OP
0 3 2 8	0	3 5

En la página 7 del Código vemos que la operación 35 significa:

$$(A) + (\bar{n}) \rightarrow A$$

Es decir, el contenido de A se suma al contenido de la celda 0328 y el resultado se guarda nuevamente en A.

1a. ETAPA.- (Ver figura 1) En la unidad de control hay un CONTADOR cuyo contenido es el número de la celda (dirección) en la que se guarda la instrucción que se va a realizar; en este ejemplo 0127.

La unidad de control selecciona de la memoria de trabajo la celda especificada por el contenido del contador, 0127, y manda el contenido de esa celda al REGISTRO DE INSTRUCCION de la unidad de control.

2a. ETAPA.- (Ver figura 2) El registro de instrucción se conecta a los circuitos decodificadores que interpretan la instrucción.

Uno de los circuitos actúa sobre los 2 últimos dígitos y establece los circuitos necesarios para que se ejecute la operación indicada, en este caso, una suma.

Otro de los circuitos analiza las cuatro primeras cifras y, en este caso, selecciona en la memoria la celda especificada por tales cifras (En este ejemplo la celda 0328).

Enseguida, el contenido de la celda elegida se manda a la unidad aritmética, donde se efectúa la operación de acuerdo con lo

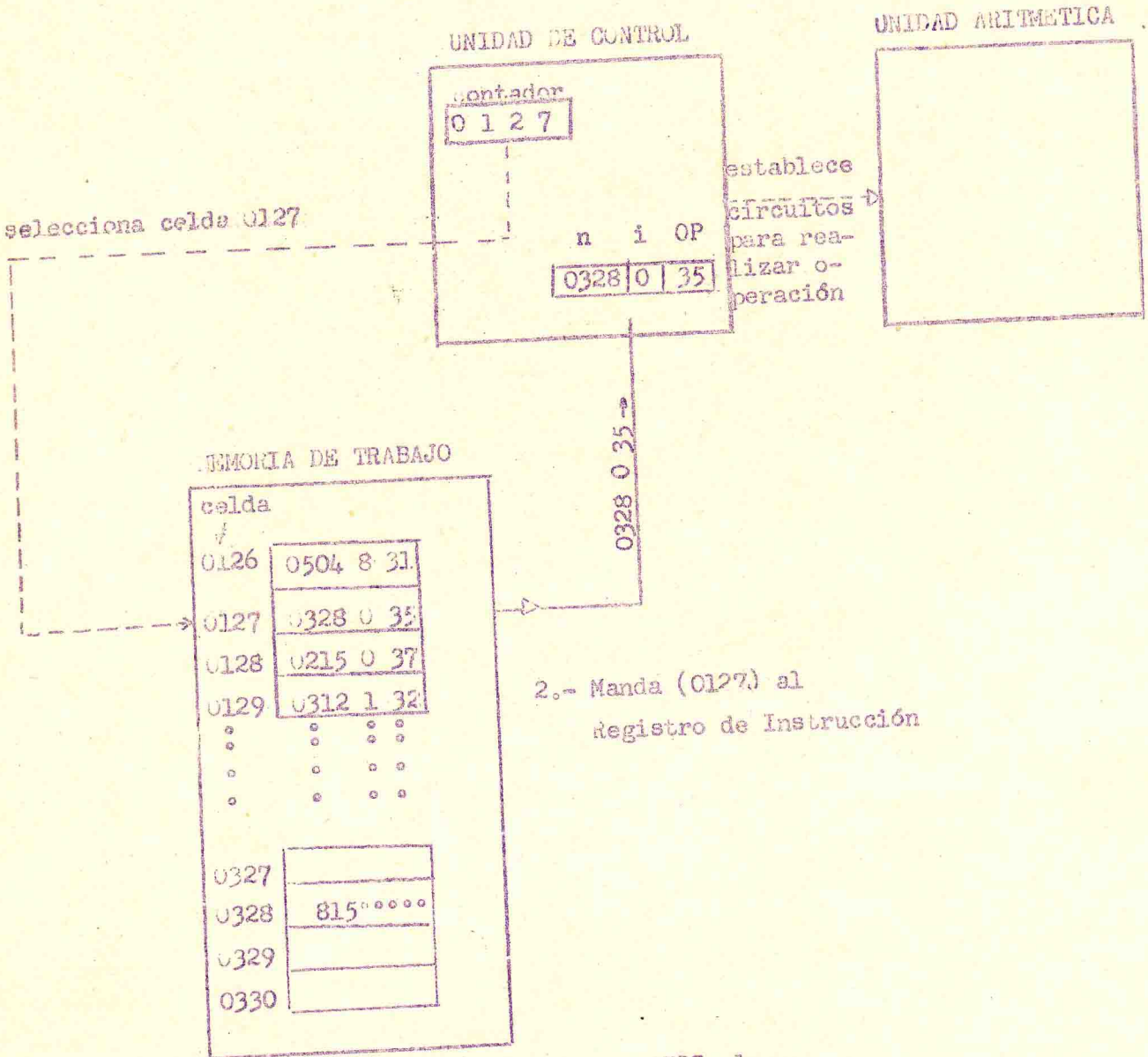


FIG. 1

PRIMERA ETAPA en la realización de una instrucción: interpretación de la instrucción.

(x) = contenido de celda o registro x

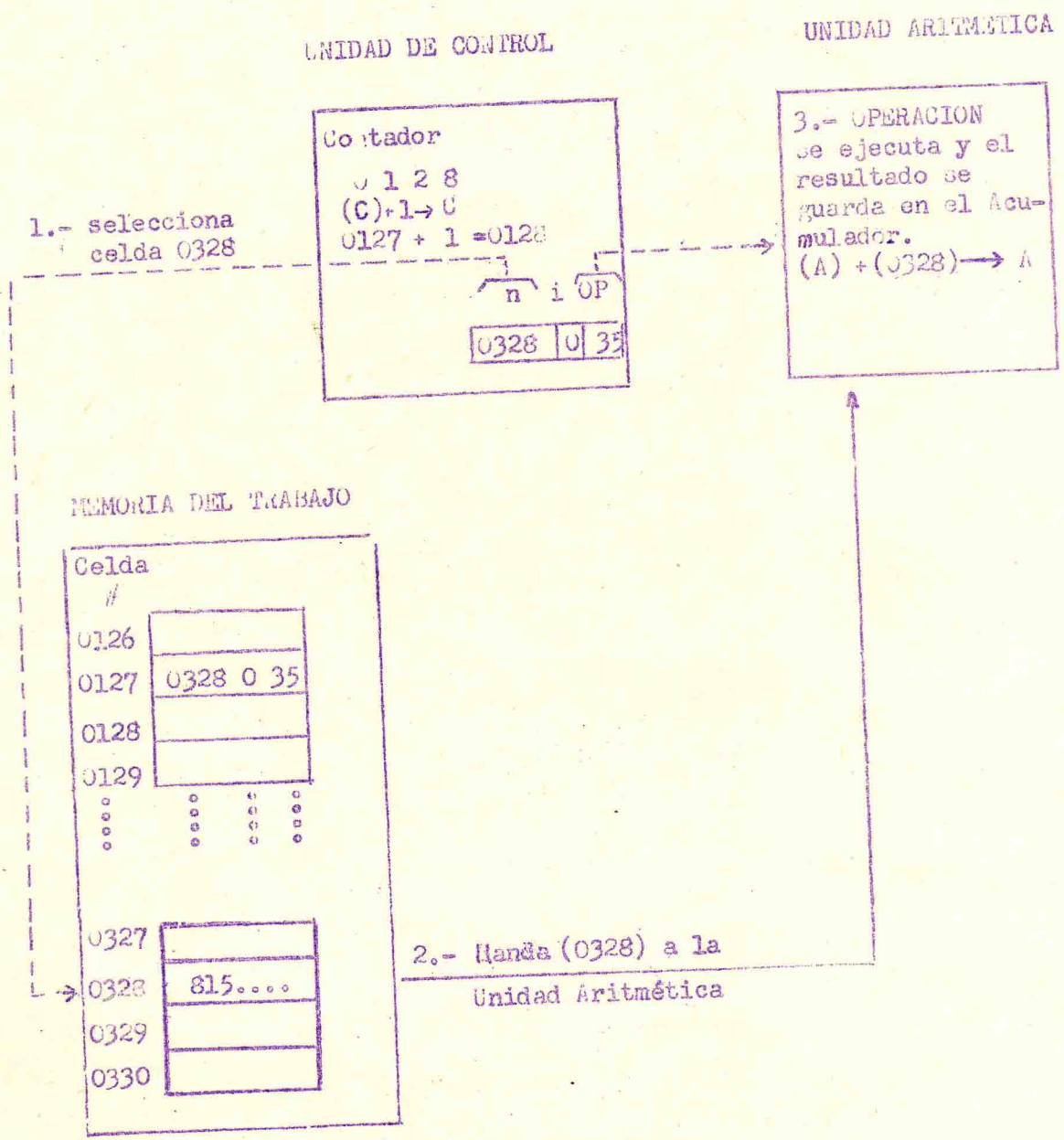


FIG. 2

SEGUNDA ETAPA: Ejecución de la instrucción.

indicado por las cifras de operación.

Como $i=0$ en este ejemplo, no hay modificación de la dirección.

A principio de esta etapa, el contenido del contador avanza en 1 con lo cual queda listo para elegir la instrucción siguiente cuando suceda de nuevo la 1a. etapa.

Una vez ejecutada la operación, se inicia nuevamente la 1a. etapa.

En este computador cualquiera celda de la memoria puede guardar una palabra de datos (un número) o una instrucción. La diferencia en la interpretación de esta palabra depende solamente de en qué etapa se realice la selección de la palabra.

Durante la fase 1 es el registro contador quien selecciona la palabra, que es luego enviada al registro de instrucción donde es interpretada como instrucción.

Si la selección ocurre en la segunda etapa la palabra es interpretada como dato y no como instrucción.

Si $i \neq 0$, a la parte n de la instrucción se le suma el contenido del registro de índice J_i . Enseguida esta instrucción, ya modificada, entra al registro de instrucción.

$$n + (J_i) \longrightarrow n$$

ELEMENTOS DE PROGRAMACION

Para comprender mejor el procedimiento que se usa para resolver problemas con el computador digital, comparémoslo con el procedimiento usado para evaluar una expresión algebraica con una máquina corriente de escritorio.

Sigamos los pasos que se deben realizar en ambos casos para calcular:

$$\begin{aligned}
 f(x) &= x^2 + 3x + 1 && \text{si } x \geq 0 \\
 f(x) &= 0 && \text{si } x < 0
 \end{aligned}$$

Supongamos que en el transcurso de cierto cálculo hay que obtener el valor $f(x)$ en función de una variable x proveniente de cálculos previos.

Si tuviéramos que encargar la solución del problema a una persona que supiera operar la máquina de calcular, pero que no supiera álgebra, se necesitaría lo siguiente :

- 1) Un procedimiento o conjunto de instrucciones para efectuar las diversas operaciones a que dá lugar el cálculo de $f(x)$. En computación digital, esta lista de instrucciones se llama PROGRAMA.
- 2) Un papel en el cual están escritos los valores de "x" y en el que se anotarán también constantes, resultados parciales y el resultado final del cálculo. Esto equivale a la MEMORIA de un computador digital. El programa también se guarda en la memoria.
- 3) Una máquina de calcular para efectuar las operaciones aritméticas. El equivalente en el computador digital es la UNIDAD ARITMETICA.
- 4) Una persona que, operando las teclas de la máquina ejecute lo ordenado por el programa, escriba los resultados y realice las decisiones. En el computador la UNIDAD DE CONTROL está encargada de desempeñar estas funciones.

El computador digital tiene en la unidad aritmética un registro llamado Acumulador que desempeña funciones similares a las del acumulador de una máquina de escritorio. El Acumulador se designa por A.

Para anotar y leer los valores con que se va a operar será necesario disponer de una lista de datos y otra de instrucciones. Estas listas que en el caso del Computador Digital se guardan en la memoria, las descompondremos en la forma que sigue:

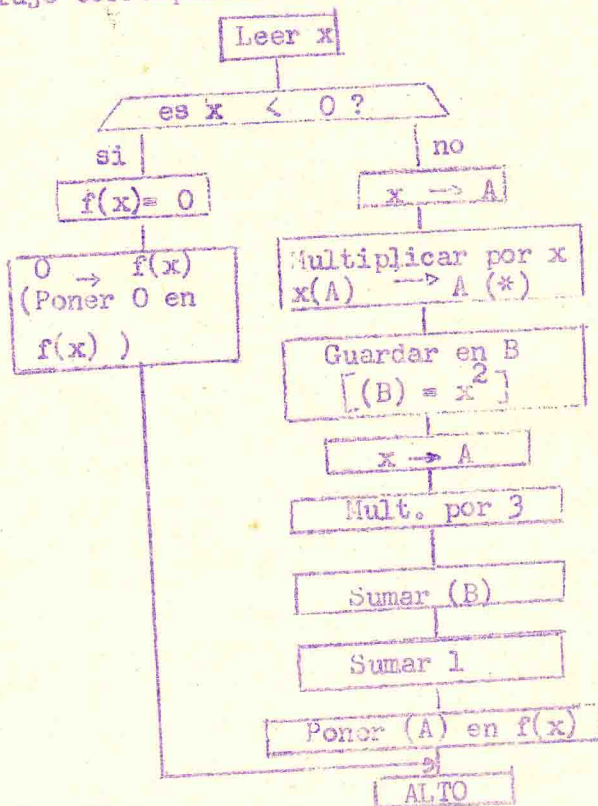
Datos iniciales	
x	

Resultados Intermedios y constantes	
B	
0	
1	
3	

Resultados finales	
f(x)	

Lista de instrucciones	

Diagrama de flujo. - El programa para realizar un cálculo puede ser expresado con ventajas en forma gráfica mediante el "diagrama de flujo". El diagrama de flujo correspondiente al cálculo de $f(x)$ se da a continuación:



Este diagrama de flujo deberá traducirse a una lista de instrucciones como un primer paso para que pueda ser entregado al computador digital. Enseguida habrá que codificar la lista de acuerdo con el código del computador que se va a usar.

(*) x multiplicado por el contenido de A se guarda en A (Ver apuntes anteriores).

Programa (*)

Instrucción		Expresión Simbólica
1	Leer x	
2	es x 0? SI: Tomar instr. siguiente NO: Saltar a instr. #5	
3	Poner 0 en f(x); f(x) = 0	$0 \longrightarrow f(x)$
4	Saltar a instr. #13	S; 13
5	Poner x en el Acumulador	$x \longrightarrow A$
6	Mult. por x $(A) = x^2$	$x(A) \longrightarrow A$
7	Guardar (A) en B $(B) = x^2$	$(A) \longrightarrow B$
8	Poner x en A $(A) = x$	$x \longrightarrow A$
9	Multiplicar por 3 $(A) = 3x$	$3(A) \longrightarrow A$
10	Sumar el contenido de B $(A) = x^2 + 3x$	$(A) + (B) \longrightarrow A$
11	Sumar 1 $(A) = x^2 + 3x + 1$	$(A) + 1 \longrightarrow A$
12	Poner (A) en f(x) $f(x) = x^2 + 3x + 1$	$(A) \longrightarrow f(x)$
13	Alto o enlace con el resto del programa	

(*) (n) = contenido de n. (Ver apuntes anteriores)

En el Computador digital, tanto los datos como las instrucciones se guardan en las celdas de la memoria. Así por ejemplo, habrá una celda para guardar x, una para B, una para guardar el cero, etc y otro conjunto de celdas para guardar el programa.

Observando tanto el diagrama de flujo como el programa veremos que hay conjuntos de instrucciones de carácter netamente aritmético. Por ejemplo, el cálculo de f(x) desde la instrucción número 5 a la número 12 es uno de tales conjuntos.

Codificación para el ER-56.- Codifiquemos la parte del programa comprendida entre la instrucción 5 y la instrucción 12.-

Instrucción # 5 x A; Cargar el Acumulador

En la página 7 del Código vemos que la operación 3l significa

(ñ) A, es decir, el contenido de la celda ñ reemplaza a lo que hay en el Acumulador.

Supongamos que x está guardado en la celda # 0110^(*), entonces la instrucción correspondiente será :

n	i	OP	
0 1 1 0	0	3 1	(0110) → A x → A

Instrucción # 6.- x · (A) → A ; Multiplicación

La Operación 37 significa:

(ñ) · (A) → (A)

De modo que para que (A) = x², la instrucción será :

(0110) · (A) que, en código, es :

n	i	OP
0 1 1 0	0	3 7

Instrucción # 7.- (A) B ; Guardar el contenido del acumulador.

Pongamos B en la celda # 0106. La instrucción será:

n	i	OP	
0 1 0 6	0	3 2	; (A) (0106)

La codificación del resto de las instrucciones hasta la # 12 se hace en forma similar. Para realizar esto necesitamos asignar en la memoria celdas a ciertas cantidades.

Por ahora distribuiremos nuestros datos en forma más o menos arbitraria según la tabla que sigue:

DISTRIBUCION DE LA MEMORIA

Datos iniciales	Resultados intermedios y constantes (*)	Resultados finales	Programa
x = (110)	0 = (0100) 1 = (0102) 3 = (0104) B = (0106)	f(x) = (0114)	Desde celda 0011

Conviene recordar en este punto que si el computador copia (o lee) una información contenida en una celda, o registro, la información permanece idéntica en

(*) Como estamos trabajando en coma flotante, se subentiende que x está en realidad guardado en la doble celda 0110, 0111.

(*) Como se trabaja en coma flotante, es necesario usar 2 celdas para cada dato..

dicha celda o registro. En cambio, si el computador carga (o inscribe), entonces lo que había antes es reemplazado por la nueva información.

Siguiendo con la codificación, tendremos para el cálculo de $f(x) = x^2 + 3x + 1$.

Instr. #	n	i (*)	OP	Expresión simbólica	Observaciones
5	0110	0	31	$x \rightarrow A$	$(A) = x$
6	0110	0	37	$x(A) \rightarrow A$	$(A) = x^2$
7	0106	0	32	$(A) \rightarrow B$	$(B) = x^2$
8	0110	0	31	$x \rightarrow A$	$(A) = x$
9	0104	0	37	$3(A) \rightarrow A$	$(A) = 3x$
10	0106	0	35	$B + (A) \rightarrow A$	$(A) = 3x + x^2$
11	0102	0	35	$1 + (A) \rightarrow A$	$(A) = x^2 + 3x + 1$
12	0114	0	32	$(A) \rightarrow f(x)$	$f(x) = x^2 + 3x + 1$

En el diagrama de flujo vemos que el computador tendrá que realizar una decisión lógica. Según el resultado de la comparación, $x < 0$?, deberá tomar la rama del programa que hace $f(x) = 0$ ó aquella que calcula $f(x) = x^2 + 3x + 1$.

Como se ve en el programa, esta ramificación del diagrama de flujo se traduce en saltos en la secuencia de ejecución del programa.

Normalmente, las instrucciones se guardan en celdas sucesivas (7 cifras).

El computador va seleccionando estas celdas en orden también sucesivo, excepto en aquellos casos cuando, mediante una instrucción de salto, se ordena tomar la instrucción siguiente de otra celda (ver etapas en la realización de una instrucción).

Saltos. Las instrucciones de salto hacen que el computador tome la instrucción siguiente de la celda especificada por la instrucción.

En la página 5 del Código se encuentran las instrucciones de salto. Hay saltos condicionales e incondicionales.

La operación 12 se usa para salto incondicional. Por ejemplo:

n	i	OP
0 0 2 8	0	1 2

(*) Si $i = 0$ no hay modificación por registros de índice.

significa: Tomar la próxima instrucción de la celda $n = 0028$ incondicionalmente.
 Hay veces en que al saltar o no saltar en el programa está condicionado por el resultado de una comparación. En nuestro ejemplo será necesario comparar x con 0 y enseguida dar una instrucción para saltar o no según sea el resultado de esa comparación.

Comparación: Las instrucciones de comparación se encuentran en la página 6 del Código. Por ejemplo la operación 28 hace que (A) se compare con $(ñ)$. El resultado de cualesquiera de estas comparaciones se guarda en unas unidades llamadas indicadores de comparación.

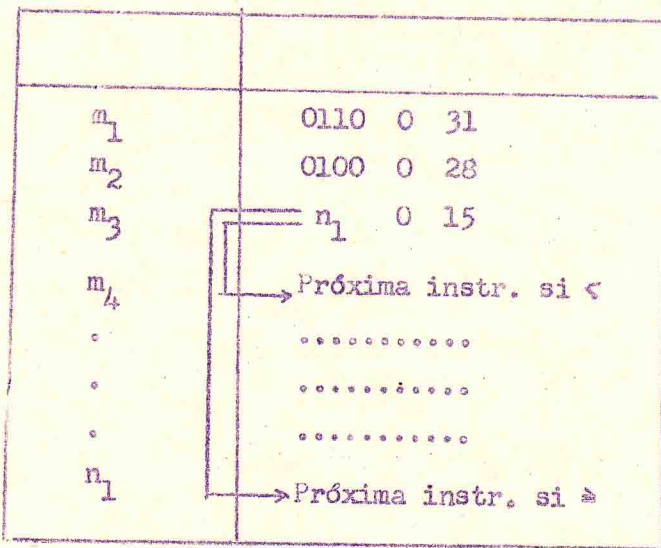
Hay cuatro indicadores de comparación: $=$; \neq ; \geq ; $<$.

El indicador $=$ se conecta si el resultado de la comparación es "igual" por ejemplo, $(A) = (ñ)$

Los indicadores \neq , \geq y $<$ se conectan cuando el resultado de la comparación es \neq , \geq y $<$, respectivamente.

Por ejemplo si resulta que $(A) > (n)$, se conectarán los indicadores \neq y \geq .

De acuerdo con esto, codifiquemos esta parte del programa:



$x \rightarrow A$
 (A) comp. 0
 Salto si \geq ; No saltar si $<$.

Finalmente, estamos en condiciones de hacer el programa completo. Supongamos que todos los datos y el programa se encuentran en la memoria y que el resul-

tado quedará en la memoria.

La primera instrucción indica al computador que debe interpretar los datos como representados en coma flotante.

Instr. #	Celda donde se guarda la instr.	Código			Expresión Simbólica
		"n" ó "p"	l	OP	
0	0011	7200	0	00	Instr. para coma flotante
1	0012	0110	0	31	$x \rightarrow A$
2	0013	0100	0	28	(A) comp. con 0
2A	0014	0018	0	15	Salto \rightarrow , no saltar \leftarrow
3	0015	0100	0	31	$0 \rightarrow A$
3A	0016	0114	0	32	$(A) \rightarrow f(x)$
4	0017	0026	0	12	Salto a instr. 13
5	0018	0110	0	31	$x \rightarrow A$
6	0019	0110	0	37	$x \cdot (A) \rightarrow A$
7	0020	0106	0	32	$(A) \rightarrow B$
8	0021	0110	0	31	$x \rightarrow A$
9	0022	0104	0	37	$3 \cdot (A) \rightarrow A$
10	0023	0106	0	35	$(B) + (A) \rightarrow A$
11	0024	0102	0	35	$(A) + 1 \rightarrow A$
12	0025	0114	0	32	$(A) \rightarrow f(x)$
13	0026	—	—	—	Alto o enlace con resto del programa.

$f(x) = x^2 + 3x + 1$

REGISTROS DE INDICE

Son registros de cuatro cifras decimales que se encuentran en la unidad de control y su contenido modifica la parte "n" ó "p" de una instrucción. El ER-56 tiene 10 registros de índice que se designan por J0, J1, J2,..... J9.

El J0 es un registro de índice especial que no modifica instrucciones. La acción de los registros J1 al J9 es tal que:

$$n + (Ji) \longrightarrow n$$

Por ejemplo : sea (J4) = 0002 y la instrucción, 0382435, que significa:

$$(A) + (n) \longrightarrow A$$

Por acción del registro de índice, antes de que la unidad de control elija el contenido de la celda especificada por la instrucción, se efectúa la operación

$$n + (Ji) \longrightarrow n$$

De esta manera, como $i = 4$, se tendrá :

$$n + (J4) \longrightarrow n$$

$$0382 + 0002 = 0384 \longrightarrow n$$

La operación efectiva es, entonces,

$$(A) + (0384) \longrightarrow A$$

Si $i = 0$, no hay modificación. ($i = 0$ NO indica modificación por J0).

Por ejemplo : 0382 0 35 significa $(0382) + (A) \longrightarrow A$

EJEMPLOS:

- Supongamos que (J1) = 0102
- (J2) = 0532
- (J7) = 0008

Entonces:

INSTRUCCION	SIGNIFICADO
0300 1 12	Tomar próx. instr. de celda 0402
0000 7 18	Tomar próx. instr. de celda 0008 si QE está desconectado
0102 2 28	(A) comparado con (0634)
0004 1 37	(A) . (0106) → (A)

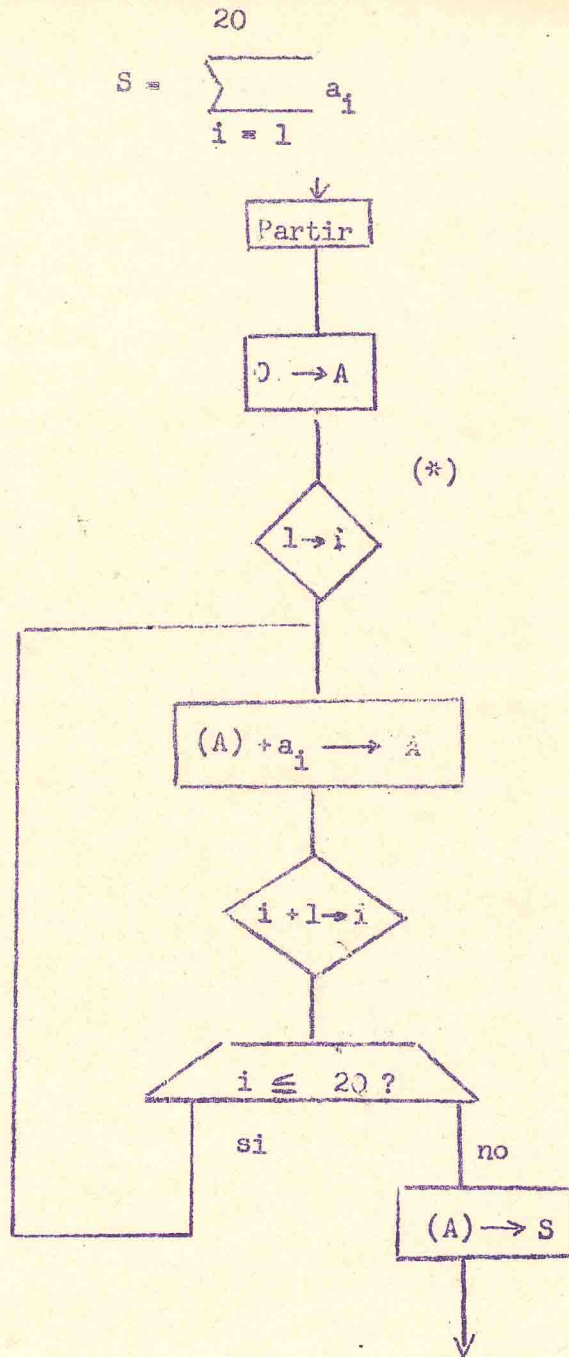
La ventaja que significan los registros de índice se hace aparente en la solución del siguiente problema:

En la memoria de trabajo del computador hay una lista de 20 números en coma flotante, $a_1 \dots a_{20}$, dispuestos en celdas consecutivas. Se pide un programa para sumar dichos números.

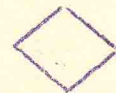
$$S = \sum_{i=1}^{20} a_i$$

El diagrama de flujo de la página 10 sirve para realizar lo indicado. Siguiendo una convención universalmente aceptada, el diagrama de flujo se ha hecho en forma general, pudiendo servir para cualquier computador.

Hagamos ahora el programa usando el código del ER-56.



(*) Las operaciones con índices se encierran en rombos



Usemos el registro de índice J7 y supongamos que:

$$a_1 = (0100)$$

$$a_2 = (0102)$$

.....

$$a_{20} = (0138)$$

$$0 = (0140)$$

$$S = (0142)$$

La operación aritmética $(A) + a_i \longrightarrow A$ se traducirá en :

$$0100 \quad 7 \quad 35,$$

siempre que al principio, $(J7) = 0$ y que por cada vuelta del bucle, $(J7) + 2 \longrightarrow J7$.

La relación $i + 1 \longrightarrow i$ se traduce, entonces, en $(J7) + 2 \longrightarrow J7$.

La comparación $i \leq 20?$ se convierte en $(J7) \leq 40?$, ya que cuando se suma a_{20} , $(J7) = 38$; pero, antes de efectuar la comparación,

$$(J7) + 2 \longrightarrow J7 ; \quad (J7) = 40$$

Operaciones con los registros de índice.— En la página 13 del Código se encuentran las instrucciones necesarias para operar con los registros de índice.

SUMA.— En nuestro ejemplo necesitamos sumar 2 a J7. Para ello podemos usar la operación 93 :

$$(Ji) + p \longrightarrow Ji$$

Es decir $(Ji) +$ la parte p de la instrucción reemplazan a (Ji) .

Por ejemplo:

$$\underbrace{\quad p \quad}_{0002} \quad \underbrace{\quad i \quad}_{7} \quad \underbrace{\quad OP \quad}_{93} \quad \text{significa,}$$

$$(J7) + 0002 \longrightarrow J7$$

Es necesario que aquí las primeras cuatro cifras de esta instrucción no se refieren a ninguna celda, sino que se interpretan sencillamente como un número entero de cuatro cifras decimales.

En algunas ocasiones conviene emplear la operación 92 para realizar esta suma.

$$92 : \quad (J_i) + (n)^* \longrightarrow J_i$$

(n)* indica las primeras cuatro cifras del contenido de la celda n.

Por ejemplo sea $n = 327$, $(n) = 0002000$, y la instrucción : 0327692,-

Entonces, $(J_6) + 0002 \longrightarrow J_6$

Aquí sucede efectivamente que las primeras cuatro cifras representan una dirección n.

COMPARACION: Se hace en forma parecida a la comparación con el Acumulador.

La operación 98, por ejemplo, hace que se compare (J_i) con el p de la instrucción y se operan los mismos indicadores de comparación que ya se habían estudiado.

Por ejemplo:

0040 7 98

significa : Comparar (J_7) con 0040 y operar los indicadores de comparación. Todas las operaciones de los grupos 80 y 90 se hacen en la unidad de control, sin interferir con la Unidad Aritmética.

Las operaciones 80, ..., 88 se refieren al registro J8, quedando libre la parte i de la instrucción para permitir una modificación por otro registro de índice.

Por ejemplo :

$$\begin{array}{l} \text{0010 2 83} \quad \text{significa :} \\ \{0010 + (J_2)\} + (J_8) \longrightarrow J_8 \quad (*) \\ \text{si} \quad (J_2) = 0002, \\ \quad \quad 0012 + (J_8) \longrightarrow J_8 \end{array}$$

(*) Solo los paréntesis redondos significan "contenido de"

El resto de las instrucciones relativas a los registros de índice son fáciles de comprender teniendo en vista lo que ya se ha explicado.

Programa para encontrar $S = \sum_{i=1}^{20} a_i$

Estamos ahora en condiciones de hacer el programa.

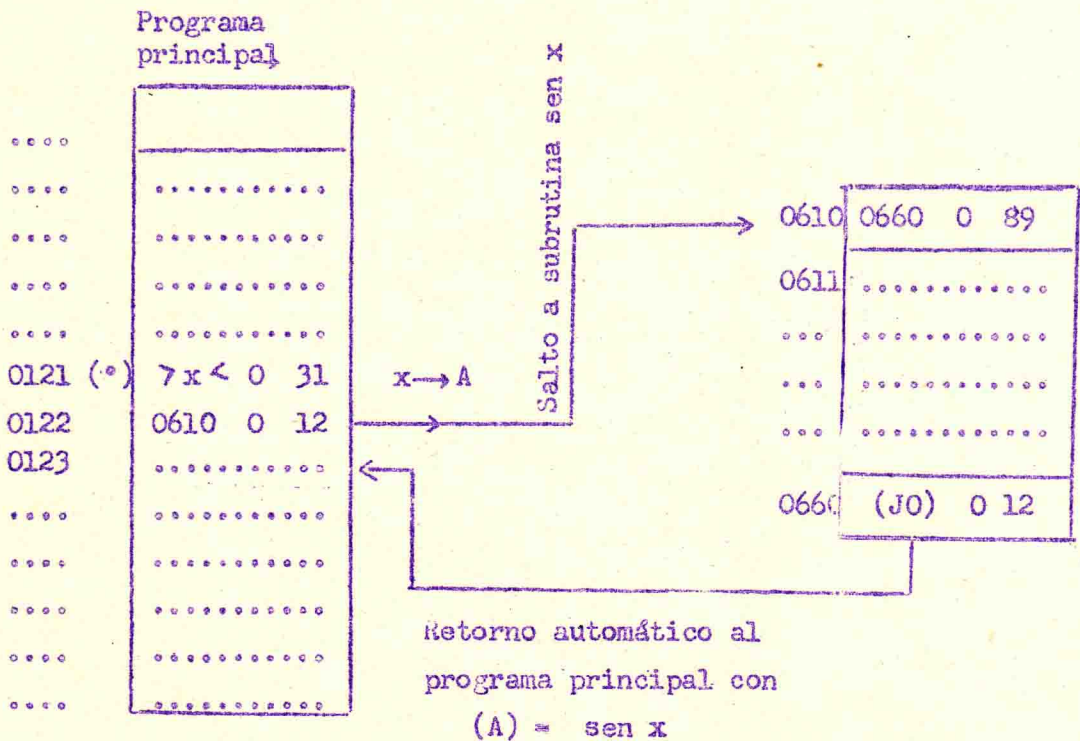
Celda de instrucción		Explicación
507	7200 0 00	Coma flotante
508	0140 0 31	$0 \longrightarrow A$
509	0000 1 91	$0 \longrightarrow J1$
→ 510	0100 1 35	$(A) + (0100 + (J1)) \longrightarrow A$
511	0002 1 93	$(J1) + 2 \longrightarrow J1$
512	0040 1 98	$(J1) \text{ comp. } 40$
← 513	0510 0 16	Salto si <
514	0142 0 32	$(A) \longrightarrow S$
.....	Alto o enlace con resto del programa

SUBROUTINAS (*)

Las subrutinas son programas generales que se preparan una sola vez y sirven para ser usados en combinación con el programa principal.

Por ejemplo, todas las funciones elementales se encuentran en forma de subrutinas. Si en el transcurso de un programa para el ER-56 es necesario obtener $\text{sen } x$, por ejemplo, entonces se pone x en el Acumulador y se da una instrucción de salto a la celda donde hemos guardado la subrutina $\text{sen } x$. El computador calcula $\text{sen } x$ con el programa de la subrutina y salta nuevamente al programa principal en forma automática, llevando $\text{sen } x$ en el acumulador.

Este procedimiento aparece esquematizado en la figura siguiente:



(*) Ver PT2. CS: "Programas y Subrutinas como Operadores Matemáticos".
(°) > x < significa "dirección de x"

Datos para la subrutina.- En el ejemplo citado, el único dato que se entrega a la subrutina es x. Sin embargo, hay otros casos, como ocurre con la subrutina para calcular

$$P_n(x) = \sum_{i=0}^n a_i x^i$$

en que es necesario entregar varios datos. Esto se verá en detalle más adelante en algunos ejemplos.

Uso de las subrutinas.- Veamos un ejemplo sencillo para ver cómo se hace uso de las subrutinas :

Ejemplo: hacer un programa para calcular

$$z = \text{Log} \sqrt{1 + \text{sen}^2 w}$$

Necesitaremos las subrutinas "Log", " $\sqrt{\quad}$ " y "sen". En los apuntes vemos cuántas celdas son necesarias para cada subrutina y el tiempo máx. que van a demorar en el cálculo de la función. Además aparece indicado cuales registros de índice va a ocupar la subrutina. Tendremos que cuidar de que sus contenidos hayan sido guardados previamente si es que pretendemos seguir usándolos después. En general, para las subrutinas se emplean los registros J8, J7, J6,..., es ese orden. De manera que en el programa principal conviene usar los registros J1, J2, J3,.. para evitar un posible conflicto. La información obtenida de los apuntes podemos resumirla en el siguiente cuadro

	Log x	\sqrt{x}	sen x
Celdas	85	72	80
T _{máx.}	33,8 ms	21,8 ms	24,2
Ji necesarios	J7, J8	J8	J6, J7, J8
Observaciones			(J6) = 0 para obtener sen x

Distribución de la memoria.-

Empecemos ubicando las subrutinas a partir de 0001.

Ubicación	Log x	\sqrt{x}	sen x
Desde	0001	0086	0158
Hasta	0085	0157	0237

w = (0250)
 sen w = (0252)
 l = (0254)
 z = (0256)

Programas: a partir de 0500

PROGRAMA

0500	7200	0	00	Coma flotante
0501	0250	0	31	w \longrightarrow A
0502	0000	6	91	0 \longrightarrow J6
0503	0158	0	12	Salto a SR sen
0504	0252	0	32	(A) = sen w \longrightarrow 0252
0505	0252	0	37	sen ² w \longrightarrow A
0506	0254	0	35	l + sen ² w \longrightarrow A
0507	0086	0	12	Salto a SR $\sqrt{\quad}$ [A] = $\sqrt{1 + \text{sen}^2 w}$
0508	0001	0	12	Salto a SR Log [A = Log $\sqrt{1 + \text{sen}^2 w}$]
0509	0256	0	32	(A) \longrightarrow z
0510	Alto o enlace con resto del programa

OBSERVACIONES.-

-) Instrucción en 0501: Es necesario hacer (J6) = 0 para que la SR calcule sen x y no cos x, de acuerdo con las observaciones de la lista de subrutinas.
-) Instrucción 0507 y 0508: En ambos casos el resultado de la subrutina queda en el Acumulador.

PROGRAMACION RELATIVA

Es evidente que sería muy deseable que los programas y subrutinas fueran independientes de su ubicación en la memoria. Esto se puede conseguir mediante la programación relativa.

Instr. #		Programa I
1	0312	0316 0 12
2	0313
3	0314
4	0315
5	0316

En el ejemplo del programa I, el salto a la instrucción número 5 está dado en forma absoluta. El salto se efectuará a la instrucción # 5 si la instrucción # 5 está en la celda 0316. Es decir, el programa depende de su ubicación en la memoria.

Mediante el uso del J9(*) podemos evitar esta dependencia. Cuando se empieza a ejecutar la instrucción # 1, J9 ya tiene el número 0313, la dirección de la próxima instrucción.

Daremos, entonces, una instrucción de salto relativo:

0312	0003 9 12
------	-----------

De este modo, el salto se efectúa a la dirección ubicada 3 celdas más adelante que la especificada por J9, es decir, a cuatro celdas más adelante de la instrucción de salto.

Salto a : $0003 + (J9) = 0003 + 0313 = 0316$.

Claramente se ve que, sin importar dónde esté ubicado el programa, el salto siempre se va a efectuar a la instrucción número 5.

(*) J9 es el contador (ver apuntes).

El salto relativo también puede hacerse hacia atrás aprovechando la propiedad del complemento de un número.

Si (J9) = 0100 por ejemplo y damos la siguiente instrucción :

9990 9 13 ,

el salto condicional se efectuará a

$$\begin{array}{r} 9990 \\ + 0100 \\ \hline \boxed{1}0090 \end{array}$$

Como el registro sólo tiene lugar para cuatro cifras, el uno se pierde y el salto se efectúa a la instrucción de la celda 0090.

En general, si queremos retroceder en n celdas, en la parte p de la instrucción debemos poner 9999 - r. En efecto, veamos un ejemplo:

# celdas	p	i	OP
....
....
....
0305	9995	9	14
....

Si queremos saltar 4 celdas hacia atrás, a la celda 0301, p = 9999 - 4 = 9995.

Comprobemos:

$$\begin{array}{r} (J9) = 0306 \qquad 9995 \\ + \qquad \qquad \qquad 0306 \\ \hline \boxed{1}0301 \\ \checkmark \\ \text{se pierde} \end{array}$$

El salto se efectúa a 0301

Resumiendo: Para saltar r instrucciones, a partir de la instrucción de salto,

se deberá tener :

	p	i	OP
Hacia adelante:	r - 1	9	Salto

Hacia atrás	9999 - r	9	Salto
-------------	----------	---	-------

Ejemplos: -) Para el enlace automático de la subrutina, la instrucción de la celda # 0610 en el ejemplo de la página 14 deberá ser :

0049 9 89

-) En el programa de la página 13 la instrucción de la celda 513 será :

9996 9 16

19/10/61

Secc. Computadores y Servomecanismos, IIEE, U. de Chile

LISTA DE SUBROUTINAS

29

\sqrt{x}

x

e

Ln x

sen x

cos x

tg x

cotg x

arcsen x

arccos x

arctg x

arccotg x

Aritmética compleja

Polinomio en x

Cálculo de raíces de polinomio de grado y con coeficientes reales

$y' = f(x, y)$ por Runge-Kutta, Resultado: $y = \phi(x)$

$y'' = f(x, y, y')$ " " " " " "

$y^{(IV)} = f(x, y, y', y'', y''')$ " " " " " "

Matriz multiplicada por vector

Módulo de un vector

Transposición de Matriz

Suma de matrices

Multiplicación de matrices

Solución de Sistema de ecuaciones

Solución de Sistemas de ecuaciones complejas

Inversión de Matriz

Determinante de Matriz

Maximización

Programación lineal

NOMBRE	MODO	RANGOS	NECESARIAS	TIEMPO MÁXIMO [ms]	VARIABLES Y DATOS	RESULTADO	REGISTRO DE INDICE e INDICADORES NECESARIOS	OBSERVACIONES
Solución de un sistema de ecuaciones lineales $\ A\ \ x\ = \ y\ $	F1	Computador	Para subrutina $\frac{14}{3}$ Para $\ A, Y\ $: $2m(n+1) + 2$ Para Resultados: $\frac{2n}{2n}$	$2,5m^3 + 5,4m^2 + 19,8m + 10$ [ms]	$\ A, Y\ $ se guarda en celdas sucesivas. Los y_j^i se marcan con Q.	b_{01}, b_{02}, \dots $\dots b_{0m}$	J5, J6, J7, J8 M8	$> a_{11} < , > a_{11} < , > a_{11} < , m$ $> b_{01} < , m$ a continuación de salto en programa principal.
Solución de un sistema de ecuaciones lineales complejas para el caso de uno a p segundos miembros. $\ A\ \ x\ = \ Y\ $	F1	Computador	Para subrutina: $\frac{181}{181}$ Para (A, Y): $4m(m+p) + 2$ Para Resultados: $4mp + 2$	$11,6m^3 + 23,2m^2 p + 3,5m^2 + 3,5mp + 5,4m + 0,2p + 15$ [ms]	(A, Y) se guarda en forma sucesiva con la parte imaginaria en la celda que sigue a la parte real. El final de los registros se marca con Q en la parte imaginaria.	Se encuentran a partir de $> x_{11} <$ $\text{hex } x_{11}, \text{ imp } x_{11}$ Van en celdas sucesivas.	J5, J6, J7, J8 M1, M9	$> a_{11} < , m$ a continuación de instrucción de salto en programa principal.
Cálculo del determinante de una matriz	F1	Computador	Para subrutina: $\frac{126}{126}$ Para datos: $2m^2 + 2$	$2,3m^3 + 3,6m^2 + 21,2m + 100$ [ms]	Los a_{ij} se guardan sucesivamente. Los a_{im} llevan marcas Q.	(A) = det. A		

NOMBRE	TÍTULO	RANGOS	CELIDAS NECESARIAS	TIEMPO MAXIMO	VARIABLE Y DATOS	RESULTADO	REGISTRO DE INDICE E INICIAL-BORES NECESARIOS	OBSERVACIONES
<p> $c_{ik} = \sum_{j=1}^n a_{ij} b_{jk}$ $i = 1, \dots, m$ $j = 1, \dots, n$ $k = 1, \dots, p$ </p> <p>Inversión de matriz</p>	F1	<p>Computador</p>	<p>Para subrutina: 208 Para datos : $2m^2 + 2$ Para memoria: $3m + 2$</p>	<p> $4, 7 m^3 + m^2 + 14m + 13$ </p>	<p> a_{im} en celdas sucesivas a_{im}² (i=1, ..., m) llevan marca U Al final va palabra Q </p>	<p>J5, J6, J7 J8 J9, J9</p>	<p>> a₁₁ <, > b₁₁ < 11 11</p> <p>a continuación de instrucción de salto en programa principal. (b: memoria temporal)</p>	
<p> Multiplicación de matrices $c_{m,n} = \begin{matrix} a_{m,n} \\ b_{m,p} \\ c_{m,p} \end{matrix}$ </p>	F1	<p>m,n,p: dependen de capacidad de la memoria a,b,c, : rango al computador</p>	<p>Para subrutina: 59 celdas Para datos 2(mn + np + mp) + 6 celdas</p>	<p> 5,6 mnp + 3,35 mmp + 1,78 m + 4,67 </p>	<p> a,b,c, en celdas sucesivas a₁₁, a₁₂, etc... Las últimas columnas llevan marca Q. Marca Q al final. </p>	<p> c_{11}, c_{12}, \dots </p> <p>J4, J5, J6 J7, J8 J8, J9</p>	<p>> a₁₁ <, > b₁₁ < 11 , 2p 11</p> <p>van en el espacio del programa principal a continuación de la instrucción de salto.</p>	

NOMBRE	MODO	RANGOS	CELIDAS NECESARIAS	TIEMPO MAXIMO ms	VARIABLE Y DATOS	RESULTADO	REGISTRO DE INDICE E INDI- CADORES NECES- ARIOS	OBSERVACIONES
Aritmética compleja in- cluyendo va- lor recíproco	F1	Computador	94		(A) = Re z (M) = Im z	(A) = Re (M) = Im	J8, M1	M1 + (*)
Polinomios en x, Término por marca Q	F1	*	10	0,94 + g . 3,71	a ₁ , a ₂ , g-1, ..., a ₂₀ en celdas sucesi- vas (A) = x	(A) = a ₁ x ⁹ + ... + a ₀	J7	(J7) = y ag Q a ₀
Polinomios en x Término por contador	F1	*	12	1,07 + g . 3,43	a ₁ , ..., a ₂₀ en celdas sucesi- vas (A) = x	(A) = a ₁ x ⁹ + ... + a ₀	J6, J7	(J7) = y a (J6) = g
Raíces λ_j de polinomio de grado n. $\sum_{j=0}^n a_j x^j$ j = 0 con coeficientes reales	F1	x Computador n ≤ 70	420 + 2n(n + 1)	*	a ₁ , ..., a ₂₀ en celdas suce- sivas.	$\lambda_1, \dots, \lambda_n$ Re λ e Im λ se guardan en forma consecuti- va.	J6, J7, J8 M9	(J6) = y c _n (J7) = y λ (J8) = y SR necesita S Q a ₀

(*) Para las siguientes operaciones se saltará a la dirección indicada relativa a la primera instrucción de la subrutina

Traspasar a celda	+	0
Comparar	+	6
Suma	+	22
Resta	+	28

Multiplicación	+	38
División	+	53
División recíproca	+	59
Valor recíproco	+	61

$y' = f(x, y)$ por Runge- Kutta resultado: $y = \varphi(x)$	F1	Computador	100 + 4 (cada punto de $\varphi(x)$)	*	Un forma conse- cutiva: abscisa inicial x ₀ (continúa)	(x ₀ , y ₀), (x ₁ , y ₁), (x _{máx} , y _m)	J6, J7, J8	Programa pa- ra calcular f(x, y) (A) = f(x, y)
---	----	------------	---	---	--	---	------------	---

NOMBRE	MODO	RANGOS	CELIDAS NECESARIAS	TIEMPO MAXIMO INS	VARIABLE Y DATOS	RESULTADO	REGISTRO DE INDICES E INDICADORES NECESARIOS	OBSERVACION
\sqrt{x}	F1	$10^{-50} \leq x < 10^{49}$	72	21,83	(A) = x	(A) = \sqrt{x}	J8	
e^x	F1	$-100 \leq x < 100$	96	26,75	(A) = x	(A) = e^x	J7, J8	
Ln x	F1	Rango de x en el computador	85	33,84	(A) = x	(A) = Ln x	J7, J8	
senx cos x	F1	$10^{-49} < x < 10$	80	24,21	(A) = x (A) = x	(A) = sen x (A) = cos x	J6, J7, J8	Para sen x, J6 = 0 Para cos x, J6 = 2500
tg x cot x	F1	$-10 < x < 10$	94	39,08	(A) = x	(A) = tg x (A) = cot x	J6, J7, J8 M9	Para tg x, J6 = 0 Para cot x, J6 = 1
arc sen x	F1	$-1 \leq x \leq 1$	107	54,46	(A) = x	(A) = sen x	J6, J7, J8 M9	
arc sen x arc cos x	F1	$-1 \leq x \leq 1$	112	55,89	(A) = x	(A) = arc sen x (A) = arc cos x	J6, J7, J8 M8, M9	M8 + da arc sen x M8 - da arc cos x Necesita: Subrutina V
arc cot x	F1	Computador	99	38,24	(A) = x	(A) = arc cot x	J7, J8 M9	
arc tg x	F1	Computador	99	38,24	(A) = x	(A) = arc tg x	J7, J8 M9	
arc tg x arc cot x	F1	Computador	101	38,38	(A) = x	(A) = arc tg x (A) = arc cot x	J6, J7, J8 M8, M9	M8+ : arc M8- : arc

CODIGO ALFANUMERICO PARA EL E R - 56

Todas las letras y signos se representan por un par de cifras decimales. Para que el computador interprete la información como del tipo alfanumérico el prefijo debe ser igual a 3.

00 = 0	25 = %
01 = 1	26 = &
02 = 2	27 = #
03 = 3	28 = &
04 = 4	29 = ß
05 = 5	30 = -
06 = 6	31 = ?
07 = 7	32 = !
08 = 8	33 = <
09 = 9	34 = >
10 = =	35 = Δ
11 = +	36 = } Reserva
12 = :	37 = }
13 = '	38 = }
14 = (39 = Espacio
15 =)	40 = -
16 = "	41 = Nueva línea
17 = [⊕	42 = Retorno del carro
18 =] ⊖	43 = una
19 = ;	44 = permanente
20 = □	45 = desconectar
21 = /	46 = Cinta roja
22 = , (coma)	47 = Cinta negra
23 = °	48 =
24 = *	49 =

a elección para el teleimpresor

conectar para letras mayúsculas

50 = Palabra tabuladora 0
 51 = " " 1
 52 = " final Z
 53 =
 54 =
 55 =
 56 = } Reserva
 57 =
 58 =
 59 =
 60 = a
 61 = ä
 62 = b
 63 = c
 64 = d
 65 = e
 66 = f
 67 = g
 68 = h
 69 = i
 70 = j
 71 = k
 72 = l
 73 = m
 74 = n
 75 = o
 76 = ö
 77 = p
 78 = q
 79 = r
 80 = s

81 = ß
 82 = t
 83 = u
 84 = ü
 85 = v
 86 = w
 87 = x
 88 = y
 89 = z
 90 =
 91 =
 92 =
 93 = } Reserva
 94 =
 95 =
 96 =
 97 =
 98 =
 99 = signo ω