



11  
2 y.

# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES

CUAUTITLAN

SIMULACION DE UN PROCESADOR DE  
ARQUITECTURA MINIMA EMPLEANDO  
UN LENGUAJE DE ALTO NIVEL

T E S I S  
QUE PARA OBTENER EL TITULO DE:  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A N :  
JUAN CARLOS MENESES SANCHEZ  
ANGEL ORTEGA GONZALEZ  
LUIS MANUEL MUÑOZ DIAZ.

MEXICO, D.F.

1989.

**FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

UNAM



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

### III

## PROLOGO

La simulación por computadora es una herramienta, que se emplea en muchos campos como la ingeniería, medicina, aeronáutica, así como en otras ciencias y actividades sociales. Permite conocer el comportamiento de un sistema o modelo en estudio, y -predeterminar sus respuestas a determinados estímulos, los cuales serían más complejos de establecer en un proceso real, de esta manera se conocen sus respuestas teóricas a dichos estímulos, lo cual es de gran utilidad y proporciona un conocimiento más amplio del sistema.

En esta tesis se define como un procesador de arquitectura mínima a un sistema electrónico digital de proceso elemental programable, el cual está integrado por las unidades de entrada, memoria, procesamiento central, control y de salida.

La finalidad del presente trabajo es con el objeto de dar a conocer un programa de computadora que simule el comportamiento de un sistema electrónico digital de proceso programable real. Este programa se realizó en una microcomputadora HP 9816, empleando un lenguaje BASIC, de alto nivel, que permite alto grado de efectividad, se procuró al mismo tiempo utilizar instrucciones - que no fuesen específicas para este tipo de máquina, lo cual obligó a realizar necesariamente un programa más amplio, pero de esta manera se puede programar en cualquier máquina con la condición de que se emplee lenguaje BASIC estándar.

Este programa simulador permite programar en lenguaje ensamblador sin tener la necesidad de estar en contacto directo -- con el hardware de un sistema digital de proceso en especial, y además tiene la ventaja de que puede ser ejecutado en una computadora con sistema multiusuario. El programa se comunica con el usuario tanto en el proceso de introducción de datos como en la obtención de resultados.

El conjunto de instrucciones que se indica como mínimo es el que se requiere para programar el sistema de proceso elemental y, establece el poder del cálculo, la rapidez y efectividad adecuada a la arquitectura simulada. Estas intrucciones modifican el estado de los registros de las unidades de proceso central y control, así como las localidades de memoria, paso a paso con la finalidad de comprender su interacción y operación.

Se muestra una arquitectura mínima de la cual se deriva un diagrama de flujo que describe cada una de las funciones del programa. Se presenta un manual del usuario del simulador, donde se da a conocer el número de instrucciones nemotécnicas, su formato; los operandos, el rango del operando se concidera por diseño de un byte o sea de 0 a 255 decimal; también se muestra los cambios que sufren los registros del simulador en sistema -- decimal o hexadecimal a opción del usuario.

Se presenta la forma de introducción de la información, - ya que tiene ciertas diferencias a un sistema de proceso común,

tan sólo por facilidad de uso del simulador. Se describen las características funcionales de las cuales se integra el simulador, así como las funciones lógicas-aritméticas que se realizan y banderas de condición que se utilizan.

Se indican las especificaciones de operación del programa, desplegados que se presentan y su significado, introducción de códigos de operación y operandos, desplegado del estado de los registros. También se presentan, algunos ejemplos con una previa explicación.

Finalmente se presentan las conclusiones y se menciona -- cual será el trabajo futuro a desarrollar para que el simulador de un sistema electrónico digital de proceso elemental programable, sea más eficiente y lo más cercano a la realidad. Se lista el programa para aquella persona que quiera profundizarse en su estudio, y lo tome como base para un proyecto más complejo.

## I N D I C E

	Pág.
Prefacio	I
Agradecimientos	II
Prólogo	III
Indice	VI
CAPITULO I	INTRODUCCION
1.1	La simulación de sistemas por computadora y su importancia. 1
1.1.1	Modelo de sistemas 3
1.2	Porqué la simulación por computadora de un sistema electrónico digital de proceso programable. 9.
1.3	Aspectos generales de la simulación de sistemas por computadora. 10
1.4	Aplicación de la simulación por computadora al sistema electrónico digital de proceso programable. 12
CAPITULO 2	DESCRIPCION DE UN SISTEMA ELECTRONICO DIGITAL DE PROCESO PROGRAMABLE 15
2.1	Qué es un sistema electrónico digital de proceso programable. 15
2.1.1	Estructura general 16
2.2	Porqué un procesador mínimo 19
2.3	Arquitectura y operación del procesador mínimo 21

## vii

2.3.1	Ciclos de trabajo del procesador mínimo	25
2.3.2	Grupo de instrucciones básicas para un procesador mínimo	27
CAPITULO 3	ESTRUCTURA Y OPERACION DEL SIMULADOR	30
3.1	Estructura y diagrama de flujo	30
3.2	Especificaciones del simulador	41
3.3	Descripción de los ciclos de trabajo del simulador y similitud con la operación real del procesador.	45
CAPITULO 4	MANUAL DEL USUARIO	48
4.1	Tipos de código de operación	48
4.2	Formatos de las instrucciones nemotécnicas.	51
4.3	Instrucciones nemotécnicas	53
4.4	Mensajes de error	63
4.5	Instrucciones de operación del programa.	64
4.5.1	Ejemplo de operación del programa.	66
4.6	Pruebas y resultados	70
CAPITULO 5	CONCLUSIONES	123

VII

BIBLIOGRAFIA	126
APENDICE 1 Características del programa fuente	129
APENDICE 2 Listado del programa fuente	132
APENDICE 3 Características de la máquina empleada	168

## INTRODUCCION

1.1 La simulación de sistemas por computadora y su importancia.

A fines de los años 60's y principios de los 70's, se emprendieron estudios sobre una nueva herramienta, capaz de evitar los grandes costos y espacios de tiempo, empleados en la experimentación tanto de fenómenos físicos como socioeconómicos, tal fue la utilidad y ventajas que se vislumbran en la simulación de sistemas por computadora.

De acuerdo a Shannon (20), la simulación es el proceso de diseñar un modelo de un sistema real y conducir experimentos con este modelo, con la finalidad de comprender el funcionamiento -- del sistema o evaluar las posibles estrategias para la operación del mismo. Esta descripción pone énfasis en las dos caras de este apasionante campo de la ingeniería, que se está expandiendo rápidamente. La primera cara es el proceso creativo de construir un modelo simple, que semeje adecuadamente el comportamiento dinámico de un sistema; la segunda, la traducción del modelo en un programa de computadora y el análisis de su comportamiento para predecir las respuestas del sistema.

La simulación por computadora, es uno de los métodos de -

solución de problemas más poderoso y versátil de nuestro tiempo.

Se puede decir que en la actualidad, la simulación se clasifica en los dos siguientes tipos: discreta o continua. Cada una tiene su propio conjunto de procedimientos para la conceptualización de modelos, y se basan en un área diferente de las matemáticas y cada una resuelve un conjunto distinto de problemas.

La simulación discreta, generalmente o típicamente trata con sistemas de "colas" en los cuales "clientes", llegan a solicitar algún servicio en algún sitio y esperan a ser atendidos -- formados en una "cola", en el caso de que todos los servicios se encuentren ocupados, eventualmente recibe servicio y finalmente abandona el lugar cuando éste se la ha prestado por completo. El principal interés en estos sistemas de "colas", está en el -- proceso rápido de atención a clientes. Ya que el tiempo en que estos llegan, y el tiempo en el que se les da el servicio son generalmente eventos aleatorios, los modelos matemáticos usados para representar sistemas de "colas", casi siempre son clasificados dentro de la probabilidad y estadística.

Intervalos de tiempo largos en los cuales el sistema no cambia, son características de los sistemas discretos. Otra característica es que las unidades empleadas en las mediciones de estos sistemas son enteros. Los eventos ocurren en instantes -- discretos, y entre uno y otro nada de importancia ocurre para modificar el estado del sistema.

La simulación continua en contraste, trata con sistemas - que cambian continuamente en el tiempo y con unidades que no están restringidas a enteros. Un ejemplo sencillo es la trayectoria que sigue un proyectil en vuelo; ya que no existe un sólo -- instante de tiempo en el cual su posición en el espacio no cambie.

El principal interés en todos los sistemas continuos, es el comportamiento variable en el tiempo de todas las cantidades involucradas en el sistema. Ya que los patrones de comportamiento son gobernados generalmente por relaciones de cambio, los modelos matemáticos para sistemas continuos están basados en ecuaciones diferenciales.

No siempre es claro si es mejor usar simulación discreta o continua en la solución de un problema particular. Por ejemplo, el problema de crecimiento poblacional, podría lucir como - un candidato ideal para usar simulación discreta. Sin embargo, los problemas de población, frecuentemente son analizados con simulación continua, debido principalmente a que el tiempo entre - eventos sucesivos (nacimientos y muertes) es pequeño (segundos), comparado con el dominio de tiempo de interés (año).

#### 1.1.1 Modelado de sistemas.

De acuerdo con Jeréz.(23), los modelos pueden ser clasificados como físicos, que son los que pueden representar materialmente al sistema, aunque no sea totalmente idéntico a él; y abstrac

tos son los que no representan materialmente al sistema. Un modelo físico que luce idéntico al sistema que representa, es un maniquí de un ser humano. Un modelo físico que no luce idéntico al sistema que representa, aunque se comporta igual; es por ejemplo, cuando los ingenieros construyen sistemas electrónicos para modelar sistemas de resortes mecánicos y predecir el movimiento de la masa en una de las puntas del resorte a través de medir corrientes y voltajes en el modelo.

Los modelos abstractos son conjuntos de ecuaciones matemáticas que relacionan las variables del sistema que modelan. Las soluciones a estas ecuaciones, son usadas para predecir el comportamiento del sistema.

La simulación discreta emplea modelos abstractos, usando archivos de computadora para representar el comportamiento del sistema. Estos archivos son manipulados durante la simulación de acuerdo a los patrones establecidos por el modelador para imitar el comportamiento del sistema.

Las ecuaciones diferenciales usadas para definir modelos abstractos, son la base de la simulación continua. Para modelos de ingeniería y otras ciencias exactas, tales ecuaciones son deducidas a partir de principios físicos (tales como leyes de Newton) que gobiernan los cambios en los estados de los sistemas.

Una vez generado el modelo, es necesario codificarlo en un programa, para poder simular su comportamiento usando una com

putadora.

Se define a un sistema como un conjunto de elementos reunidos en alguna interacción o interdependencia regular. Frecuentemente a estas interacciones del sistema, motivan cambios dentro del mismo, sin embargo, ciertas actividades del sistema, también pueden producir cambios que no reaccionan en éste. Se dice que los cambios que ocurren fuera del sistema, ocurren en el medio ambiente del sistema.

Un paso importante en los modelos de los sistemas, es establecer el límite entre estos y su medio ambiente. Esta decisión siempre está relacionada con el propósito del estudio.

Se emplea el término "endógeno" para describir las actividades que ocurren dentro del sistema y "exógeno" para describir las actividades en el medio ambiente que afectan al sistema.

El sistema donde no hay actividad exógena se llama cerrado, mientras aquél en donde si existe actividad exógena, se denomina como sistema abierto.

Una observación que también concierne al sistema, es que en donde es posible describir completamente el resultado de una actividad en términos de su entrada, se dice que la actividad es determinística.

Si la ocurrencia de la actividad está bajo control del sistema, se le considera como "endógena", de otra forma, si la ocurrencia es aleatoria, constituye parte del medio ambiente que rodea al sistema.

Es evidente, que no es posible realizar un experimento de un sistema, cuando aquél se halla en su forma hipotética, en tal caso se puede tomar la alternativa de construir prototipos y probarlos, lo cual generalmente resulta ser costoso y dilatado; debido a estos inconvenientes, los estudios de sistemas pueden realizarse, ejecutando un programa en computadora que simule el comportamiento del modelo planteado.

Cuando se inicia el estudio de un sistema, no siempre se hace en forma general, es decir, no necesariamente se le ve como un todo, sino que dependiendo del fin buscado, se evoca a ciertos puntos claves del sistema. Por ejemplo, si se desea estudiar la distribución de campo magnético en el núcleo de un transformador, un punto clave en este estudio, sería el material del cual está hecho el núcleo bajo estudio, mientras que no sería relevante, saber la naturaleza de la fuente que va a provocar la excitación.

Un modelo se puede plantear tan complicado o tan sencillo como se desee, dependiendo su finalidad. Por ejemplo, si se quiere simular un microprocesador interactuado eléctricamente con algún transductor, el modelo que permitiera la programación

y ejecución de programas para el microprocesador sería insuficiente, como también lo sería un modelo que permitiera la simulación de la interacción eléctrica de circuitos para aplicaciones de verificación de lógica de programas.

Si se quiere simular una presa, se hacen simulacros de ella para descubrir los fenómenos asociados a ésta y después poder construir una presa confiable.

Se efectúan modelos a escala de sistemas mecánicos para probar sus capacidades y limitaciones; o se simula el comportamiento de un dispositivo electrónico mediante el uso de modelos que se asemejan al dispositivo real en cuanto a su comportamiento se refiere. En la ingeniería electrónica, este hecho ha sido de gran utilidad, pues se ha podido diseñar en base a simulación, transistores, capacitores, resistencias y circuitos integrados. Además se pueden simular circuitos completos que de esta manera pueden ser estudiados más fácilmente, sin la necesidad de complicados prototipos hechos a mano, que tomarían una cantidad enorme de tiempo, esfuerzo y recursos económicos.

Dentro de la educación, que es una rama en la que se puede utilizar la simulación por computadora como reforzador del proceso de enseñanza-aprendizaje, se tiene la siguiente aplicación:

Siscom (simulation and computer).- Que es un sistema de computación educativo, cuyo fin es enseñar a utilizar la simula-

ción. Este sistema se encuentra en la U.N.A.M. en el centro de servicios de cómputo.

- 1.2 Porqué la simulación por computadora de un sistema electrónico digital de proceso programable.

Comprender la arquitectura de un sistema electrónico digital de proceso programable, y como éste manipula los números, es de gran ayuda en el aprendizaje de circuitos lógicos y de lenguajes de alto nivel. El conocimiento de la operación de las instrucciones de máquina, permite al programador hacer mejores programas, además que propicia el uso pleno de las características del sistema de proceso.

Sin embargo, para comprender programas escritos en lenguaje de máquina, es necesario primero entender la arquitectura propia del sistema de proceso, que se va a emplear, así como tipos de direccionamiento. Una herramienta útil en la comprensión de la arquitectura de sistemas electrónicos de proceso programables, es el objetivo del simulador.

El simulador sirve como material didáctico de apoyo para impartir la cátedra de introducción a circuitos lógicos y microprocesadores.

Empleando el programa simulador, en un sistema multiusuario, varios alumnos podrían programar el sistema electrónico digital de proceso a la vez, sin interactuar con el hardware de la máquina.

### 1.3 Aspectos generales de la simulación de sistemas por computadora.

La aplicación de la simulación a diversos tipos de sistemas produce gran variedad en la forma de como se desarrolla un estudio de simulación, pero se puede identificar determinados pasos básicos en este proceso y son los siguientes:

- 1.- Definición del problema.
- 2.- Formulación de un modelo.
- 3.- Construcción de un programa de computadora para el modelo.
- 4.- Validación del modelo.
- 5.- Ejecución del programa de simulación y análisis de resultados.

**Definición del problema.-** No debe desarrollarse ningún estudio de simulación, hasta que se enuncie claramente el problema y los objetivos; posteriormente, se pueden hacer la estimación del trabajo por realizar y el tiempo requerido.

**Formulación de un modelo.-** En este punto es necesario establecer la estructura del modelo, analizando los aspectos del comportamiento del sistema, que son relevantes en el problema que se trata, y es importante reunir los datos para obtener parámetros correctos y adecuados para su estudio.

**Construcción de un programa de computadora para el modelo.-**

Esta es una tarea relativamente bien definida, lo cual no implica que sea fácil; el modelo establece las especificaciones de lo que se debe programar, la dificultad de programarlo radica en la forma de cómo se construye, la finalidad del mismo y la aproximación a la realidad que se desee.

Validación del modelo.- Aquí se sabe si el simulador se comporta tal como se diseñó. Desde luego pueden ocurrir errores al programar el modelo. Idealmente los errores del modelo y los de programación, se separan validando el modelo antes de iniciar la programación.

Ejecución del programa de simulación y análisis de resultados.- Un factor que debe tomarse en cuenta, es el costo por ejecución del programa simulador en la computadora, ya que ello puede limitar el número de corridas que pueden hacerse. Aún cuando no existan estas limitaciones, se debe verificar cuidadosamente el número de ejecuciones que se necesiten. Si se tiene un estudio bien planeado, se habrá planteado un conjunto bien definido de preguntas y el análisis y los resultados tratarán de responderlas.

#### 1.4 Aplicación de la simulación por computadora al sistema electrónico digital de proceso programable.

Los pasos básicos de la simulación de sistemas detallados en el apartado 1.3, son aplicados a la simulación del sistema -- electrónico digital de proceso programable como se menciona a -- continuación:

1.- Definición del problema.- El objetivo del presente trabajo es crear un programa de computadora, que simule a un sistema electrónico digital de proceso programable, llamado en esta tesis procesador mínimo, y que puede ser ejecutado en una microcomputadora HP - 9816.

Tendrá como entrada una serie de instrucciones que -- modifiquen el estado de los registros del procesador. Su salida el estado del procesador después de ejecutar cada instrucción y el estado final al término de ejecutadas todas las instrucciones.

2.- Formulación de un modelo.- Dada la finalidad del -- trabajo, el punto clave del modelo comprende los procesos de ejecución de instrucciones, por ello es que se presenta la información sobre las partes fundamentales de un sistema electrónico digital de proceso programable. A continuación se describe esta información básica:

Partes de un sistema electrónico digital de proceso programable:

- Unidad de entrada.
- Unidad de salida.
- Unidad de control.
- Unidad de procesamiento central.
- Unidad de memoria.

Instrucciones:

- Grupo de carga.
- Grupo aritmético-lógico.
- Manipulación de bits.
- Grupo de salto y retorno.
- Grupo de entrada y salida.
- Grupo de control.

Una vez obtenidos, se procede a conocer su forma de actuar como parte del sistema y al mismo tiempo se genera y trasa un -- diagrama de flujo de estas funciones, para luego crear un programa que las realice.

- 3.- Construcción de un programa de computadora.- En este punto sólo se menciona el hecho de que existe este programa, como consecuencia de la información del modelo, ya que el programa se explica en detalle en el capítulo 3 del presente trabajo.

- 4.- Validación del modelo.- El modelo se valida al comprobar que éste se comporta como se esperó, ya que esto se aprecia en función de los resultados obtenidos en el capítulo 4 del presente trabajo, que trata sobre las corridas del programa que ilustran la interacción de los registros de la unidad central de proceso.
  
- 5.- Análisis de resultados.- Aquí se trata propiamente de analizar los resultados, para saber el grado de eficiencia del programa simulador.

## C A P I T U L O 2

DESCRIPCION DE UN SISTEMA ELECTRONICO  
DIGITAL DE PROCESO PROGRAMABLE

## 2.1 Qué es un sistema electrónico digital de proceso programable

Todo sistema electrónico digital de proceso programable es es tá compuesto de una parte física llamada Hardware y de otra lógica o secuencia de instrucciones conocidas como software.

El Hardware son los dispositivos electrónicos que forman la estructura del sistema, la cual se encarga de captar la información, de realizar las operaciones aritmético-lógicas y de la im presión de los resultados. Estos dispositivos son las unidades de procesamiento central (CPU), memoria, control, entrada y salida.

La unidad central de proceso y la unidad de control son el cerebro del sistema. Hay muchas unidades centrales de procesamiento en el mercado, aunque la nomenclatura de las instrucciones es algo diferente para cada una, el proceso básico de cómputo es semejante en todas, aunque no se puede aseverar que son idénticas, son numerosas las similitudes.

La unidad central de proceso (CPU) maneja la información, a través de una secuencia automática de operaciones elementales

de proceso.

El software, está formado por los programas escritos en el lenguaje apropiado, los que se almacenan en dispositivos de memoria con los cuales será utilizable el sistema.

### 2.1.1 Estructura general.

La estructura general de un sistema electrónico digital de proceso programable debe de tener la capacidad de captar la información, almacenarla, procesarla y darle salida a esta información ya procesada; por lo tanto, el sistema digital de proceso programable debe estar integrado por los siguientes dispositivos:

- a) Dispositivos de entrada: son aquellos que permiten la entrada de datos e instrucciones.
- b) Dispositivos de memoria: son los que almacenan el programa, datos de entrada y los datos resultantes de la ejecución del cómputo.
- c) Dispositivos de procesamiento: es la unidad aritmético-lógica, la cual se encarga de realizar las operaciones con los datos almacenados en los registros y/o datos de entrada.
- d) Dispositivos de control: son los que toman decisiones con respecto al flujo del programa y el control de los recursos demandados por el proceso, basándose en

los estados internos de los registros y de los cálculos efectuados.

- e) Dispositivos de salida: son para el almacenamiento -- provisional y despliegue de resultados.

Los elementos que fueron definidos anteriormente y su interconexión, se muestran en el diagrama a bloques de la figura -  
No. 1



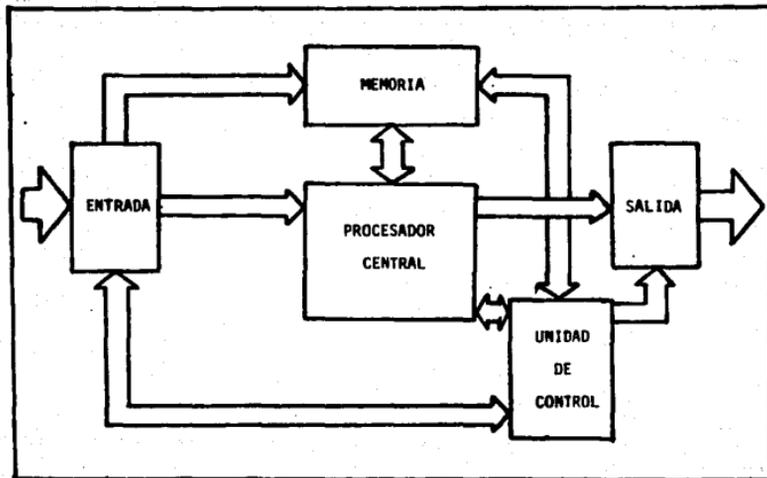


FIGURA No. 1. Diagrama a bloques de un sistema electrónico digital de proceso programable

## 2.2 Porqué un procesador mínimo.

Se simula un procesador de arquitectura mínima, con la finalidad de demostrar su funcionamiento básico. Debido a que su diseño está realizado con el mínimo de elementos, como se muestra en la figura No. 2; es más sencillo con esta arquitectura explicar la interacción de los registros y unidad aritmético-lógica de la unidad de procesamiento central, con la unidad de control, y con los dispositivos de apoyo como la unidad de memoria y las unidades de entrada y salida.

Adecuándose a la arquitectura de la figura No. 2, se define, un conjunto de 38 instrucciones, con las cuales se pueden demostrar el funcionamiento básico.

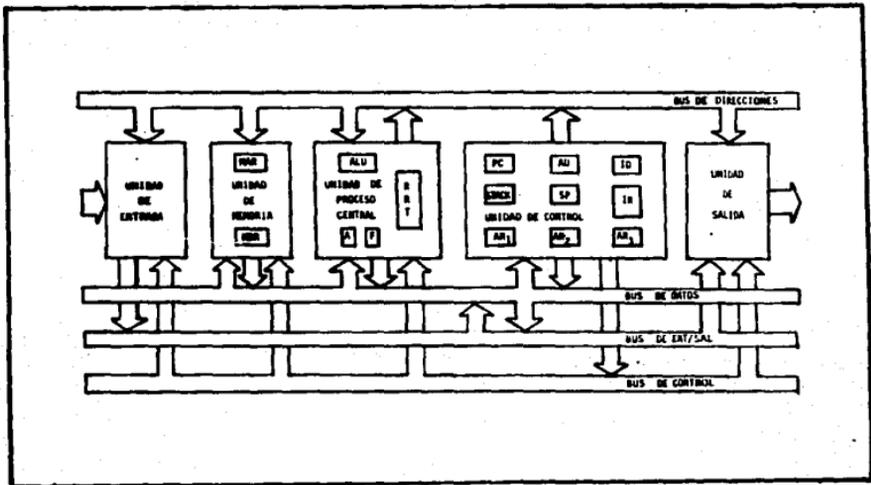


FIGURA No 2. Arquitectura de un sistema electrónico digital de proceso programable.

### 2.3 Arquitectura y operación del procesador mínimo.

Cada sistema posee una organización que debe ser fuertemente influenciada por la tarea que se realiza en él, por lo que se toma como base la estructura de la figura No. 2; ya que ésta se considera como mínima para explicar los objetivos mencionados. Y en dicha figura se distinguen los siguientes elementos funcionales:

- Unidad de entrada
- Unidad de memoria
- Unidad de proceso central
- Unidad de control
- Unidad de salida

y los siguientes buses:

- Bus de direcciones
- Bus de datos
- Bus de entradas-salidas
- Bus de control

Los buses son los encargados de intercomunicar a la unidad de proceso central, con las unidades de memoria, control y de entrada-salida.

En las cinco unidades de la figura No. 2, se presentan los siguientes registros básicos:

En la memoria, el registro de direcciones (MAR), y el registro de datos de memoria (MDR).

En la unidad de proceso, se encuentra el acumulador (A), el registro de banderas de estado (F), el registro rápido de trabajo (RRT) y la unidad aritmético-lógica (ALU).

En la unidad de control se presenta el contador de programa (PC), el stack con su apuntador correspondiente (SP), los registros auxiliares (AR), el registro de instrucciones (IR), el decodificador (ID) y la parte aritmética de direcciones (AD), teniendo en cuenta que la parte básica de generación de tiempos y de secuencia de las microinstrucciones no se presenta, aunque existe, y origina todos los movimientos generados por la unidad de control.

La unidad de entrada en ocasiones es un teclado con pantalla, que es controlado y direccionado por la unidad de control a través del bus de direcciones y de control.

La unidad de memoria es una serie de registros que son el almacén tanto de instrucciones como de datos.

La unidad de proceso central, realiza las operaciones aritmético-lógicas tales como SUMA, CORRIMIENTOS y lógicas, tales como AND, NOT, OR, etc; con la información que entra a ella a partir del bus de datos y direcciones. Tiene para ello una unidad aritmético-lógica (ALU), un acumulador, un registro de --

banderas de estado (arrastre, signos), así como unos registros rápidos de trabajo (RRT), en los que se almacena temporalmente los datos que no se interesa guardar indefinidamente en memoria.

La unidad de control es el cerebro del sistema. En ella se encuentra el contador de programa (PC), que direcciona localidades de memoria; el sistema aritmético de direcciones (AD), que incrementa o decrementa direcciones de la memoria; y su stack -- con el apuntador correspondiente (SP), que almacena las direcciones de las localidades de memoria.

También se encuentra en la unidad de control, el registro de instrucciones (IR) en el que se almacenan las instrucciones - obtenidas de la memoria y/o entrada a través del bus de datos; - el decodificador de instrucciones (ID), interpreta la información de la parte llamada código de operación de la instrucción y genera la secuencia de microinstrucciones que hace abrir o cerrar -- las puertas de los buses exteriores e internos de cada unidad -- funcional, así como grabar información de los buses en los registros apropiados. Por último se encuentran los registros auxiliares (AR1), (AR2), (AR3), encargados de almacenar los operandos de formato extenso.

La unidad de salida, recibe por el bus de datos la información a publicar al exterior y por el bus de direcciones, la dirección por la cual debe salir la información mencionada. Consiste en una serie de registros, en los que se deposita la in-

formación y son accionados por el bus de control. Esta unidad puede ser una terminal de video, impresora y/o transductor.

La información del bus de direcciones, la proporciona -- bien el contador de programa (PC) en una búsqueda normal de instrucciones, bien un registro auxiliar de la unidad de control en caso de direccionamiento directo, o bien uno de los registros rápidos de trabajo de la unidad de procesamiento central (CPU). La información del bus de direcciones, la utiliza la memoria a través del registro (MAR), para direccionar una localidad en la que se encuentra una instrucción o un dato, la unidad aritmético lógica para direccionar su acumulador o sus registros rápidos de trabajo. También la utilizan las unidades de entrada y salida para direccionar el canal correspondiente.

La información del bus de datos, la proporciona el registro (MDR), cuando se requiere obtener información de la memoria o también la proporciona la unidad aritmético-lógica a partir -- del acumulador, del registro de banderas de estado, de los registros auxiliares de la unidad de control y de los registros rápidos de trabajo.

La información del bus de datos la utiliza el acumulador, los registros rápidos de trabajo, los registros auxiliares, también el registro de instrucciones en cada operación o ciclo de búsqueda y el registro (MDR), en la operación de escritura y lectura en memoria.

El bus de entradas-salidas, es una extensión del bus de - datos, que recibe los datos de la unidad de entrada y los entrega a la unidad demandante u obtiene los datos de la unidad correspondiente y los entrega a la unidad de salida.

### 2.3.1 Ciclos de trabajo del procesador mínimo.

En principio, la ejecución de cualquier instrucción es de características parecidas. Todas las instrucciones tienen dos - ciclos fundamentales, llamados búsqueda y ejecución. El ciclo - de ejecución empieza por un subtiempo de decodificación de la -- instrucción y prosigue con la parte de ejecución propiamente dicha, que puede tener varios subtiempos dependiendo del tipo de - instrucción.

**Búsqueda.**- En todos los procesadores, el ciclo de búsqueda tiene las mismas características. Es el comienzo de una nueva instrucción, por lo tanto, el procesador no sabe a priori qué le va a pedir que ejecute aquella instrucción y por lo tanto los ciclos de búsqueda son exactamente iguales. El desarrollo de - un ciclo de búsqueda es el siguiente:

- 1.- El contenido del contador de programa (PC), a través del bus interno de la unidad de control, aparece en el bus de direcciones y temporalmente se grava en el registro de direcciones de memoria (MAR).
- 2.- El contenido de la posición de memoria direccionada por (MAR), aparece en el registro de datos de memo--

ria (MDR) y en el bus de datos. Esta información se deposita en el registro de instrucciones (IR). Al mismo tiempo, el contador de programa (PC) se incrementa en una unidad, ya que éste debe señalar siempre, la dirección de la siguiente instrucción a realizar.

Ejecución.- El desarrollo de un ciclo de ejecución es el siguiente:

- 1.- La primera parte del ciclo de ejecución, es también exactamente igual para todas las instrucciones, pues to que aún no se sabe qué operación hay que realizar.
- 2.- De momento, la parte de la instrucción llamada código de operación (opcode), se transfiere al decodificador de instrucciones. Entonces la unidad de control manda las instrucciones necesarias al CPU y a los dispositivos de apoyo a través del bus de direcciones y de control para realizar las instrucciones de acuerdo a sus características.

Dependiendo de las características del procesador, los diferentes tipos de instrucción serán de 1, 2 o más bytes. Cada byte está compuesto de 8 bits (unidad básica de información). En función del número de bytes de la instrucción, la ejecución tiene características diferentes.

Los ciclos de búsqueda y ejecución se realizan, casi siempre en 3 ciclos de máquina (M), y un ciclo de máquina consta de 3 y 4 ciclos de reloj.

En los ciclos de máquina M1 y M2 se realiza el ciclo de búsqueda y durante el ciclo M3 se realiza el ciclo de ejecución.

### 2.3.2 Grupo de instrucciones básicas para el procesador mínimo.

Las instrucciones de un procesador varían de acuerdo a sus características de diseño establecido. El conjunto de instrucciones que debe de ejecutar el procesador, se depositan en la memoria en forma de código de máquina (unos y ceros). Los códigos de operación son diferentes dependiendo de la instrucción a realizar. El programador puede establecer la secuencia de instrucciones necesarias para realizar las tareas requeridas, y variarlas cuando sea conveniente.

Estas instrucciones se van realizando secuencialmente, ya que la unidad de control va ordenando a la memoria que las introduzca en el registro de instrucciones, para que éstas se vayan al decodificador de instrucciones de la unidad de control y posteriormente se ejecuten en las otras unidades funcionales.

Las instrucciones pueden agruparse de acuerdo con diferentes criterios, funcionalidad, número de ciclos de máquina utili-

zados, número de bytes que utilizan. La clasificación más general es por su funcionalidad y pueden obtenerse los siguientes -- grupos:

- a) Instrucciones relativas al acumulador, a las banderas, a los registros de trabajo rápido, incremento y decremento, desplazamiento a derecha o izquierda, complementación, puesta a cero.
- b) Instrucción de intercambio de información entre registros y acumulador.
- c) Instrucciones de intercambio de información entre memoria y registros o acumulador.
- d) Instrucciones inmediatas de carga en registros, acumulador y memoria.
- e) Instrucciones aritmético-lógicas entre el acumulador o registros y memoria; sumas, restas, interacciones y comparaciones.
- f) Instrucciones de cambio de secuencia, saltos condicionales, incondicionales y saltos a subrutinas.
- g) Instrucciones de entrada-salida. Lectura y escritura de datos.

- h) Instrucciones especiales y de control; alto y no operación.

## CAPITULO 3

## ESTRUCTURA Y OPERACION DEL SIMULADOR

## 3.1 Estructura y diagrama de flujo.

Las partes funcionales de las que está integrado el simulador llamado procesador de arquitectura mínima se presentan en la figura No. 3, y se describen a continuación:

a) Unidad de entrada.- Las instrucciones y datos entran al procesador por esta unidad, la cual en el simulador es el teclado de la computadora HP 9816.

b) Unidad de salida.- La información procesada llega a esta unidad por medio del bus de datos, la secuencia de salida la proporciona la unidad de control a través de los buses.

En el simulador la salida es la terminal de video o la impresora.

c) Unidad de procesamiento central.- En esta unidad se realizan las operaciones aritmético-lógicas, con la información que llega desde la unidad de control, memoria o la unidad de entrada. Está integrada por un acumulador (ACUM), un registro de banderas de estado (s, z, c), un registro de uso general (B); y una unidad aritmético-lógica (ALU).

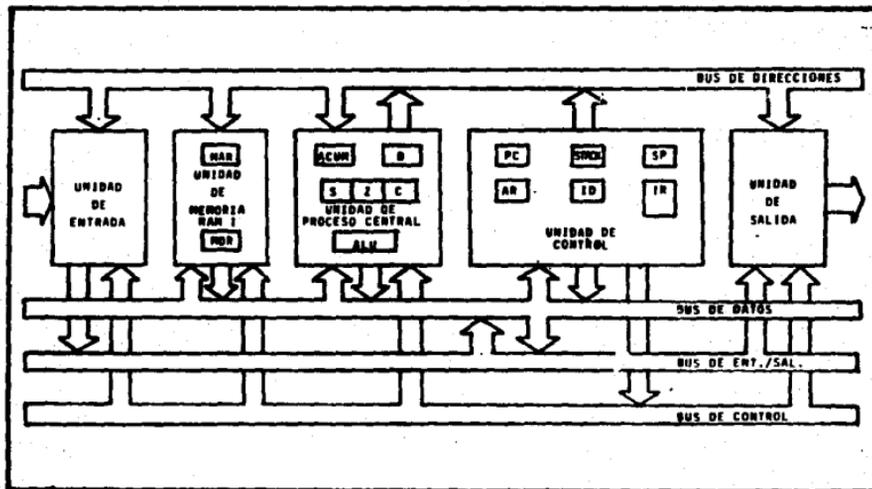


FIGURA No. 3. Arquitectura del simulador

d) Unidad de control.- Controla la secuencia de funcionamiento del procesador central, con los dispositivos de apoyo, los cuales son la memoria de escritura-lectura para el almacenamiento de instrucciones y datos (RAM 1), y los dispositivos de entrada y salida. Contiene el contador de programa (PC), el registro de stack con su apuntador correspondiente (SP), el registro de instrucciones (IR), el decodificador de instrucciones (ID) que interpreta el código de operación de la instrucción, un registro auxiliar de datos (AR). Todas las instrucciones, pasan primero de la memoria RAM I a la unidad de control y posteriormente a la unidad de procesamiento central.

e) Memoria de lectura y escritura (RAM I).- Almacena -- las instrucciones que van a ser ejecutadas por el procesador central (CPU) y cuyos resultados son dirigidos a la misma unidad de memoria (RAM I), o a la unidad de salida.

Está integrada por un registro de datos (MDR), un registro de direcciones (MAR), y 256 localidades de memoria de las cuales 246 localidades, con la dirección 00H a la F5H, son para almacenar instrucciones de un programa ensamblador; y las 10 localidades restantes con la dirección F6H a la FFH, son para el almacenamiento provisional de datos durante la ejecución de un programa.

El diagrama de flujo del algoritmo del procesador que se simula se muestra en la figura No. 4

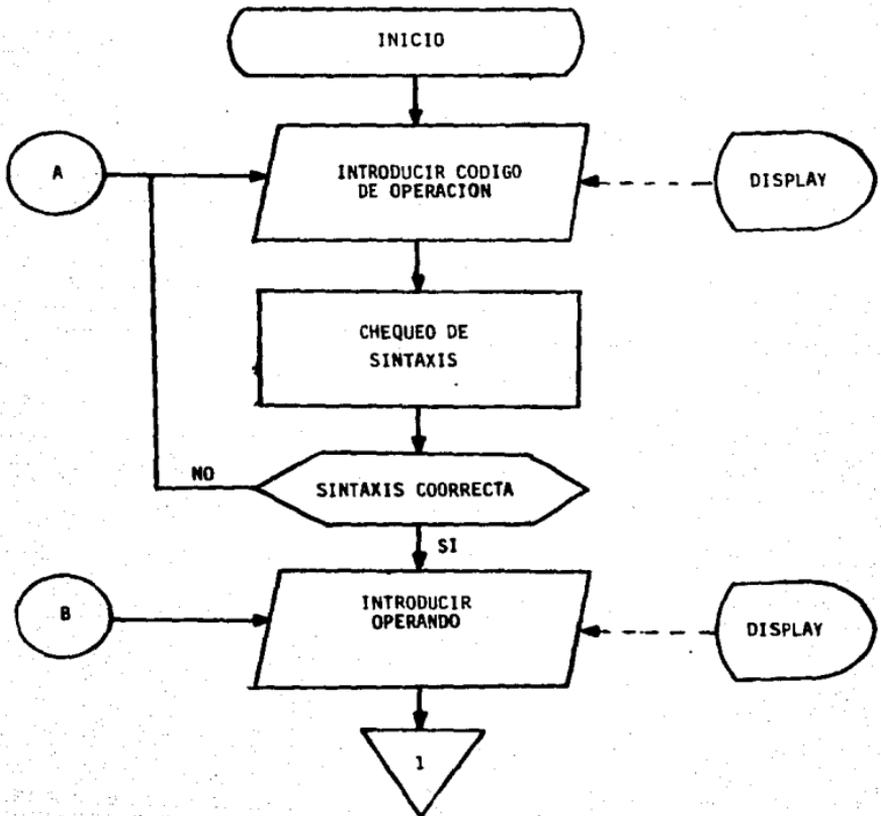


FIGURA No. 4. Diagrama de flujo del algoritmo del simulador.

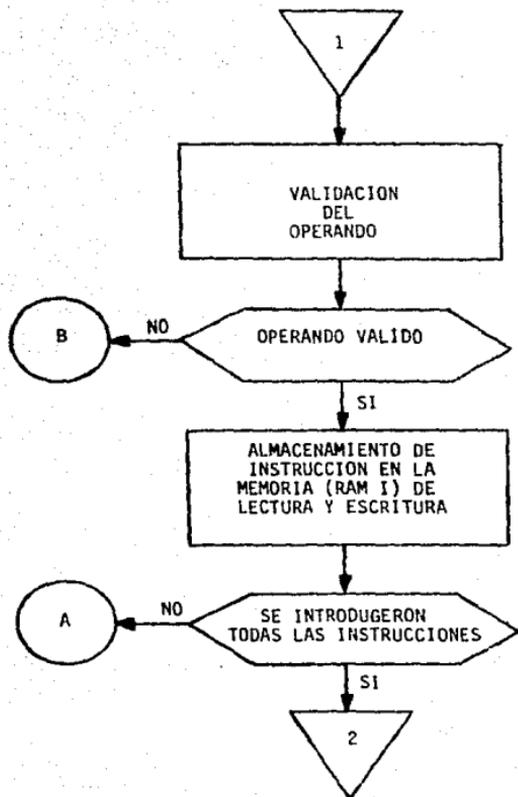


FIGURA No. 4 Diagrama de flujo del algoritmo del Simulador (Continuación)

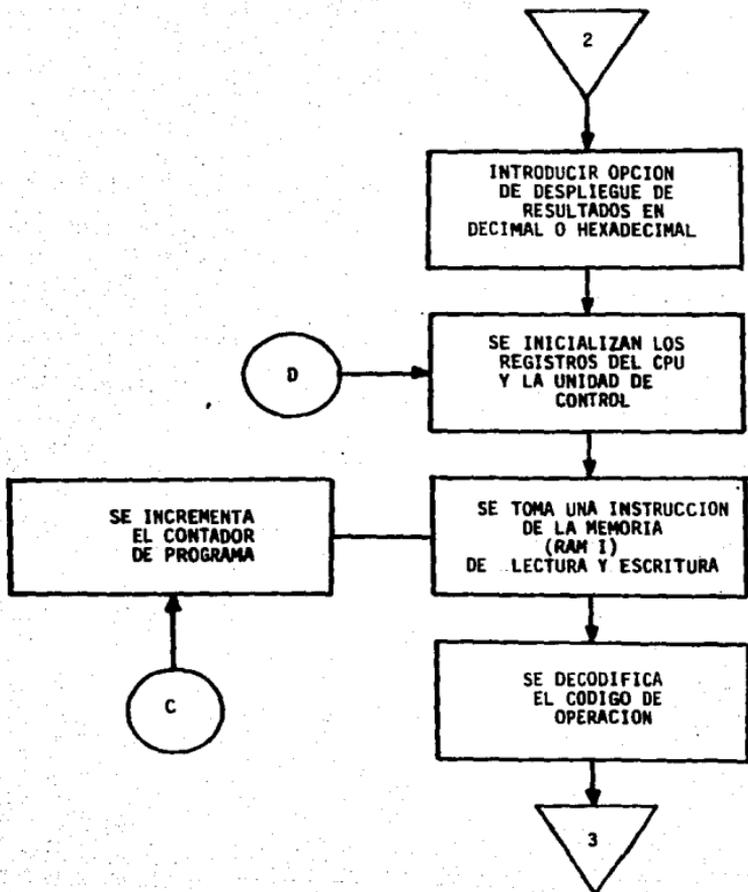


FIGURA No. 4 Diagrama de flujo del algoritmo del simulador. (Continuación)

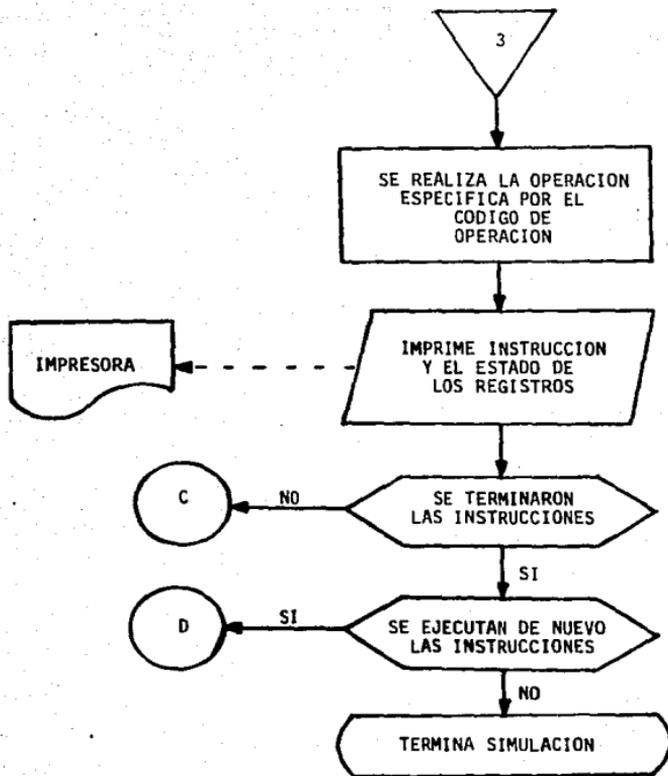


FIGURA No. 4 Diagrama de flujo del algoritmo del simulador (Continuación)

El programa simulador procesa la información de la siguiente forma:

- 1.- Una vez que el programa inicia su ejecución, el primer proceso que se realiza, es la impresión en la pantalla de la computadora las leyendas de "SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA", "PROGRAMARLO EN LENGUAJE ENSAMBLADOR", - "MAXIMO 246 INSTRUCCIONES POR PROGRAMA".
- 2.- Se introduce el código de operación de la instrucción.
- 3.- Realiza el chequeo de sintaxis del código de operación, si el código de operación es correcto, continúa la simulación - en caso contrario marca error, e informa que se introduzca - nuevamente el código de operación con sintaxis correcta.
- 4.- Se introduce el operando de la instrucción.
- 5.- Verifica que el operando sea un número hexadecimal dentro -- del rango 00H a FFH, todo operando debe estar integrado por dos dígitos y seguidos por la letra "H". Si el operando es -- correcto continúa la simulación en caso contrario, marca -- error e informa que se introduzca nuevamente el operando.
- 6.- Se finaliza el programa con la instrucción END 00H.
- 7.- Se almacenan las instrucciones en la memoria de lectura-es- -- critura (RAM I).

- 8.- Se pide la opción para que el usuario elija el despliegue de resultados en sistema decimal o hexadecimal.
- 9.- Se inicializan los registros de la unidad de proceso central y de control.
- 10.- Se toma una instrucción de una localidad de la memoria de lectura-escritura (RAM I).
- 11.- Se decodifica el código de operación de la instrucción en la unidad de control.
- 12.- La unidad de control manda las órdenes necesarias al procesador central y a los demás dispositivos de apoyo para ejecutar las instrucciones decodificadas.
- 13.- Se imprime la instrucción, el estado de los registros del procesador central y las 10 últimas localidades de memoria, después de ser afectadas por dicha instrucción en sistema decimal o hexadecimal, según la opción que haya elegido el usuario.
- 14.- Se incrementa el contador de programa y se realizan los pasos del 10 al 14; hasta ejecutar la última instrucción que se encuentra en la memoria de lectura-escritura (RAM I).
- 15.- La instrucción END OOH manda imprimir en sistema decimal o

hexadecimal el estado final de la unidad de proceso central, el contador de programa y las 10 últimas localidades de memoria en la unidad de salida (terminal de video o impresora).

- 16.- El simulador da la opción de ejecutar de nueva cuenta las -- instrucciones del programa ensamblador.

### 3.2 Especificaciones del simulador.

Los dispositivos funcionales del simulador tienen las siguientes características:

Unidad de entrada.- Comunica al simulador con el usuario, y por medio de esta unidad se introduce la información a las unidades de procesamiento central, de control, y de memoria a través del bus de direcciones y de datos. La información que se introduce a esta unidad son instrucciones y datos. La unidad de entrada del simulador es el teclado de la computadora HP 9816.

Unidad de memoria de lectura-escritura (RAM I).- Esta unidad sirve para almacenar el programa a ejecutar. Recibe la información de la unidad de entrada a través del bus de direcciones y de datos, está integrada por 256 localidades de memoria, de la dirección 00H a la F5H, son únicamente para el almacenamiento de instrucciones, o sea se dispone de 246 localidades de memoria para almacenar un programa ensamblador; y las 10 localidades restantes de la dirección F6H a la FFH, son para el almacenamiento provisional de datos durante la ejecución de un programa.

Se simula que cada localidad de memoria está integrada por 2 byte. En las primeras 246 localidades de memoria, el 1er. byte contiene el código de operación de la instrucción y el 2o. byte contiene al operando. En las 10 localidades restantes el 1er. byte contiene el dato a almacenar provisionalmente, y el 2o. byte se considera nulo. Esta unidad está simulada por medio de dos --

arreglos unidimensionales.

Cada Instrucción se almacena en una localidad de memoria. En instrucciones implícitas, el operando toma el valor de OOH -- (nulo), por lo cual el valor del 2o. byte es cero. El valor del 2o. byte se considera unicamente en instrucciones de direccionamiento directo e instrucciones que impliquen un registro de la unidad central de proceso.

Cuenta con dos registros extras para extraer la información, los cuales son el registro de direcciones de memoria (MAR) y el registro de datos de memoria (MDR).

Unidad de procesamiento central (CPU).- Esta unidad realiza las operaciones de SUMA, COMPARACION, RESTA, CORRIMIENTO DE IZQUIERDA Y DERECHA. Operaciones lógicas como la AND, OR, NEG, XOR. Operaciones de salto y de control. Recibe la información de las unidades de control, de entrada y de memoria, a través de los buses de dirección y de datos.

Está integrada por un acumulador (ACUM) de 8 bits y una unidad aritmético-lógica de 8 bits por dato, un registro de uso general (B) de 8 bits, que se creó para aumentar la eficiencia del simulador. Un registro de banderas de estado de 3 bits (s, z, c); la bandera S toma el valor de uno cuando el acumulador es negativo, la bandera Z toma el valor de uno cuando el acumulador es igual a cero, la bandera C se afecta con el valor de uno cuando hay sobreflujo en el acumulador o se afecta con el valor de -

un corrimiento aritmético. La bandera C elimina su valor de sobreflujo, cuando al acumulador se carga con un nuevo valor.

Unidad de control.- Realiza las funciones de control y supervisión del sistema de proceso. La información la recibe de las unidades de proceso central, memoria y de entrada; a través de los buses de datos y direcciones. El bus de control interviene en la selección de los dispositivos a emplear en la ejecución de una instrucción. Está integrada por los registros de stack - de 10 localidades de memoria con el apuntador (SP), que indica la dirección de la entrada hecha más recientemente ya que las posiciones de memoria están organizadas como un fichero de últimas entradas primeras salidas; un contador de programa (PC) que direcciona 256 localidades de memoria; un registro de instrucciones (IR) donde se almacena la instrucción que es producto de la lectura de la unidad de memoria; un decodificador de instrucciones (ID) es donde se decodifica el código de operación; el registro auxiliar (AR) es donde se almacena el operando, cuando se está decodificando el código de operación. A excepción del contador de programa (PC), los demás registros no se presentan en pantalla ya que son secciones del programa simulador y el usuario no tiene acceso directo a ellos.

Unidad de salida.- Proporciona el despliegue de instrucciones y el estado de los registros. Todos los datos procesados y enviados a esta unidad por las unidades de control, memoria, entrada, y procesamiento central, llegan por el bus datos, estos datos son el estado de los registros del procesador central y --

del contador de programa, que son entregados en sistema decimal o hexadecimal, esta unidad está representada por la terminal de video o la impresora de la computadora HP 9816.

### 3.3 Descripción de los ciclos de trabajo del simulador y similitud con la operación real del procesador.

El simulador del procesador de arquitectura mínima, ejecuta cada instrucción, de acuerdo a los tres ciclos de trabajo que se mencionan a continuación:

- a) M1 ciclo de direccionamiento.
- b) M2 ciclo de lectura de memoria.
- c) M3 ciclo de ejecución.

A los ciclos de direccionamiento y de lectura de memoria juntos, se les llama ciclo de búsqueda (15).

Ciclo de direccionamiento (M1).- El contenido del contador de programa (PC), que es un registro de la unidad de control, aparece en el bus de direcciones y se almacena en el registro de direcciones (MAR), de la memoria RAM I para elegir sólo una localidad de las 256, de la figura No. 3.

Ciclo de lectura de memoria (M2).- El contenido de la localidad de memoria direccionada por el registro llamado (MAR), se coloca en el registro de datos de memoria (MDR) y en el bus de datos. Esta información se deposita en el registro de instrucciones (IR), de la figura No. 3.

Ciclo de ejecución (M3).- La parte de la instrucción llamada código de operación, se transfiere al decodificador de ins-

trucciones (ID) y la parte llamada operando, se transfiere al registro auxiliar (AR), todo esto se realiza en la unidad de control, de la figura No. 3.

Al decodificar el código de operación, la unidad de control manda las instrucciones necesarias al CPU y a los dispositivos de apoyo a través del bus de control y direcciones, para ejecutar la instrucción de acuerdo a sus características.

La parte de la instrucción llamada operando, pasa del registro auxiliar (AR) de la unidad de control, al acumulador - - (ACUM), o al registro de uso general (B), o bien a la memoria -- RAM I como dirección de localidad. Los datos de estos registros y localidades de memoria serán procesados por la unidad aritmético-lógica (ALU). Al terminarse este ciclo de máquina, el contador de programa (PC), se incrementa en una unidad.

Estos tres ciclos de trabajo los realiza el procesador a cada instrucción, hasta terminar de ejecutar todas las instrucciones.

Entre el simulador y un sistema de proceso real se pueden ver diferencias tales como:

- 1.- El incremento en una unidad del contador de programa (PC) en un procesador real, se realiza al terminar el ciclo de trabajo M2, cuando la instrucción se encuentra en el registro de instrucciones (IR) de la unidad de control. En el simulador,

el incremento del contador de programa (PC), se realiza al terminar el ciclo de ejecución (M3).

- 2.- En un procesador real una instrucción, puede ocupar una o varias localidades de memoria. En el simulador cada instrucción cabe en una sola localidad de memoria.

## C A P I T U L O 4

## MANUAL DEL USUARIO

## 4.1 Tipos de código de operación

El programa simulador puede ejecutar 38 códigos de operación diferentes, los cuales se pueden agrupar como se muestra a continuación:

## 1.- Grupo de carga.

Los códigos de operación de carga desplazan datos, entre registros o entre registros y memorias. La fuente y el destino de estos datos se especifican en el mismo código de operación.

## 2.- Grupo aritmético-lógico.

Los códigos de operación aritmético-lógico actúan sobre los datos en el acumulador, en el registro de uso general B, la memoria y el operando de la instrucción. Los resultados de las operaciones lógicas se colocan en el acumulador y consecuentemente se establecen los indicadores de estado.

## 3.- Grupo de salto y retorno.

Los códigos de operación de salto y retorno, proporcionan una manera de cambiar el valor del contador de programa (PC) y por lo tanto alteran la secuencia nor

mal de la ejecución del programa. En función del valor de registro de banderas de estado.

4.- Grupo de entrada y salida.

Los códigos de operación de entrada y salida transfieren los datos de los registros y localidades de memoria a dispositivos de entrada y salida.

5.- Grupo de control.

Los códigos de operación de control, varían el tiempo de ejecución de un programa.

Los grupos de códigos de operación descritos anteriormente, afectan directa o indirectamente al acumulador, el cual para indicar qué tipo de dato tiene almacenado, lo hace por medio de los indicadores de estado (registro de banderas), los cuales se muestran a continuación:

S= 1

El acumulador es negativo. El indicador de estado " S " toma el valor de 1.

Z= 1

El acumulador es igual a cero. El indicador de estado " Z ", toma el valor de 1.

C= 1

El acumulador con sobreflujo. Cuando existe en el acumu

lador un número mayor a FFH o bien un número menor a -  
-FFH, el indicador de estado " C ", toma el valor de 1.  
También el valor de esta bandera será afectado con los  
corrimientos aritméticos.

#### 4.2 Formatos de instrucciones nemotécnicas.

Los tres formatos de instrucciones nemotécnicas utilizadas por el procesador simulado se especifican en la figura No. 5.

El formato de instrucción implícito, consiste de un código de operación que implica a un registro en la unidad procesadora. Se puede usar para especificar operaciones tales como "borrar el registro del procesador o negar un registro", o "transferir el contenido de un registro a un segundo registro".

El formato de la instrucción operando inmediato, tiene un código de operación seguido de un operando. Se llama instrucción de operando inmediato, porque el operando sigue inmediatamente después del código de operación de la instrucción. Se puede usar para especificar operaciones tales como "sumar el operando al contenido presente del acumulador" o "transferir el operando al acumulador" o pueden especificar cualquier otra operación a ejecutar, entre el contenido del acumulador y el operando dado.

El formato de instrucción de direccionamiento directo es similar al de operando inmediato, excepto que el operando debe extraerse de la memoria RAM I, de la localidad especificada por la parte de dirección de instrucción. En otras palabras, la operación especificada por el código de operación, se hace entre un registro del procesador y un operando que es almacenado en una localidad de memoria RAM I.

```

*****
* CODIGO * OPERANDO *
* DE * *
* OPERACION * NULO *
*****

```

a) Implícito

```

*****
* CODIGO * OPERANDO *
* DE * *
* OPERACION * INMEDIATO *
*****

```

b) Operando Inmediato

```

*****
* CODIGO * DIRECCION *
* DE * DE *
* OPERACION * OPERANDO *
*****

```

c) Direccionamiento Directo

Figura No. 5 Formato de los tres tipos de instrucciones nemotécnicas utilizadas por el procesador simulado.

#### 4.3 Instrucciones nemotécnicas.

Instrucciones de carga a registros.

Toda instrucción que involucre al acumulador, afectará al registro de banderas de estado, indicando si el valor del acumulador es positivo, negativo, cero o con sobreflujo.

LAC XXH

Instrucción de formato de direccionamiento directo. Carga al acumulador con el contenido de la localidad de memoria especificada por XXH.

LOA XXH

Instrucción de formato de operando inmediato. Carga al acumulador con el operando XXH y afecta al registro de banderas de estado.

EXA XXH

Instrucciones de formato de direccionamiento directo. Intercambia el contenido del acumulador con el contenido de la localidad de memoria especificada por XXH.

RED OOI!

Instrucción de formato implícito. Lee dato en sistema decimal en el dispositivo de entrada y lo carga en el acumulador. Rango del dato de 0 a 255 en sistema decimal.

WRT OOH

Instrucción de formato implícito. Escribe el valor del acumulador, en dispositivo de salida, en sistema decimal dentro del rango 0 a 255.

LMA XXH

Instrucción de formato de direccionamiento directo. Carga una localidad de memoria direccionada por XXH con el contenido del acumulador.

PUSH OOH

Instrucción de formato implícito. Carga al stack con el contenido del registro B, en forma de pila.

POP OOH

Instrucción de formato implícito. Carga al registro B con la última dirección de memoria almacenada en el stack.

LBA OOH

Instrucción de formato implícito. Carga al registro B con el contenido del acumulador.

LOB XXH.

Instrucción de formato de operando inmediato. Carga al registro de uso general B con el operando XXH.

#### Instrucciones Aritméticas.

SUB XXH

Instrucción de formato de operando inmediato. Resta al acumulador el valor del operando XXH.

DBM XXH

Instrucción de formato de direccionamiento directo. Suma el contenido del registro B a la localidad de memoria direccionada por XXH.

SBM XXH

Instrucción de formato de direccionamiento directo. Al contenido de la localidad de memoria especificada por XXH, se le resta el contenido del registro B.

SBA OOH

Instrucción de formato implícito.  
Al contenido del acumulador se le resta el contenido del registro B.

DCB OOH

Instrucción de formato implícito.  
El contenido del registro B se decrementa en una unidad.

INB OOH

Instrucción de formato implícito.  
El contenido del registro B se incrementa en una unidad.

IAC OOH

Instrucción de formato implícito.  
Incrementa al acumulador en una -  
unidad. Afecta al registro de --  
banderas de estado.

DEC OOH

Instrucción de formato implícito.  
Decrementa al acumulador en una -  
unidad. Afecta al registro de --  
banderas de estado.

ADA XXH

Instrucción de formato de operando inmediato. Al acumulador se le suma el valor del operando XXH y el resultado lo almacena en el acumulador.

ADM XXH

Instrucción de direccionamiento directo. Al contenido de la dirección de memoria XXH, se le suma el contenido del acumulador dejando el resultado en dicha localidad de memoria.

SAM XXH

Instrucción de direccionamiento directo. A la localidad de memoria direccionada por XXH, se le resta el contenido del acumulador.

AMA XXH

Instrucción de formato de direccionamiento directo. Al contenido del acumulador se le suma el contenido de la localidad de memoria direccionada por XXH. Afecta al registro de banderas de estado.

CMP. XXH

Instrucción de direccionamiento directo. Compara el contenido del acumulador con el contenido de la localidad de memoria direccionada por XXH; sin modificar el contenido del acumulador. Solamente afecta el valor del registro de banderas de estado.

CMB OOH

Instrucción de formato implícito. Compara el contenido del acumulador con el contenido del registro B. Solamente afecta el valor del registro de banderas de estado.

SLA OOH

Instrucción de formato implícito. Desplaza los bits del acumulador hacia la izquierda, asignando al bit menos significativo el valor de cero, afecta al registro de banderas de estado, dándole a la bandera C el valor del bit más significativo.

SRA OOH

Instrucción de formato implícito. Desplaza los bits del acumulador - hacia la derecha, dándole el valor de cero al bit más significativo. Dándole a la bandera C el valor -- del bit menos significativo.

### Instrucciones lógicas.

XOR XXH

Instrucción de formato de operando inmediato. Efectúa la operación - lógica OR exclusiva, entre el acumulador y el operando XXH. Afecta al registro de banderas de estado.

AND XXH

Instrucción de formato de operando inmediato. Efectúa la operación - AND entre el acumulador y el operando XXH, bit a bit y el resultado se almacena en el acumulador. - Afecta al registro de banderas de estado.

OR XXH

Instrucción de formato de operando inmediato. Efectúa la operación - lógica OR entre el acumulador y el operando XXH, bit a bit y el resultado se almacena en el acumulador. Afecta al registro de banderas de estado.

NEG OOH

Instrucción de formato implícito. Efectúa la operación de negación - (complemento a uno), de los 8 bits del acumulador o sea se invierte - el valor de cada bit. Afecta al - registro de banderas estado.

#### Instrucciones de salto.

JZA XXH

Instrucción de formato de direccio- namiento directo. Salta a la ins- trucción direccionada por XXH, si la bandera de estado Z es igual a uno, de lo contrario el programa - sigue su flujo normal.

JMA XXH

Instrucción de formato de direccio  
namiento directo. Salta a la ins-  
trucción con dirección de memoria  
XXH, si la bandera de estado S es  
igual a uno (acumulador negativo),  
en caso contrario el programa si-  
gue su flujo normal.

JMS XXH

Instrucción de formato de direccio  
namiento directo. Salta a la sub-  
rutina direccionada con XXH.

REGR OOH

Instrucción de formato implícito.  
Regresa a la instrucción siguiente  
de la última instrucción JMS que -  
previamente envió a la subrutina.

JMP XXH

Instrucción de formato de direccio  
namiento directo. Envía el flujo  
del programa a la instrucción XXH,  
sin ninguna condición.

JAC XXH

Instrucción de formato de direccionamiento directo. Salta a la instrucción XXH, si la bandera de estado C es igual a uno.

**Instrucción de Control.**

NOP OOH

Instrucción de formato implícito. Sirve para realizar ciclos de espera sin detener la simulación, sin efectuar operación alguna.

END OOH

Instrucción de formato implícito. Instrucción que se utiliza para finalizar un programa. Manda las instrucciones necesarias para el despliegue final de resultados.

#### 4.4 Mensajes de error.

El programa simulador, presenta los siguientes mensajes de error:

- 1.- Cuando un código de operación con sintaxis incorrecta es introducido en la unidad de entrada, se presenta el siguiente mensaje:

"CODIGO DE OPERACION INCORRECTO, INTRODUCIRLO NUEVAMENTE"

- 2.- Cuando se introduce un operando en la unidad de entrada y no es un número hexadecimal, dentro del rango OOH-FFH, el programa simulador despliega el siguiente mensaje de error:

"OPERANDO NO VALIDO, INTRODUCIRLO NUEVAMENTE"

- 3.- Cuando un programa es introducido en la unidad de entrada y no se finaliza, el simulador no lo procesa y despliega el mensaje de error de:

"NO SE FINALIZO EL PROGRAMA, INTRODUCIRLO NUEVAMENTE"

#### 4.5 Instrucciones de operación del programa.

A continuación se muestran las instrucciones de operación, del programa simulador del procesador de arquitectura mínima:

- 1.- Carga del programa "simulador" del disco a la memoria RAM - de la computadora. Comando LOAD "SIMULADOR", "ENTER".
- 2.- Corrida del programa simulador. Teclee RUN para este efecto.
- 3.- Aparece en pantalla:

```
"SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA"
  "PROGRAMARLO EN LENGUAJE ENSAMBLADOR"
    "MAXIMO 246 INSTRUCCIONES"
  "PROGRAMA QUE REALIZA" :
```

- 4.- Aparece en pantalla:

```
"INTRODUZCA COMENTARIOS DEL PROGRAMA A EJECUTAR"
```

El usuario en este momento podrá titular su programa, sin -- utilizar comas, comillas, punto y coma. Posteriormente se - teclea "ENTER", y aparece la opción de despliegue de resulta - dos en sistema decimal o hexadecimal.

- 5.- Aparece en pantalla "código de operación", para introducir - el primer código de operación, y se teclea "ENTER".

- 6.- Finalizada la introducción del código de operación en la unidad de entrada, el programa simulador inicia su chequeo de sintaxis.
- 7.- De existir error de sintaxis, aparece en pantalla:  
  
"CODIGO DE OPERACION INCORRECTO, INTRODUCIRLO NUEVAMENTE"
- 8.- El usuario introducirá el código de operación correcto.
- 9.- Aparece en pantalla "operando", que es el operando del código de operación anterior. Este debe ser un número hexadecimal comprendido entre 00H-FFH, si se hallase fuera del rango, o bien no perteneciera al sistema hexadecimal, el programa simulador detecta error e indica que se introduzca nuevamente el operando.
- 10.- Los pasos 5, 6, 7, 8 y 9 se repetirán, tantas veces como número de instrucciones esté integrado el programa ensamblador.
- 11.- Se debe finalizar el programa con la instrucción "END 00H", de otra manera, el programa simulador detecta error, detiene la corrida y aparece en pantalla "NO SE FINALIZA EL PROGRAMA".
- 12.- Aparece el comentario de: PARA RESULTADOS EN DECIMAL TECLE "DECI", EN HEXADECIMAL TECLE "HEXA". El usuario elegirá la

opción deseada y teclleará "ENTER".

- 13.- Terminada la introducción del programa, se inicializan los registros de las unidades de proceso y de control.
- 14.- Cada instrucción procesada modifica el estado de los registros del simulador y son impresos en pantalla o en papel en sistema decimal o hexadecimal; para continuar el despliegue se tecllea "ENTER".
- 15.- Aparece en pantalla los resultados finales en sistema decimal o hexadecimal y termina la ejecución del programa ensamblador.
- 16.- Los registros que se imprimen son: el acumulador (ACUM), el registro de uso general (B), el contador de programa (PC), el registro de banderas de estado (S, Z, C), y las 10 últimas localidades de la memoria RAM I de la dirección F6H a la -- FFH.
- 17.- Se le proporciona al usuario la opción de ejecutar nuevamente el programa ensamblador.

#### 4.5.1 Ejemplo de operación del programa.

A continuación se ejemplifica la corrida de un programa ensamblador en el simulador, que realiza la carga del dato numérico AAH al acumulador.

Las instrucciones del programa realizan el siguiente proceso:

La instrucción (LOA, AAH), carga el dato AAH en el acumulador.

La instrucción (END, OOH), finaliza el programa y manda las instrucciones para imprimir el estado final del procesador.

El programa ejemplo 1, muestra cómo se ejecutan las instrucciones anteriores.

## PROGRAMA 1

SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA  
 PROGRAMARLO EN LENGUAJE ENSAMBLADOR  
 MAXIMO 246 INSTRUCCIONES POR PROGRAMA

## PROGRAMA QUE REALIZA:

LOCALIDAD DE MEMORIA	CARGA AL ACUMULADOR CON AAH	INSTRUCCION OPCODE	OP
00H		LOA	AAH
01H		END	00H

TERMINO LA INTRODUCCION DEL PROGRAMA;  
 TOTAL DE INSTRUCCIONES A EJECUTAR 2  
 CODIGOS DE OPERACION Y OPERANDOS VALIDOS

## PROGRAMA ALMACENADO EN LA MEMORIA RAM 1

INICIALIZACION DE REGISTROS DE LA UNIDAD DE PROCESO CENTRAL Y DE  
 LA UNIDAD DE CONTROL

LOA	AAH	ACUM =AAH	REG. B=00H	
			PC =00H	
			ACUM. CERO	Z=00H
			ACUM. SIGNO	S=00H
			SOBREFLUJO	C=00H

M(F6)=00H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....

## ----- ESTADO FINAL DEL PROCESADOR -----

END            00H

ACUM =AAH

REG. B=00H

PC        =01H

ACUM. CERO

Z=00H

ACUM. SIGNO

S=00H

SOBREFLUJO

C=00H

M(F6)=00H

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

-----  
GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR  
ELABORADO EN LA FES CUAUTITLAN.

#### 4.6 Pruebas y resultados.

Al programa "simulador", se le hicieron pruebas con instrucciones de carga, aritmético-lógicas, salto y programas de aplicación de operaciones aritméticas de suma, multiplicación, división, raíz cuadrada y potenciación. Con lo cual se obtuvieron los resultados que se muestran en los programas del 2 al 6.

- 1.- El programa 2, ejemplifica la carga del registro de uso general B, con el dato FAH. Está integrado por dos instrucciones (LOB FAH) y (END OOH). Se puede observar que no se afecta al registro de banderas de estado.
- 2.- El programa 3, ejemplifica una operación aritmética al acumulador, el acumulador se carga con el dato O6H y se le suma el operando 10H. El programa está integrado por las instrucciones (LOA O6H), (ADA 10H) y (END OOH). Se observa -- que se afecta al registro de banderas de estado.
- 3.- El programa 4, ejemplifica la multiplicación de 20H por 03H. El programa está integrado por 12 instrucciones, las cuales realizan la multiplicación de la siguiente forma:  
La localidad de memoria F6H se carga con el número 20H, el registro B se carga también con 20H, la localidad de memoria F7H se carga con el número 03H, a la localidad de memoria F6H se le suma el contenido del registro B tantas veces como lo especifica el contenido de la localidad de memoria F7H, el resultado se obtiene en la localidad de memoria F6H,

y para finalizar este resultado se carga en el acumulador.

- 4.- El programa 5, ejemplifica la división de 09H entre 02H. - El programa está integrado por 10 instrucciones, las cuales realizan la división de la siguiente forma:

La localidad de memoria F6H se carga con el número 09H, el acumulador se carga con el 02H, a la localidad de memoria - F6H se le resta el contenido del acumulador, hasta que el - contenido de la localidad de memoria F6H sea menor o igual al contenido del acumulador, el resultado de la operación - se obtiene en el registro de uso general B.

- 5.- El programa 6, ejemplifica la raíz cuadrada del número 09H. El programa está integrado por 20 instrucciones, las cuales realizan la raíz cuadrada de la siguiente forma:

La localidad de memoria F6H se carga con el 09H, el registro de uso general B se carga con el número 01H, y posteriormente se incrementa en una unidad este valor, se obtiene su cuadrado y se compara con el contenido de la localidad de memoria F6H. Estos pasos se realizan hasta que el - cuadrado del número que se encuentra en el registro B, sea el más cercano menor o igual al valor que se encuentra en - la localidad de memoria F6H. El resultado se obtiene en el acumulador y en el registro de uso general B.

- 6.- El programa 7, ejemplifica el cuadrado del número 05H. El programa está integrado por 9 instrucciones, las cuales rea

lizan la operación de la siguiente forma:

El número 05H, se carga en el registro de uso general B; a la localidad de memoria F6H, que es igual a cero se le suma el contenido del registro B, el acumulador se iguala a cero y se incrementa su valor tantas veces, hasta que el valor del acumulador sea igual al valor del registro B. El resultado se obtiene en la localidad de memoria F6H, pero se coloca en el acumulador este resultado para el despliegue final de resultados.

## PROGRAMA 2

SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA  
 PROGRAMARLO EN LENGUAJE ENSAMBLADOR  
 MAXIMO 246 INSTRUCCIONES POR PROGRAMA

## PROGRAMA QUE REALIZA:

CARGA AL REG B CON FAH

LOCALIDAD DE MEMORIA	INSTRUCCION OPCODE OP
-------------------------	--------------------------

00H	LOB	FAH
01H	END	00H

TERMINO LA INTRODUCCION DEL PROGRAMA;  
 TOTAL DE INSTRUCCIONES A EJECUTAR 2  
 CODIGOS DE OPERACION Y OPERANDOS VALIDOS  
 PROGRAMA ALMACENADO EN LA MEMORIA RAM 1

INICIALIZACION DE REGISTROS DE LA UNIDAD DE PROCESO CENTRAL Y DE  
 LA UNIDAD DE CONTROL

LOB FAH

ACUM =00H

REG. B=FAH

PC =00H

ACUM. CERO

ACUM. SIGMO

SOBREFLUJO

Z=01H

S=00H

C=00H

M(F6)=00H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....

```
----- ESTADO FINAL DEL PROCESADOR -----  
END          00H          ACUM =00H          REG. B=FAH  
                                           PC      =01H  
                                           ACUM. CEFO          Z=01H  
                                           ACUM. SIGNO        S=00H  
                                           SOBREFLUJO        C=00H  
  
M(F6)=00H  
M(F7)=00H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H  
-----
```

GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR  
ELABORADO EN LA FES CUAUTITLAN

## PROGRAMA 3

SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA  
 PROGRAMARLO EN LENGUAJE ENSAMBLADOR  
 MAXIMO 246 INSTRUCCIONES POR PROGRAMA

## PROGRAMA QUE REALIZA:

LA SUMA DE 06H Y 10H RESULTADO EN ACUM	INSTRUCCION
LOCALIDAD DE	OPCODE
MEMORIA	OP

00H	LOA	06H
01H	ADA	10H
02H	END	00H

TERMINO LA INTRODUCCION DEL PROGRAMA;  
 TOTAL DE INSTRUCCIONES A REALIZAR 3  
 CODIGOS DE OPERACION Y OPERANDOS VALIDOS

PROGRAMA ALMACENADO EN LA MEMORIA RAM I

INICIALIZACION DE REGISTROS EN LA UNIDAD DE PROCESO CENTRAL Y DE  
 LA UNIDAD DE CONTROL

LOA 06H

ACUM =06H

REG. B=00H

PC =00H

ACUM. CERO

Z=00H

ACUM. SIGNO

S=00H

SOBREFLUJO

C=00H

M(F6)=00H

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

.....

```

ADA          10H          ACUM  -16H          REG.  B=00H
                                         PC    =01H
                                         ACUM. CERO
                                         ACUM. SIGNO
                                         SOBREFLUJO          Z=00H
                                                         S=00H
                                                         C=00H

```

```

M(F6)=00H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

\*\*\*\*\*

```

----- ESTADO FINAL DEL PROCESADOR -----
END          00H          ACUM  -16H          REG.  B=00H
                                         PC    =02H
                                         ACUM. CERO
                                         ACUM. SIGNO
                                         SOBREFLUJO          Z=00H
                                                         S=00H
                                                         C=00H

```

```

M(F6)=00H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

-----

GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR  
ELABORADO EN LAS FES CUAUTITLAN

## PROGRAMA 4

SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA  
 PROGRAMARLO EN LENGUAJE ENSAMBLADOR  
 MAXIMO 246 INSTRUCCIONES POR PROGRAMA

PROGRAMA QUE REALIZA:  
 MULTIPLICACION DE 20H POR 03H RESULTADO EN ACUM

LOCALIDAD DE MEMORIA	INSTRUCCION OPCODE	OP
00H	LOA	20H
01H	LMA	F0H
02H	LBA	03H
03H	LOA	03H
04H	LMA	F7H
05H	LOA	01H
06H	IAC	00H
07H	DBM	F6H
08H	CMP	F7H
09H	JMA	06H
0AH	LAC	F6H
0BH	END	00H

TERMINO LA INTRODUCCION DEL PROGRAMA;  
 TOTAL DE INSTRUCCIONES A EJECUTAR 12  
 CODIGOS DE OPERACION Y OPERANDOS VALIDOS

PROGRAMA ALMACENADO EN LA MEMORIA RAM I

INICIALIZACION DE REGISTROS DE LA UNIDAD DE PROCESO CENTRAL Y DE  
 LA UNIDAD DE CONTROL

```

LOA      20H      ACUM  =20H      REG.  B=00H
PC      =00H
ACUM.  CERO
ACUM.  SIGNO      Z=00H
SOBREFLUJO      S=00H
C=00H

```

```

M(F6)=00H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LMA      F6H      ACUM  =20H      REG.  B=00H
PC      =01H
ACUM.  CERO      Z=00H
ACUM.  SIGNO      S=00H
SOBREFLUJO      C=00H

```

```

M(F6)=20H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LBA      00H      ACUM  =0H      REG.  B=20H
PC      =02H
ACUM.  CERO      Z=00H
ACUM.  SIGNO      S=00H
SOBREFLUJO      C=00H

```

```

M(F6)=20H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

LOA 03H

ACUM =03H

REG. B=20H  
PC =03H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJO

Z=00H  
S=00H  
C=00H

M(F6)=20H  
M(F7)=00H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

.....  
LMA F7H

ACUM =03H

REG. B=20H  
PC =04H  
ACUM CERO  
ACUM SIGNO  
SOBREFLUJO

Z=00H  
S=00H  
C=00H

M(F6)=20H  
M(F7)=03H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

.....  
LOA 01H

ACUM =01H

REG. B=20H  
PC =05H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJO

Z=00H  
S=00H  
C=00H

M(F6)=20H  
M(F7)=03H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H  
.....

```

IAC      00H          ACUM  =02H          REG.  B=20H
                                         PC   =06H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=20H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
DBM      F6H          ACUM  =02H          REG.  B=20H
                                         PC   =07H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=40H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
CMP      F7H          ACUM  =02H          REG.  B=20H
                                         PC   =08H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=01H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=40H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

JMA 06H

ACUM =02H

REG. B=20H  
 PC =09H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=01H  
 C=00H

M(F6)=40H  
 M(F7)=03H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

IAC 00H

ACUM =03H

REG B=20H  
 PC =06H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=40H  
 M(F7)=03H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

DBM F6H

ACUM =03H

REG. B=20H  
 PC =07H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=60H  
 M(F7)=03H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

```

CMP          F7H          ACUM  =03H          REG.  B=20H
                                         PC   =08H
                                         ACUM. CERO          Z=01H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=60H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
JMA          06H          ACUM  =03H          REG.  B=20H
                                         PC   =09H
                                         ACUM. CERO          Z=01H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=60H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LAC          F6H          ACUM  =60H          REG.  B=20H
                                         PC   =0AH
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=64H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```
----- ESTADO FINAL DEL PROCESADOR -----  
END          00H          ACUM  =60H          REG.  B=20H  
                                           PC    =08H  
                                           ACUM. CERO          Z=00H  
                                           ACUM. SIGNO        S=00H  
                                           SOBREFLUJO         C=00H  
  
M(F6)=60H  
M(F7)=03H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H  
-----
```

GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR  
ELABDRADO EN LA FES CUAUTITLAN

## PROGRAMA 5

SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA  
 PROGRAMARLO EN LENGUAJE ENSAMBLADOR  
 MAXIMO 246 INSTRUCCIONES POP PROGRAMA

## PROGRAMA QUE REALIZA:

LA DIVISION DE 09H ENTRE 02H RESULTADO EN REGISTRO B  
 LOCALIDAD DE MEMORIA INSTRUCCION  
 OPCODE OP

00H	LOA	09H
01H	LMA	F6H
02H	LOA	02H
03H	LOB	00H
04H	INB	00H
05H	SAB	F6H
06H	CMF	F6H
07H	JMA	04H
08H	JZA	04H
09H	END	00H

TERMINO LA INTRODUCCION DEL PROGRAMA;  
 TOTAL DE INSTRUCCIONES A EJECUTAR 10  
 CODIGOS DE OPERACION Y OPERANDOS VALIDOS

PROGRAMA ALMACENADO EN LA MEMORIA RAM I

INICIALIZACION DE REGISTROS DE LA UNIDAD DE PROCESO CENTRAL Y DE  
 LA UNIDAD DE CONTROL

```

LOA      09H          ACUM  =09H          REG.  B=00H
                                         PC   =00H
                                         ACUM. CERO
                                         ACUM. SIGNO          Z=00H
                                         SOBREFLUJO          S=00H
                                         C=00H

```

```

M(F6)=00H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LMA      F6H          ACUM  =09H          REG  B=00H
                                         PC   =01H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LOA      02H          ACUM  =02H          REG.  B=00H
                                         PC   =02H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

LOB 00H  
 ACUM = 02H  
 REG. B=00H  
 PC = 03H  
 ACUM. CERO Z=00H  
 ACUM. SIGNO S=00H  
 SOBREFLUJO C=00H

M(F6)=09H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....

INB 00H  
 ACUM = 02H  
 REG B=01H  
 PC = 04H  
 ACUM. CERO Z=00H  
 ACUM. SIGNO S=00H  
 SOBREFLUJO C=00H

M(F6)=09H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....

SAM F6H  
 ACUM = 02H  
 REG. B=01H  
 PC = 05H  
 ACUM. CERO Z=00H  
 ACUM. SIGNO S=00H  
 SOBREFLUJO C=00H

M(F6)=07H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....

```

CMP          F6H          ACUM  =02H          REG.  B=01H
                                         PC    =06H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=01H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=07H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
JMA          04H          ACUM  =02H          REG.  B=01H
                                         PC    =07H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=01H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=07H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
INB          00H          ACUM  =02H          REG.  B=02H
                                         PC    =04H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=07H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

SAM

F6H

ACUM =02H

REG. B=02H  
 PC =05H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=05H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....  
CMP

F6H

ACUM =02H

REG B=02H  
 PC =06H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=01H  
 C=00H

M(F6)=05H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....  
JMA

04H

ACUM =02H

REG. B=02H  
 PC =07H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=01H  
 C=00H

M(F6)=05H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....

```

INB          00H          ACUM  =02H          REG.  B=03H
                                     PC    =04H
                                     ACUM. CERO          Z=00H
                                     ACUM. SIGNO         S=00H
                                     SOBREFLUJO        C=00H

```

```

M(F6)=05H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
SAM          F6H          ACUM  =02H          REG.  B=03H
                                     PC    =05H
                                     ACUM. CERO          Z=00H
                                     ACUM. SIGNO         S=00H
                                     SOBREFLUJO        C=00H

```

```

M(F6)=03H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
CMP          F6H          ACUM  =02H          REG.  B=07H
                                     PC    =06H
                                     ACUM. CERO          Z=00H
                                     ACUM. SIGNO         S=01H
                                     SOBREFLUJO        C=00H

```

```

M(F6)=03H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

JMA

04H

ACUM = 02H

REG. B=03H  
 PC =07H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=01H  
 C=00H

M(F6)=03H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

INB

00H

ACUM = 02H

REG B=04H  
 PC =04H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=03H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

SAM

F6H

ACUM = 02H

REG. B=04H  
 PC =05H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=01H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

CMP	F6H	ACUM = 02H	REG. B = 04H PC = 06H ACUM. CERO ACUM. SIGNO SOBREFLUJO	Z = 00H S = 00H C = 00H
-----	-----	------------	---	-------------------------------

M(F6) = 01H  
M(F7) = 00H  
M(F8) = 00H  
M(F9) = 00H  
M(FA) = 00H  
M(FB) = 00H  
M(FC) = 00H  
M(FD) = 00H  
M(FE) = 00H  
M(FF) = 00H

.....

JMA	04H	ACUM = 02H	REG B = 04H PC = 07H ACUM. CERO ACUM. SIGNO SOBREFLUJO	Z = 00H S = 00H C = 00H
-----	-----	------------	--	-------------------------------

M(F6) = 01H  
M(F7) = 00H  
M(F8) = 00H  
M(F9) = 00H  
M(FA) = 00H  
M(FB) = 00H  
M(FC) = 00H  
M(FD) = 00H  
M(FE) = 00H  
M(FF) = 00H

.....

JZA	04H	ACUM = 02H	REG. B = 04H PC = 08H ACUM. CERO ACUM. SIGNO SOBREFLUJO	Z = 00H S = 00H C = 00H
-----	-----	------------	---	-------------------------------

M(F6) = 01H  
M(F7) = 00H  
M(F8) = 00H  
M(F9) = 00H  
M(FA) = 00H  
M(FB) = 00H  
M(FC) = 00H  
M(FD) = 00H  
M(FE) = 00H  
M(FF) = 00H

.....

```
----- ESTADO FINAL DEL PROCESADOR -----  
END          00H          ACUM =02H          REG. B=04H  
                                           PC      =09H  
                                           ACUM. CERO          Z=00H  
                                           ACUM. SIGNO        S=00H  
                                           SOBREFLUJO         C=00H  
  
M(F6)=01H  
M(F7)=00H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H  
-----
```

GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR  
ELABORADO EN LA FES CUAUTITLAN.

## PROGRAMA 6

SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA  
 PROGRAMARLO EN LENGUAJE ENSAMBLADOR  
 MAXIMO 246 INSTRUCCIONES POR PROGRAMA

## PROGRAMA QUE REALIZA:

LA RAIZ CUADRADA DEL NUMERO LOCALIDAD DE MEMORIA	09H RESULTADO EN ACUM INSTRUCCION OPCODE	OP
00H	LOA	09H
01H	LMA	F6H
02H	LOA	00H
03H	EXA	F7H
04H	IAC	00H
05H	LBA	00H
06H	LOA	00H
07H	LMA	F7H
08H	IAC	00H
09H	DSH	F7H
0AH	CMB	00H
0BH	JMA	09H
0CH	EXA	F7H
0DH	CMF	F6H
0EH	JMA	03H
0FH	JZA	12H
10H	EXA	F7H
11H	DEC	00H
12H	EXA	F7H
13H	END	00H

TERMINO LA INTRODUCCION DEL PROGRAMA;  
 TOTAL DE INSTRUCCIONES A EJECUTAR 20  
 CODIGOS DE OPERACION Y OPERANDOS VALIDOS

PROGRAMA ALMACENADO EN LA MEMORIA RAM I

INICIALIZACION DE REGISTROS DE LA UNIDAD DE PROCESO CENTRAL Y DE  
 LA UNIDAD DE CONTROL

```

LOA      09H          ACUM  =09H          REG.  B=00H
                                         PC    =00H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=00H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LMA      F6H          ACUM  =09H          REG  B=00H
                                         PC    =01H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LOA      00H          ACUM  =00H          REG.  B=00H
                                         PC    =02H
                                         ACUM. CERO          Z=01H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

EXA      F7H      ACUM  =00H      REG.   B=00H
                                   PC     =03H
                                   ACUM.  CER0      Z=01H
                                   ACUM.  SIGN0     S=00H
                                   SOBREFLUJO      C=00H

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H
.....

IAC      00H      ACUM  =01H      REG   B=00H
                                   PC    =04H
                                   ACUM. CER0      Z=00H
                                   ACUM.  SIGN0     S=00H
                                   SOBREFLUJO      C=00H

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H
.....

LBA      00H      ACUM  =01H      REG.   B=01H
                                   PC     =05H
                                   ACUM.  CER0      Z=00H
                                   ACUM.  SIGN0     S=00H
                                   SOBREFLUJO      C=00H

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H
.....

```

```

LOA      00H          ACUM  =00H          REG.  B=01H
                                         PC    =06H
                                         ACUM. CERO
                                         ACUM. SIGNO          Z=01H
                                         SOBREFLUJO          S=00H
                                         C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LMA      F7H          ACUM  =00H          REG  B=01H
                                         PC    =07H
                                         ACUM. CERO          Z=01H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
IAC      00H          ACUM  =01H          REG.  B=01H
                                         PC    =08H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

DBM          F7H          ACUM  =01H          REG.  B=01H
                                         PC   =09H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=09H
M(F7)=01H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
CMB          00H          ACUM  =01H          REG.  B=01H
                                         PC   =0AH
                                         ACUM. CERO          Z=01H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=09H
M(F7)=01H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
JMA          08H          ACUM  =01H          REG.  B=01H
                                         PC   =0BH
                                         ACUM. CERO          Z=01H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=09H
M(F7)=01H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```



EXA

F7H

ACUM =01H

REG. B=01H  
 PC =03H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=09H  
 M(F7)=01H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

IAC

00H

ACUM =02H

REG B=01H  
 PC =04H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=09H  
 M(F7)=01H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

LBA

00H

ACUM =02H

REG. B=02H  
 PC =05H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=09H  
 M(F7)=01H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H



```

DBM          F7H          ACUM  -01H          REG.  B=02H
                                         PC   =09H
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=00H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=09H
M(F7)=02H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
CMB          00H          ACUM  -01H          REG  B=02H
                                         PC   =0AH
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=01H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=09H
M(F7)=02H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
JMA          08H          ACUM  -01H          REG.  B=02H
                                         PC   =0BH
                                         ACUM. CERO          Z=00H
                                         ACUM. SIGNO        S=01H
                                         SOBREFLUJO         C=00H

```

```

M(F6)=09H
M(F7)=02H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

IAC      00H      ACUM  =02H      REG.  B=02H
                                       PC   =08H
                                       ACUM. CERO
                                       ACUM. SIGNO
                                       SOBREFLUJO      Z=00H
                                                         S=00H
                                                         C=00H

```

```

M(F6)=09H
M(F7)=02H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
```

```

DBM      F7H      ACUM  =02H      REG.  B=02H
                                       PC   =09H
                                       ACUM. CERO      Z=00H
                                       ACUM. SIGNO      S=00H
                                       SOBREFLUJO      C=00H

```

```

M(F6)=09H
M(F7)=04H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
```

```

CMB      00H      ACUM  =02H      REG.  B=02H
                                       PC   =0AH
                                       ACUM. CERO      Z=01H
                                       ACUM. SIGNO      S=00H
                                       SOBREFLUJO      C=00H

```

```

M(F6)=09H
M(F7)=04H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
```

JMA 08H

ACUM =02H

REG. B=02H  
 PC =08H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=01H  
 S=00H  
 C=00H

M(F6)=09H  
 M(F7)=04H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

EXA

F7H

ACUM =04H

REG B=02H  
 PC =0CH  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=09H  
 M(F7)=02H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

CMP

F6H

ACUM =04H

REG. B=02H  
 PC =0DH  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=01H  
 C=00H

M(F6)=09H  
 M(F7)=02H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

JMA 03H

ACUM =04H

REG. B=02H  
PC =0EH  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=01H  
C=00HM(F6)=09H  
M(F7)=02H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

EXA F7H

ACUM =02H

REG B=02H  
PC =03H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=09H  
M(F7)=04H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

IAC 00H

ACUM =03H

REG. B=02H  
PC =04H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=09H  
M(F7)=04H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

```

LBA          00H          ACUM  =03H          REG.   B=03H
                                           PC    =05H
ACUM.  CERO          Z=00H
ACUM.  SIGNO        S=00H
SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=04H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LDA          00H          ACUM  =00H          REG.   B=03H
                                           PC    =06H
ACUM.  CERO          Z=01H
ACUM.  SIGNO        S=00H
SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=04H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
LMA          F7H          ACUM  =00H          REG.   B=03H
                                           PC    =07H
ACUM.  CERO          Z=01H
ACUM.  SIGNO        S=00H
SOBREFLUJO        C=00H

```

```

M(F6)=09H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

IAC          00H          ACUM  =01H          REG.  B=03H
                                         PC   =08H
                                         ACUM. CERO
                                         ACUM. SIGNO
                                         SOBREFLUJO          Z=00H
                                                         S=00H
                                                         C=00H

```

```

M(F6)=009
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

*****
DBM          F7H          ACUM  =01H          REG  B=03H
                                         PC   =08H
                                         ACUM. CERO
                                         ACUM. SIGNO
                                         SOBREFLUJO          Z=00H
                                                         S=00H
                                                         C=00H

```

```

M(F6)=09H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

*****
CMB          00H          ACUM  =01H          REG.  B=03H
                                         PC   =08H
                                         ACUM. CERO
                                         ACUM. SIGNO
                                         SOBREFLUJO          Z=00H
                                                         S=01H
                                                         C=00H

```

```

M(F6)=09H
M(F7)=03H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

*****

```

JMA 08H

ACUM =01H

REG. B=03H  
PC =08H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=01H  
C=00HM(F6)=09H  
M(F7)=03H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

IAC

00H

ACUM =02H

REG. B=03H  
PC =08H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=09H  
M(F7)=03H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

DBM

F7H

ACUM =02H

REG. B=03H  
PC =09H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=09H  
M(F7)=06H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

CMB 00H

ACUM -02H

REG. B=03H  
PC =0AH  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=01H  
C=00HM(F6)=09H  
M(F7)=06H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H\*\*\*\*\*  
JMA 08H

ACUM -02H

REG B=03H  
PC =0BH  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=01H  
C=00HM(F6)=09H  
M(F7)=06H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H\*\*\*\*\*  
IAC 00H

ACUM -03H

REG. B=03H  
PC =0BH  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=09H  
M(F7)=06H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

\*\*\*\*\*

DBM F7H

ACUM =03H

REG. B=03H  
PC =09H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=09H  
M(F7)=09H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

CMB 00H

ACUM =03H

REG B=03H  
PC =1AH  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=01H  
S=00H  
C=00HM(F6)=09H  
M(F7)=09H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

JMA 08H

ACUM =03H

REG. B=03H  
PC =0BH  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=01H  
S=00H  
C=00HM(F6)=09H  
M(F7)=09H  
M(F8)=10H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H



JZA 12H

ACUM =09H

REG. B=03H  
PC =0FH  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=01H  
S=00H  
C=00HM(F6)=09H  
M(F7)=03H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

EXA F7H

ACUM =03H

REG B=03H  
PC =12H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=09H  
M(F7)=09H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H

----- ESTADO FINAL DEL PROCESADOR -----

END

00H

ACUM =03H

REG. B=03H

PC =13H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=00H

C=00H

M(F6)=09H

M(F7)=09H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

-----  
GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR  
ELABORADO EN LA FES CUAUTITLAN.

## PROGRAMA 7

SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA  
 PROGRAMARLO EN LENGUAJE ENSAMBLADOR  
 MAXIMO 246 INSTRUCCIONES POR PROGRAMA

## PROGRAMA QUE REALIZA:

ELEVAR AL CUADRADO A 05H RESULTADO EN ACUM:

LOCALIDAD DE MEMORIA	INSTRUCCION OPCODE	OP
00H	LOA	05H
01H	LBA	00H
02H	LOA	00H
03H	IAC	00H
04H	DBM	F6H
05H	CMB	00H
06H	JMA	03H
07H	EXA	F6H
08H	END	00H

TERMINO LA INTRODUCCION DEL PROGRAMA;  
 TOTAL DE INSTRUCCIONES A EJECUTAR 9  
 CODIGOS DE OPERACION Y OPERANDOS VALIDOS

PROGRAMA ALMACENADO EN LA MEMORIA RAM 1

INICIALIZACION DE REGISTROS DE LA UNIDAD DE PROCESO CENTRAL Y DE  
 LA UNIDAD DE CONTROL

LOA	05H	ACUM =05H	REG. B=00H	
			PC =00H	
			ACUM. CERO	Z=00H
			ACUM. SIGNO	S=00H
			SOBREFLUJO	C=00H

M(F6)=00H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

.....

LBA 00H

ACUM =05H

REG. B=05H  
 PC =01H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=00H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

LBA 00H

ACUM =00H

REG. B=05H  
 PC =02H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=00H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

LBA 00H

ACUM =01H

REG. B=05H  
 PC =03H  
 ACUM. CERO  
 ACUM. SIGNO  
 SOBREFLUJO

Z=00H  
 S=00H  
 C=00H

M(F6)=00H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

DBM

F6H

ACUM. =01H

REG. B=05H

PC =04H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=00H

C=00H

M(F6)=05H

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

.....

CMB

00H

ACUM. =01H

REG. B=05H

PC =05H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=01H

C=00H

M(F6)=05H

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

.....

JMA

03H

ACUM. =01H

REG. B=05H

PC =06H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=01H

C=00H

M(F6)=05H

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

.....

IAC 00H

ACUM =02H

REG. B=05H

PC =03H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=00H

C=00H

M(F6)=05H

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

DBM

F6H

ACUM =02H

REG. B=05H

PC =04H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=00H

C=00H

M(F6)=0AH

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

CMB

00H

ACUM =02

REG. B=05H

PC =05H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=01H

C=00H

M(F6)=0AH

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

```

JMA      03H      ACUM =02H      REG.  B=05H
                                      PC   =06H
                                      ACUM. CERO          Z=00H
                                      ACUM. SIGNO         S=01H
                                      SOBREFLUJO         C=00H

```

```

M(F6)=00H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
IAC      00H      ACUM =03H      REG.  B=05H
                                      PC   =03H
                                      ACUM. CERO          Z=00H
                                      ACUM. SIGNO         S=00H
                                      SOBREFLUJO         C=00H

```

```

M(F6)=00H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

.....
DBM      F6H      ACUM =03H      REG.  B=05H
                                      PC   =04H
                                      ACUM. CERO          Z=00H
                                      ACUM. SIGNO         S=00H
                                      SOBREFLUJO         C=00H

```

```

M(F6)=0FH
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

CMB 00H

ACUM =03H

REG. B=05H

PC =05H

ACUM. CERO

Z=00H

ACUM. SIGNO

S=01H

SOBREFLUJO

C=00H

M(F6)=0FH

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

\*\*\*\*\*

JMA

03H

ACUM =03H

REG. B=05H

PC =06H

ACUM. CERO

Z=00H

ACUM. SIGNO

S=01H

SOBREFLUJO

C=00H

M(F6)=0FH

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

\*\*\*\*\*

IAC

00H

ACUM =04H

REG. B=05H

PC =03H

ACUM. CERO

Z=00H

ACUM. SIGNO

S=00H

SOBREFLUJO

C=00H

M(F6)=0FH

M(F7)=00H

M(F8)=00H

M(F9)=00H

M(FA)=00H

M(FB)=00H

M(FC)=00H

M(FD)=00H

M(FE)=00H

M(FF)=00H

\*\*\*\*\*

DBM F6H

ACUM =04H

REG. B=05H

PC =04H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=00H

C=00H

M(F6)=14H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

\*\*\*\*\*  
 CMB 00H

ACUM =04H

REG. B=05H

PC =05H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=01H

C=00H

M(F6)=14H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

\*\*\*\*\*  
 JMA 03H

ACUM =04H

REG. B=05H

PC =06H

ACUM. CERO

ACUM. SIGNO

SOBREFLUJO

Z=00H

S=01H

C=00H

M(F6)=14H  
 M(F7)=00H  
 M(F8)=00H  
 M(F9)=00H  
 M(FA)=00H  
 M(FB)=00H  
 M(FC)=00H  
 M(FD)=00H  
 M(FE)=00H  
 M(FF)=00H

\*\*\*\*\*

```

IAC          00H          ACUM  -05H          REG.  B=05H
                                     PC    -03H
                                     ACUM. CERO          Z=00H
                                     ACUM. SIGNO          S=00H
                                     SOBREFLUJO          C=00H

```

```

M(F6)=14H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

*****
DBM          F6H          ACUM  -05H          REG.  B=05H
                                     PC    -04H
                                     ACUM. CERO          Z=00H
                                     ACUM. SIGNO          S=00H
                                     SOBREFLUJO          C=00H

```

```

M(F6)=19H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

*****
CMB          00H          ACUM  -05H          REG.  B=05H
                                     PC    -05H
                                     ACUM. CERO          Z=01H
                                     ACUM. SIGNO          S=00H
                                     SOBREFLUJO          C=00H

```

```

M(F6)=19H
M(F7)=00H
M(F8)=00H
M(F9)=00H
M(FA)=00H
M(FB)=00H
M(FC)=00H
M(FD)=00H
M(FE)=00H
M(FF)=00H

```

```

*****

```

JMA 03H

ACUM =05H

REG. B=05H  
PC =06H  
ACUM. CERO  
ACUM. SIGNO  
SOBREFLUJOZ=01H  
S=00H  
C=00HM(F6)=19H  
M(F7)=00H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H.....  
EXA F6H

ACUM =19H

REG. B=05H  
PC =07H  
ACUM CERO  
ACUM SIGNO  
SOBREFLUJOZ=00H  
S=00H  
C=00HM(F6)=05H  
M(F7)=00H  
M(F8)=00H  
M(F9)=00H  
M(FA)=00H  
M(FB)=00H  
M(FC)=00H  
M(FD)=00H  
M(FE)=00H  
M(FF)=00H  
.....

----- ESTADO FINAL DEL PROCESADOR -----

END	00H	ACUM -19H	REG. B=05H	
			PC =08H	
			ACUM. CERO	Z=00H
			ACUM. SIGNO	S=00H
			SOBREFLUJO	C=00H
M(F6)-05H				
M(F7)-00H				
M(F8)-00H				
M(F9)-00H				
M(FA)-00H				
M(FB)-00H				
M(FC)-00H				
M(FD)-00H				
M(FE)-00H				
M(FE)-00H				
M(FE)-00H				

-----

GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR  
ELABORADO EN LA FES CUAUTITLAN.

## CAPITULO 5

## CONCLUSIONES

Esta tesis presenta un programa simulador, hecho en un lenguaje de alto nivel que permite ejecutar programas escritos en lenguaje ensamblador para un sistema electrónico digital de proceso programable llamado procesador de arquitectura mínima, que se define en esta tesis. El programa permite ejecutar hasta 246 instrucciones por programa.

Ejecuta 38 tipos de instrucciones diferentes, las cuales constan de un código de operación de tres caracteres y un operando de tres dígitos en sistema hexadecimal. Simula una memoria de lectura-escritura (RAM I) de 256 localidades de memoria, 246 localidades de la dirección 00H a la F5H, son para almacenar el programa ensamblador, y las 10 localidades restantes de la dirección F6H a la FFH, son para almacenar datos provisionalmente entre cada operación del simulador. Una unidad de procesamiento central, donde se encuentra el acumulador (ACUM), el registro de uso general B, el registro de banderas de estado (S, C, Z), la unidad aritmético-lógica. Una unidad de control que está integrada por el contador de programa (PC), el registro de stack y su apuntador correspondiente (SP), el registro de instrucciones (IR), el decodificador del código de operación (ID), el registro auxiliar (AR).

A la salida se despliega el estado final del contador de -

programa (PC), el acumulador (ACUM), el registro de uso general (B), el registro de banderas de estado (S, C, Z), Y las últimas 10 localidades de la memoria RAM I.

Este programa de computadora, permite iniciarse en el conocimiento de un sistema electrónico digital de proceso programable, al interactuar con los registros de la unidad de procesamiento central y las localidades de memoria. A través de programas sencillos en un lenguaje ensamblador propio, permite leer fácilmente cómo el estatus de la unidad central de procesamiento cambia al ir ejecutando cada instrucción a través del despliegue en pantalla de los distintos registros.

Permite programar un sistema electrónico digital de proceso elemental programable, sin interactuar directamente con el complicado hardware de la computadora; lo cual prueba ser muy conveniente en un ambiente académico, ya que los alumnos pueden practicar la programación en ensamblador, en un ambiente didáctico.

El simulador permite una enseñanza conceptual de una unidad de procesamiento central, sin tener que utilizar unidades de procesamiento central de arquitectura más compleja o modelos comerciales específicos.

## TRABAJO FUTURO

En una segunda versión del programa, es posible simular un sistema de proceso mucho más poderoso, con mayor número de localidades de memoria, registros en la unidad de procesamiento central (CPU) y una variedad mayor de operaciones aritmético-lógicas con la (ALU). También es posible emplear las capacidades gráficas de la computadora, para desplegar de una manera más ilustrativa la arquitectura del sistema y los cambios que en él ocurran al ejecutar un programa.

## B I B L I O G R A F I A

- 1.- Amsterdam Jonatan, "Building a computer in software", Byte vol 10, octubre 1985 pp 113-118.
- 2.- Barden William, "The Z-80 microcomputer handbook", Howard w sams 1978.
- 3.- Ciarcia Steve, "Build the sd 180 single board computer part 1: The Hardware", Byte vol 10 No. 9, septiembre 1985 pp 86-101.
- 4.- Ciarcia Steve, "Build the sd 180 single board computer part 2: The Software", Byte vol. 10 No. 10, octubre 1985 pp 86-101.
- 5.- Coorddon Coeoffrey "Simulación de sistemas", Ed, Diana Mex 1982.
- 6.- Enshoff, J. R. and R. L. Sisson, Design and Use of computer simulation models. New York: The MacMillan Company, 1970.
- 7.- García Núñez Cano, José "Herramientas para el desarrollo de sistemas digitales basadas en microprocesadores simulación", tesis UNAM 1981.

- 8.- Gill, Arthur, "Machine and assembly language programming of the PDP-11", Prentice Hall, 1978.
- 9.- Ghehen Richard, "Simulated computer II", Byte vol. 9, No. 3 pp 274-278.
- 10.- Kelley Cufford, "Ego a homebuilt cpu part 1: The Software", Byte vol 10 no 9, septiembre 1985.
- 11.- Kelley Cufford, "Ego a homebuilt cpu part 2: The Hardware", Byte vol 10, octubre 1985.
- 12.- Korn, G. A. and J. V. Wait. Digital Continuous System Simulation. Englewood Cliffs, NJ: Prentice Hall Inc., 1975.
- 13.- Mc Gwen, Aleen H, "Introduction to computer systems using the PDP-11 and pascal" Mc Graw Hill 1980.
- 14.- Mompin Poblet José, "Microprocesadores y microcomputadoras" serie mundo electrónico, publicaciones Marcombo, S. A.
- 15.- Morris Mano M, "Lógica digital y diseño de computadoras" - Prentice Hall 1982.
- 16.- Murrill, Pw Smith, Cl "Lenguaje de Programación Basic", Internacional Textbook company, 1971.

- 17.- Peatman, John B, "Digital hardware desing", Mc Graw Hill 1980.
- 18.- Roberto, N., D. Andersen, et al. Introduction to computer simulation. Reading, MA: Addison-Wesley publishing Co. - 1983.
- 19.- Robins on Laurence, Specification tecnique communications acm 15,5, May 1972 pp 470-478.
- 20.- Shannon, R. E. Systems Simulations: The art and science. - Englewood Cliffs, Nj: Prentice Hall Inc., 1975.
- 21.- Taub Herbert, "Circuitos digitales y microprocesadores", - Mc Graw Hill 1983.
- 22.- Taylor, L Booth, "Digital Network : Computer Systems", -- John Willey Sons 1978.
- 23.- Jeréz, Gijalvo, "El enfoque de sistemas". Limusa 1983.

## APENDICE 1

## Características del programa fuente.

El programa simulador del procesador de arquitectura mínima fue desarrollado en el lenguaje de alto nivel basic, debido a su característica de universalidad y que puede ser encontrado en la mayoría de los centros de cómputo de universidades e industrias. Una característica del programa, es que se puede correr en cualquier tipo de computadora, haciéndole el mínimo de modificaciones al programa fuente, ya que no se utilizan instrucciones particulares para un tipo de computadora en especial.

Es un programa de 5930 líneas de trabajo, las cuales están divididas en cuatro arreglos de variables alfanuméricas o de archivo, como se muestra a continuación:

- 1.- Arreglo de trabajo llamado "nemo", se caracteriza -- por tener 3 o 4 caracteres cada dato alfanumérico -- (código de operación u operando); se pueden almace-- nar 256 datos (instrucciones) como máximo. Este -- arreglo simula a la memoria de lectura-escritura -- RAM I.
- 2.- Arreglo de trabajo llamado "data", se caracteriza en -- checar la sintaxis de los códigos de operación, o -- sea para ver si se encuentran correctamente escritos, -- es un arreglo que está compuesto de 38 datos (código

de operación previamente definidos).

- 3.- Arreglo de trabajo llamado "hexa", se caracteriza por tener números hexadecimales de 3 caracteres por número la dimensión del arreglo es de 256 datos. Sirve para checar que el operando es un número hexadecimal y que es correcto y le asigna su número decimal correspondiente en una conversión numérica por asignación - que realiza, después de checar la validez de los operandos.
  
- 4.- Arreglo de trabajo llamado "deco", es un arreglo de 38 datos que sirven como decodificador de código de operación y manda el flujo del programa a las subrutinas correspondientes de trabajo, llamadas subrutinas de proceso, las cuales se pueden clasificar como se muestra a continuación:
  - Subrutinas de carga de datos a registros.
  - Subrutinas de operaciones aritmético-lógicas.
  - Subrutinas de salto y retorno.
  - Subrutinas de control de proceso.
  - Subrutinas de entrada y salida de datos.
  - Subrutinas de conversión de sistemas numéricos. (decimal-binario, binario-decimal, decimal-hexadecimal, hexadecimal-decimal).
  
- 5.- Arreglo de trabajo M(N), simula a las 10 últimas loca

lidades de memoria RAM I, de la dirección F6H a la di  
rección FFH, que sirven para almacenar datos provisionales  
durante la ejecución de un programa ensamblador.

**APENDICE 2**  
**LISTADO DEL PROGRAMA FUENTE**

```

1  !PROGRAMA SIMULADOR DE UN PROCESADOR; 8-DIC-87
2  DIM X$ *255),Y$(255) 4 ,M(255),N$(10),Tt$ 100
3  PRINT TAB(15);"SIMULACION DE UN PROCESADOR DE ARQUITECTURA MINIMA"
4  PRINT
5  PRINT TAB(23);"PROGRAMARLO EN LENGUAJE ENSAMBLADOR"
6  PRINT
7  PRINT TAB(22)$" MAXIMO 246 INSTRUCCIONES POR PROGRAMA"
8  PRINT
9  PRINT TAB(10);"PROGRAMA QUE REALIZA:"
10 INPUT "INTRODUCIR COMENTARIOS",Tt$
11 PRINT.
12 PRINT. Tt$
13 PRINT
14 !ARREGLO NEMO
15 D=0
16 FOR K=1 TO 255
17 D=D+1
18 INPUT "CODIGO DE OPERACION",Y$(K)
19 GOTO 33 !CHEQUEO DE SINTAXIS
20 INPUT "OPERANDO EN HEXADECIMAL 00H-PFHJ",X$(K)
21 GOTO 101 !VALIDACION DE OPERANDOS
22 PRINT " "
23 GOSUB 3205 !SUBROUTINA DE IMPRESION
24 IF Y$(K)="END" THEN 29 ! SE FINALIZO EL PROGRAMA
25 NEXT K
26 PRINT
28 PRINT "NO SE FINALIZO EL PROGRAMA"
29 PRINT
30 PRINT "TERMINO LA INTRODUCCION DEL PROGRAMA;
    TOTAL DE INSTRUCCIONES A EJECUTAR";D
31 PRINT " CODIGOS DE OPERACION Y OPERANDOS VALIDOS"
32 GOTO 1643 !LIMPIA LA PANTALLA
33 !ARREGLO DATA
34 IF Y$(K)="SAM" THEN 82 !TOMA EL SIGUIENTE NEMONICO
35 IF Y$(K)="AND" THEN 82 !TOMA EL SIGUIENTE NEMONICO
36 IF Y$(K)="OR" THEN 82 !TOMA EL SIGUIENTE NEMONICO
37 IF Y$(K)="NEG" THEN 82 !TOMA EL SIGUIENTE NEMONICO
38 IF Y$(K)="SAL" THEN 82 !TOMA EL SIGUIENTE NEMONICO
39 IF Y$(K)="SRA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
40 IF Y$(K)="SRM" THEN 82 !TOMA EL SIGUIENTE NEMONICO
41 IF Y$(K)="LAC" THEN 82 !TOMA EL SIGUIENTE NEMONICO
42 IF Y$(K)="LOA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
43 IF Y$(K)="JZA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
44 IF Y$(K)="SUB" THEN 82 !TOMA EL SIGUIENTE NEMONICO
45 IF Y$(K)="JMA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
46 IF Y$(K)="DEC" THEN 82 !TOMA EL SIGUIENTE NEMONICO
47 IF Y$(K)="SBM" THEN 82 !TOMA EL SIGUIENTE NEMONICO

```

```
51 IF Y$(K)+"JMS" THEN 82 !TOMA EL SIGUIENTE NEMONICO
52 IF Y$(K)="JMP" THEN 82 !TOMA EL SIGUIENTE NEMONICO
53 IF Y$(K)="XOR" THEN 82 !TOMA EL SIGUIENTE NEMONICO
54 IF Y$(K)="RED" THEN 82 !TOMA EL SIGUIENTE NEMONICO
55 IF Y$(K)="DCB" THEN 82 !TOMA EL SIGUIENTE NEMONICO
56 IF Y$(K)="WRT" THEN 82 !TOMA EL SIGUIENTE NEMONICO
57 IF Y$(K)="EXA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
58 IF Y$(K)="SBA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
60 IF Y$(K)="NOP" THEN 82 !TOMA EL SIGUIENTE NEMONICO
61 IF Y$(K)="CMB" THEN 82 !TOMA EL SIGUIENTE NEMONICO
62 IF Y$(K)="ADA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
64 IF Y$(K)="ADM" THEN 82 !TOMA EL SIGUIENTE NEMONICO
```

```

66 IF Y$(K)="AMA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
67 IF Y$(K)="INB" THEN 82 !TOMA EL SIGUIENTE NEMONICO
68 IF Y$(K)="CMP" THEN 82 !TOMA EL SIGUIENTE NEMONICO
69 IF Y$(K)="JAC" THEN 82 !TOMA EL SIGUIENTE NEMONICO
70 IF Y$(K)="LMA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
71 IF Y$(K)="REGR" THEN 82 !TOMA EL SIGUIENTE NEMONICO
72 IF Y$(K)="PUSH" THEN 82 !TOMA EL SIGUIENTE NEMONICO
73 IF Y$(K)="POP" THEN 82 !TOMA EL SIGUIENTE NEMONICO
74 IF Y$(K)="LBA" THEN 82 !TOMA EL SIGUIENTE NEMONICO
75 IF Y$(K)="LOB" THEN 82 !TOMA EL SIGUIENTE NEMONICO
76 IF Y$(K)="END" THEN 82 !VERIFICA SI HUBO ERROR
78 INPUT "CODIGO DE OPERACION INCORRECTO, INTRODUCIRLO NUEVAMENTE", Y$(K)
80 GOTO 19 !CHEQUEO DE SINTAXIS
82 GOTO 20 !INTRODUCIR OPERANDO
98 PRINT
100 PRINT
101 !ARREGLO HEXA
102 !VALIDACION DE OPERANDOS
106 IF X$(K) = "00H" THEN 618 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
198 IF X$(K) = "01H" THEN 622 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
110 IF X$(K) = "02H" THEN 626 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
112 IF X$(K) = "03H" THEN 630 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
114 IF X$(K) = "04H" THEN 634 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
116 IF X$(K) = "05H" THEN 638 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
118 IF X$(K) = "06H" THEN 642 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
120 IF X$(K) = "07H" THEN 646 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
122 IF X$(K) = "08H" THEN 650 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
124 IF X$(K) = "09P" THEN 654 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
126 IF X$(K) = "0AH" THEN 658 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
128 IF X$(K) = "0BH" THEN 662 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
130 IF X$(K) = "0CH" THEN 666 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
132 IF X$(K) = "0DH" THEN 670 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
134 IF X$(K) = "0EH" THEN 674 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
136 IF X$(K) = "0FH" THEN 678 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL

```

138	IF	X*(K) = "10H"	THEN	682	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
140	IF	X*(K) = "11H"	THEN	686	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
142	IF	X*(K) = "12H"	THEN	690	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
144	IF	X*(K) = "13H"	THEN	694	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
146	IF	X*(K) = "14H"	THEN	698	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
148	IF	X*(K) = "15H"	THEN	702	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
150	IF	X*(K) = "16H"	THEN	706	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
152	IF	X*(K) = "17H"	THEN	710	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
154	IF	X*(K) = "18H"	THEN	714	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
156	IF	X*(K) = "19H"	THEN	718	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
158	IF	X*(K) = "1AH"	THEN	722	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
160	IF	X*(K) = "1BH"	THEN	726	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
162	IF	X*(K) = "1CH"	THEN	730	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
164	IF	X*(K) = "1DH"	THEN	734	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
166	IF	X*(K) = "1EH"	THEN	738	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
168	IF	X*(K) = "1FH"	THEN	742	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
170	IF	X*(K) = "20H"	THEN	746	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
172	IF	X*(K) = "21H"	THEN	750	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
174	IF	X*(K) = "22H"	THEN	754	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
176	IF	X*(K) = "23H"	THEN	758	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
178	IF	X*(K) = "24H"	THEN	762	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
180	IF	X*(K) = "25H"	THEN	766	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
182	IF	X*(K) = "26H"	THEN	770	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
184	IF	X*(K) = "27H"	THEN	774	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
186	IF	X*(K) = "28H"	THEN	778	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						
188	IF	X*(K) = "29H"	THEN	782	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE DECIMAL						





294	IF	X*(K)	=	"5EH"	THEN	994	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
296	IF	X*(K)	=	"5FH"	THEN	998	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
298	IF	X*(K)	=	"60H"	THEN	1002	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
300	IF	X*(K)	=	"61H"	THEN	1006	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
302	IF	X*(K)	=	"62H"	THEN	1010	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
304	IF	X*(K)	=	"63H"	THEN	1014	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
306	IF	X*(K)	=	"64H"	THEN	1018	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
308	IF	X*(K)	=	"65H"	THEN	1022	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
310	IF	X*(K)	=	"66H"	THEN	1026	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
312	IF	X*(K)	=	"67H"	THEN	1030	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
314	IF	X*(K)	=	"68H"	THEN	1034	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
316	IF	X*(K)	=	"69H"	THEN	1038	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
318	IF	X*(K)	=	"6AH"	THEN	1042	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
320	IF	X*(K)	=	"6BH"	THEN	1046	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
322	IF	X*(K)	=	"6CH"	THEN	1050	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
324	IF	X*(K)	=	"6DH"	THEN	1054	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
326	IF	X*(K)	=	"6EH"	THEN	1058	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
328	IF	X*(K)	=	"6FH"	THEN	1062	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
330	IF	X*(K)	=	"70H"	THEN	1066	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
332	IF	X*(K)	=	"71H"	THEN	1070	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
334	IF	X*(K)	=	"72H"	THEN	1074	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
336	IF	X*(K)	=	"73H"	THEN	1078	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
338	IF	X*(K)	=	"74H"	THEN	1082	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
340	IF	X*(K)	=	"75H"	THEN	1086	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
341	IF	X*(K)	=	"76H"	THEN	1090	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						
342	IF	X*(K)	=	"77H"	THEN	1094	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE		DECIMAL						





438	IF	X*(K) = "ACH"	THEN	1306	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
440	IF	X*(K) = "ADH"	THEN	1310	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
442	IF	X*(K) = "AEH"	THEN	1314	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
444	IF	X*(K) = "AFH"	THEN	1318	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
446	IF	X*(K) = "BOH"	THEN	1322	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
448	IF	X*(K) = "B1H"	THEN	1326	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
450	IF	X*(K) = "B2H"	THEN	1330	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
452	IF	X*(K) = "B3H"	THEN	1334	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
454	IF	X*(K) = "B4H"	THEN	1338	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
456	IF	X*(K) = "B5H"	THEN	1342	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
458	IF	X*(K) = "B6H"	THEN	1346	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
460	IF	X*(K) = "B7H"	THEN	1350	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
462	IF	X*(K) = "B8H"	THEN	1354	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
464	IF	X*(K) = "B9H"	THEN	1358	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
466	IF	X*(K) = "BAH"	THEN	1362	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
468	IF	X*(K) = "BBH"	THEN	1366	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
470	IF	X*(K) = "BCH"	THEN	1370	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
472	IF	X*(K) = "BDH"	THEN	1374	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
474	IF	X*(K) = "BEH"	THEN	1378	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
476	IF	X*(K) = "BFH"	THEN	1382	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
478	IF	X*(K) = "COH"	THEN	1386	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
480	IF	X*(K) = "C1H"	THEN	1390	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
482	IF	X*(K) = "C2H"	THEN	1394	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
484	IF	X*(K) = "C3H"	THEN	1398	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
486	IF	X*(K) = "C4H"	THEN	1402	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		
488	IF	X*(K) = "C5H"	THEN	1406	!LE	ASIGNA	SU	NUMERO
		CORRESPONDIENTE				DECIMAL		

490 IF X*(K) = "C6H"	THEN	1410	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
492 IF X*(K) = "C7H"	THEN	1414	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
494 IF X*(K) = "C8H"	THEN	1418	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
496 IF X*(K) = "C9H"	THEN	1422	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
498 IF X*(K) = "CAH"	THEN	1426	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
500 IF X*(K) = "CBH"	THEN	1430	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
502 IF X*(K) = "CCH"	THEN	1434	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
504 IF X*(K) = "CDH"	THEN	1438	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
506 IF X*(K) = "CEH"	THEN	1442	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
508 IF X*(K) = "CFH"	THEN	1446	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
510 IF X*(K) = "DOH"	THEN	1450	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
512 IF X*(K) = "DIH"	THEN	1454	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
514 IF X*(K) = "D2H"	THEN	1458	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
516 IF X*(K) = "D3H"	THEN	1462	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
518 IF X*(K) = "D4H"	THEN	1466	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
520 IF X*(K) = "D5H"	THEN	1470	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
522 IF X*(K) = "D6H"	THEN	1474	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
524 IF X*(K) = "D7H"	THEN	1478	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
526 IF X*(K) = "D8H"	THEN	1482	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
528 IF X*(K) = "D9H"	THEN	1486	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
530 IF X*(K) = "DAH"	THEN	1490	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
532 IF X*(K) = "DBH"	THEN	1494	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
534 IF X*(K) = "DCH"	THEN	1498	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
536 IF X*(K) = "DDH"	THEN	1502	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
538 IF X*(K) = "DEH"	THEN	1506	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
540 IF X*(K) = "DFH"	THEN	1510	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						

542 IF X0(K) = "EOH"	THEN	1514	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
544 IF X0(K) = "E1H"	THEN	1518	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
546 IF X0(K) = "E2H"	THEN	1522	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
548 IF X0(K) = "E3H"	THEN	1526	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
550 IF X0(K) = "E4H"	THEN	1530	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
552 IF X0(K) = "E5H"	THEN	1534	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
554 IF X0(K) = "E6H"	THEN	1538	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
556 IF X0(K) = "E7H"	THEN	1542	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
558 IF X0(K) = "E8H"	THEN	1546	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
560 IF X0(K) = "E9H"	THEN	1550	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
562 IF X0(K) = "EAH"	THEN	1554	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
564 IF X0(K) = "EBH"	THEN	1558	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
566 IF X0(K) = "ECH"	THEN	1562	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
568 IF X0(K) = "EDH"	THEN	1566	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
570 IF X0(K) = "EEH"	THEN	1570	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
572 IF X0(K) = "EFH"	THEN	1574	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
574 IF X0(K) = "FOH"	THEN	1578	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
576 IF X0(K) = "F1H"	THEN	1582	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
578 IF X0(K) = "F2H"	THEN	1586	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
580 IF X0(K) = "F3H"	THEN	1590	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
582 IF X0(K) = "F4H"	THEN	1594	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
584 IF X0(K) = "F5H"	THEN	1598	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
586 IF X0(K) = "F6H"	THEN	1602	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
588 IF X0(K) = "F7H"	THEN	1606	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
589 IF X0(K) = "F8H"	THEN	1610	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						
590 IF X0(K) = "F9H"	THEN	1614	!LE	ASIGNA	SU	NUMERO
CORRESPONDIENTE DECIMAL						

```

592 IF X$(K) = "FAH" THEN 1618 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
594 IF X$(K) = "FBH" THEN 1622 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
596 IF X$(K) = "FHC" THEN 1626 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
598 IF X$(K) = "FDH" THEN 1630 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
600 IF X$(K) = "FEH" THEN 1634 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
602 IF X$(K) = "FFH" THEN 1638 !LE ASIGNA SU NUMERO
CORRESPONDIENTE DECIMAL
603 INPUT "OPERANDO NO VALIDO, INTRODUCIRLO NUEVAMENTE (OOH-FFH)", X$(K)
606 GOTO 22 !VALIDACION DE OPERANDO
608 GOTO 23
610 IF Er O THEN 616
612 PRINT "NUMEROS HEXADECIMALES CORRECTOS"
614 GOTO 1650 !VA A LIMPIAR LA PANTALLA
616 STOP
617 REM "COMIENZA ASIGNACION DE NUMERO DECIMAL A HEXADECIMAL"
618 X(K)=0
620 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
622 X(K)=1
624 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
626 X(K)=2
628 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
630 X(K)=3
632 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
634 X(K)=4
636 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
638 X(K)=5
640 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
642 X(K)=6
644 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
646 X(K)=7
648 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
650 X(K)=8
652 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
654 X(K)=9
656 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
658 X(K)=10
660 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
662 X(K)=11
664 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
666 X(K)=12
668 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
670 X(K)=13
672 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
674 X(K)=14
676 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL

```

678	X(K)=15	
680	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
682	X(K)=16	
684	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
686	X(K)=17	
688	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
690	X(K)=18	
692	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
694	X(K)=19	
696	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
698	X(K)=20	
700	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
702	X(K)=21	
704	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
706	X(K)=22	
708	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
710	X(K)=23	
712	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
714	X(K)=24	
716	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
718	X(K)=25	
720	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
722	X(K)=26	
724	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
726	X(K)=27	
728	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
730	X(K)=28	
732	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
734	X(K)=29	
736	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
738	X(K)=30	
740	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
742	X(K)=31	
744	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
746	X(K)=32	
748	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
750	X(K)=33	
752	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
754	X(K)=34	
756	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
758	X(K)=35	
760	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
762	X(K)=36	
764	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
766	X(K)=37	
768	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
770	X(K)=38	
772	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
774	X(K)=39	
776	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
778	X(K)=40	
780	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL

782	X(K)=41			
784	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
786	X(K)=42			
788	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
790	X(K)=43			
792	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
794	X(K)=44			
796	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
798	X(K)=45			
800	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
802	X(K)=46			
804	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
806	X(K)=47			
808	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
810	X(K)=48			
812	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
814	X(K)=49			
816	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
818	X(K)=50			
820	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
822	X(K)=51			
824	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
826	X(K)=52			
828	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
830	X(K)=53			
832	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
834	X(K)=54			
836	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
838	X(K)=55			
840	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
842	X(K)=56			
844	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
846	X(K)=57			
848	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
850	X(K)=58			
852	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
854	X(K)=59			
856	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
858	X(K)=60			
860	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
862	X(K)=61			
864	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
866	X(K)=62			
868	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
870	X(K)=63			
872	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
874	X(K)=64			
876	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
878	X(K)=65			
880	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	
882	X(K)=66			
884	GOTO 608	!TOMA EL SIGUIENTE NUMERO	HEXADECIMAL	

886	X(K)=67				
888	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
890	X(K)=68				
892	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
894	X(K)=69				
896	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
898	X(K)=70				
900	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
902	X(K)=71				
904	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
906	X(K)=72				
908	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
910	X(K)=73				
912	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
914	X(K)=74				
916	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
918	X(K)=75				
920	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
922	X(K)=76				
924	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
926	X(K)=77				
928	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
930	X(K)=78				
932	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
934	X(K)=79				
936	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
938	X(K)=80				
940	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
942	X(K)=81				
944	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
946	X(K)=82				
948	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
950	X(K)=83				
952	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
954	X(K)=84				
956	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
958	X(K)=85				
960	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
962	X(K)=86				
964	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
966	X(K)=87				
968	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
970	X(K)=88				
972	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
974	X(K)=89				
976	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
978	X(K)=90				
980	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
982	X(K)=91				
984	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL
986	X(K)=92				
988	GOTO 608	!TOMA	EL	SIGUIENTE	NUMERO HEXADECIMAL

```
990 X(K)=93
992 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
994 X(K)=94
996 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
998 X(K)=95
1000 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1002 X(K)=96
1004 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1006 X(K)=97
1008 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1010 X(K)=98
1012 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1014 X(K)=99
1016 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1018 X(K)=100
1020 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1022 X(K)=101
1024 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1026 X(K)=102
1028 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1030 X(K)=103
1032 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1034 X(K)=104
1036 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1038 X(K)=105
1040 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1042 X(K)=106
1044 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1046 X(K)=107
1048 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1050 X(K)=108
1052 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1054 X(K)=109
1056 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1058 X(K)=110
1060 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1062 X(K)=111
1064 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1066 X(K)=112
1068 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1070 X(K)=113
1072 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1074 X(K)=114
1076 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1078 X(K)=115
1080 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1082 X(K)=116
1084 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1086 X(K)=117
1088 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1090 X(K)=118
1092 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
```

1094	X(K)=119	
1096	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1098	X(K)=120	
1100	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1102	X(K)=121	
1104	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1106	X(K)=122	
1108	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1110	X(K)=123	
1112	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1114	X(K)=124	
1116	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1118	X(K)=125	
1120	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1122	X(K)=126	
1124	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1126	X(K)=127	
1128	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1130	X(K)=128	
1132	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1134	X(K)=129	
1136	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1138	X(K)=130	
1140	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1142	X(K)=131	
1144	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1146	X(K)=132	
1148	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1150	X(K)=133	
1152	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1154	X(K)=134	
1156	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1158	X(K)=135	
1160	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1162	X(K)=136	
1164	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1166	X(K)=137	
1168	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1170	X(K)=138	
1172	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1174	X(K)=139	
1176	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1178	X(K)=140	
1180	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1182	X(K)=141	
1184	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1186	X(K)=142	
1188	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1190	X(K)=143	
1192	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1194	X(K)=144	
1196	GOTO 608	!TOMA EL SIGUIENTE NUMERO HEXADECIMAL

1198 X(K)=145  
1200 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1202 X(K)=146  
1204 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1206 X(K)=147  
1208 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1210 X(K)=148  
1212 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1214 X(K)=149  
1216 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1218 X(K)=150  
1220 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1222 X(K)=151  
1224 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1226 X(K)=152  
1228 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1230 X(K)=153  
1232 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1234 X(K)=154  
1236 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1238 X(K)=155  
1240 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1242 X(K)=156  
1244 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1246 X(K)=157  
1248 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1250 X(K)=158  
1252 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1254 X(K)=159  
1256 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1258 X(K)=160  
1260 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1262 X(K)=161  
1264 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1266 X(K)=162  
1268 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1270 X(K)=163  
1272 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1274 X(K)=164  
1276 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1278 X(K)=165  
1280 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1282 X(K)=166  
1284 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1286 X(K)=167  
1288 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1290 X(K)=168  
1292 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1294 X(K)=169  
1296 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1298 X(K)=170  
1300 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL

1302 X(K)=171  
 1304 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1306 X(K)=172  
 1308 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1310 X(K)=173  
 1312 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1314 X(K)=174  
 1316 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1318 X(K)=175  
 1320 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1322 X(K)=176  
 1324 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1326 X(K)=177  
 1328 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1330 X(K)=178  
 1332 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1334 X(K)=179  
 1336 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1338 X(K)=180  
 1340 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1342 X(K)=181  
 1344 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1346 X(K)=182  
 1348 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1350 X(K)=183  
 1352 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1354 X(K)=184  
 1356 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1358 X(K)=185  
 1360 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1362 X(K)=186  
 1364 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1366 X(K)=187  
 1368 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1370 X(K)=188  
 1372 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1374 X(K)=189  
 1376 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1378 X(K)=190  
 1380 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1382 X(K)=191  
 1384 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1386 X(K)=192  
 1388 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1390 X(K)=193  
 1392 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1394 X(K)=194  
 1396 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1398 X(K)=195  
 1400 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1402 X(K)=196  
 1404 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL

1406 X(K)=197  
1408 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1410 X(K)=198  
1412 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1414 X(K)=199  
1416 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1418 X(K)=200  
1420 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1422 X(K)=201  
1424 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1426 X(K)=202  
1428 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1430 X(K)=203  
1432 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1434 X(K)=204  
1436 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1438 X(K)=205  
1440 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1442 X(K)=206  
1444 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1446 X(K)=207  
1448 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1450 X(K)=208  
1452 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1454 X(K)=209  
1456 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1458 X(K)=210  
1460 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1462 X(K)=211  
1464 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1466 X(K)=212  
1468 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1470 X(K)=213  
1472 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1474 X(K)=214  
1476 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1478 X(K)=215  
1480 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1482 X(K)=216  
1484 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1486 X(K)=217  
1488 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1490 X(K)=218  
1492 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1494 X(K)=219  
1496 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1498 X(K)=220  
1500 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1502 X(K)=221  
1504 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
1506 X(K)=222  
1508 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL

1510 X(K)=223  
 1512 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1514 X(K)=224  
 1516 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1518 X(K)=225  
 1520 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1522 X(K)=226  
 1524 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1526 X(K)=227  
 1528 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1530 X(K)=228  
 1532 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1534 X(K)=229  
 1536 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1538 X(K)=230  
 1540 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1542 X(K)=231  
 1544 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1546 X(K)=232  
 1548 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1550 X(K)=233  
 1552 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1554 X(K)=234  
 1556 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1558 X(K)=235  
 1560 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1562 X(K)=236  
 1564 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1566 X(K)=237  
 1568 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1570 X(K)=238  
 1572 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1574 X(K)=239  
 1576 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1578 X(K)=240  
 1580 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1582 X(K)=241  
 1584 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1586 X(K)=242  
 1588 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1590 X(K)=243  
 1592 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1594 X(K)=244  
 1596 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1598 X(K)=245  
 1600 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1602 X(K)=246  
 1604 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1606 X(K)=247  
 1608 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL  
 1610 X(K)=248  
 1612 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL

```
1614 X(K)=249
1616 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1618 X(K)=250
1620 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1622 X(K)=251
1624 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1626 X(K)=252
1628 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1630 X(K)=253
1632 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1634 X(K)=254
1636 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1638 X(K)=255
1640 GOTO 608 !TOMA EL SIGUIENTE NUMERO HEXADECIMAL
1641 REM "TERMINA CONVERSION HEXADECIMAL A DECIMAL"
1642 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1643 PRINT
1644 PRINT
1646 PRINT TAB(23); "PROGRAMA ALMACENADO EN LA MEMORIA RAM I"
1647 PRINT
1648 INPUT "PARA DESPLIEGUE DE RESULTADOS ES SISTEMA DECIMAL INTRODUCZA DECI
SI NO
ENTER", A#
1650 FOR I1=1 TO 5
1651 PRINT " "
1652 NEXT I1
1654 PRINT "INICIALIZACION DE REGISTROS DE LA UNIDAD DE PROCESO CENTRAL Y DE
CONTROL"
```

## LA UNIDAD DE CONTROL

```

1454 FOR K=1 TO D
1458 IF K=1 THEN 1472 !SALTA LA IMPRESION
1460 IF Aas="DEC1" THEN 1466 !IMPRESION DECIMAL
1462 @SUB 4000 !IMPRESION EN HEKADecimal
1464 @TO 1468 !VA A LA DECODIFICACION
1466 @SUB 4360 !IMPRESION EN DECIMAL
1468 PRINT TAB(25);"....."
1472 INPUT "ENTER PARA CONTINUAR",Mm#
1473 Pp#=V#(K)
1474 Qq#=K#(K)
1476 P#=-1
1478 REM !COMIENZA DECODIFICACION
1479 !ARREGLO DECO
1480 IF V#(K)="DBM" THEN 1744 !SUMA DEL REG B A MEMORIA
1481 IF V#(K)="SUB" THEN 1762 !RESTA DEL ACUMULADOR DF1
1482 IF V#(K)="IAC" THEN 1754 !INCREMENTA ACUMULADOR
1487 IF V#(K)="DEC" THEN 1770 !DECREMENTA EN UNA UNIDAD EL ACUMULADOR
1484 IF V#(K)="LAC" THEN 1784 !CARGA ACUMULADOR CON MEMORIA
1485 IF V#(K)="XO" THEN 2300 !FUNCION LOGICA XOR
1484 IF V#(K)="LOA" THEN 1794 !CARGA ACUMULADOR CON OPERAND
1487 IF V#(K)="SAM" THEN 2872 !A LA MEMORIA SE LE RESTA EL ACUMULADOR
1488 IF V#(K)="JZA" THEN 1804 !SI ACUM+O SALTA
1490 IF V#(K)="JMA" THEN 1820 !SI ACUM O SALTA
1492 IF V#(K)="SEM" THEN 1836 !A EL CONTENIDO DE MEMORIA SE LE RESTA EL REG B
1494 IF V#(K)="JMS" THEN 1854 !SALTO A SUBROUTINA
1496 IF V#(K)="JMP" THEN 1868 !SALTO INCONDICIONAL SIN REGRESO
1498 IF V#(K)="RED" THEN 2156 !LEE DATOS EN DISPOSITIVO DE ENTRADA
1700 IF V#(K)="MRT" THEN 2172 !RESCRIBE DATOS EN DISPOSITIVO DE SALIDA
1702 IF V#(K)="SBA" THEN 1880 !RESTA A EL ACUM EL CONTENIDO DEL REG B
1704 IF V#(K)="NDF" THEN 1894 !NO OPERACION
1706 IF V#(K)="ADA" THEN 1908 !ACUM + DF1 Y DEJA EN ACUM
1707 IF V#(K)="ADM" THEN 1952 !ACUM+DF1 Y DEJA EN MEMO
1708 IF V#(K)="AMA" THEN 1976 !ACUM+MEMO Y DEJA EN ACUM
1710 IF V#(K)="CMP" THEN 2022 !COMPARA Y DEJA RESULTADO EN ACUM
1712 IF V#(K)="JAC" THEN 2056 !SALTA SI C=1
1714 IF V#(K)="LMA" THEN 2070 !CARGA UNA LOCALIDAD DE MEMORIA
1716 IF V#(K)="REGR" THEN 2084 !REGRESO DE SUBROUTINA
1718 IF V#(K)="FUSH" THEN 2092 !CARGA EL ETAD
1720 IF V#(K)="POP" THEN 2114 !SALIDA DEL STACK
1722 IF V#(K)="LBA" THEN 2132 !CARGA AL REGISTRO B CON CONTENIDO DE ACUMULADOR
1724 IF V#(K)="EXA" THEN 2908 !INTERCAMBIO DE ACUM-LOC DE MEMORIA
1724 IF V#(K)="LOB" THEN 2144 !CARGA A REG B CON OPERANDO
1725 IF V#(K)="CMB" THEN 2854 !COMPARA ACUM CON REGISTRO B
1726 IF V#(K)="END" THEN 3154 !FIN DE PROGRAMA
1726 IF V#(K)="AND" THEN 2300 !FUNCION LOGICA AND
1730 IF V#(K)="OR" THEN 2300 !FUNCION LOGICA OR
1732 IF V#(K)="NEG" THEN 2300 !FUNCION LOGICA NOT
1732 IF V#(K)="DEC" THEN 2898 !DECREMENTO DEL REGISTRO B

```

```

1734 IF Y4 (I)="SLA" THEN 2300 'CORRIMIENTO A LA IZQUIERDA
1735 IF Y8 (K)="INE" THEN 2800 'INCREMENTA EL REGISTRO B
1736 IF Y8 (K)="GRA" THEN 2300 'CORRIMIENTO A LA DERECHA
1740 REM "COMIENZAN SUBROUTINAS"
1742 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1744 REM " SUBROUTINA DE SUMA DEL REGISTRO B A LA LOCALIDAD DE MEMORIA"
1745 G=X (K)
1746 M(G)=M(G)+B
1747 C=0
1748 GOSUB 2500 'SUBROUTINA DE BANDERAS
1750 GOTO 3158 'TOMA EL SIGUIENTE NEMONICO
1752 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1754 REM "SUBROUTINA PARA INCREMENTAR ACUMULADOR "
1756 Acum=Acum+1
1757 C=0
1758 GOSUB 2500 'SUBROUTINA DE BANDERAS
1760 GOTO 3158 'TOMA EL SIGUIENTE NEMONICO
1761 !XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1762 !SUBROUTINA ACUM=ACUM-OP1
1764 Acum=Acum-(I (I))
1765 C=0
1767 GOSUB 2500 'SUBROUTINA DE BANDERAS
1768 GOTO 3158 'TOMA EL SIGUIENTE NEMONICO
1769 !*****
1770 !SUB DECREMENTA ACUMULADOR EN UNA UNIDAD
1772 Acum=Acum-1
1773 C=0
1774 GOSUB 2500 'SUBROUTINA DE BANDERAS
1776 GOTO 3158 'TOMA EL SIGUIENTE NEMONICO
1782 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1784 REM "SUBROUTINA CARGA ACUMULADOR CON LOCALIDAD DE MEMORIA"
1785 C=0
1786 G=X (K)
1788 Acum=M(G)
1789 GOSUB 2500 'SUBROUTINA DE BANDERAS
1790 GOTO 3158 'TOMA EL SIGUIENTE NEMONICO
1792 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1794 REM "SUBROUTINA CARGA ACUMULADOR CON OPERANDO"
1795 C=0
1796 Acum=X (K)
1797 GOSUB 2500 'SUBROUTINA DE BANDERAS
1798 GOTO 3158
1802 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1804 REM " SUBROUTINA SALTA SI ACUMULADOR = 0 "
1806 IF Z=0 THEN 3158
1807 Pc=K-1
1808 K=X (K)+1
1810 Z=1
1812 GOTO 1660 'TOMA EL NEMONICO ASIGNADO POR X(K)

```

```

1014 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1020 REM"SUBROUTINA SALTA SI ACUMULADOR ES NEGATIVO"
1022 IF P=0 THEN 3158
1023 P=K-1
1024 K=X(K)+1
1026 P=1
1028 GOTO 1660 !TOMA EL NEMONICO ASIGNADO POR X(K)
1030 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1036 REM"SUBROUTINA RESTA A EL CONTENIDO DE MEMORIA EL CONTENIDO DEL "
1037 REM"REGISTRO DE PROPOSITO GENERAL B"
1038 B=X(K)
1040 M(G)=M(G)-B
1042 C=0
1044 GOSUB 2500 !SUBROUTINA DE BANDERAS
1046 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
1048 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1054 REM"SUBROUTINA SALTA SUBROUTINA"
1055 MU=0
1056 MU=MU+1
1057 IF E(MU)<>0 THEN 1056
1058 E(MU)=1
1059 P=K-1
1060 K=X(K)+1
1061 C=0
1062 GOSUB 2500
1064 GOTO 1660 !TOMA EL NEMONICO ASIGNADO POR X(K)
1066 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1068 REM"SUBROUTINA SALTO INCONDICIONAL"
1069 P=K-1
1070 K=X(K)+1
1071 C=0
1072 GOSUB 2550
1073 GOTO 1660 !TOMA EL NEMONICO ASIGNADO POR X(K)
1078 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1080 REM"SUBROUTINA RESTA A EL ACUMULADOR EL CONTENIDO DEL REG B"
1081 ACUM=ACUM-B
1082 C=0
1083 GOSUB 2500 !SUBROUTINA DE BANDERAS
1084 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
1092 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1094 REM"SUBROUTINA NO OPERACION"
1095 C=0
1096 FOR J=1 TO 1000
1098 NEXT J
1099 GOSUB 2500
1900 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
1906 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1908 REM" ACUM + OP1 Y DEJA RESULTADO EN ACUM "
```

```

1910 Acum=Acum+X(I.)
1911 C=0
1912 GOSUB 2500 !SUBROUTINA DE BANDERAS
1914 GOTO 3158 !TOMA EL SIGUIENTE MEMONCO
1950 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1952 REM"ACUM + MEMORIA Y DEJA RESULTADO EN MEMORIA"
1953 C=0
1954 B=X(K)
1956 M(G)=M(G)+Acum
1957 GOSUB 2500
1958 GOTO 3158 !TOMA EL SIGUIENTE MEMONCO
1974 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
1976 REM"ACUM + MEMORIA Y DEJA EN ACUM"
1978 B=X(K)
1980 Acum=Acum+M(G)
1981 C=0
1982 GOSUB 2500 !SUBROUTINA DE BANDERAS
1984 GOTO 3158 !TOMA EL SIGUIENTE MEMONCO
2020 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2022 REM" COMPARA LOCALIDAD DE MEMORIA CON ACUM"
2024 Acum=Acum-M(I(K))
2025 C=0
2026 GOSUB 2500 !SUBROUTINA DE BANDERAS
2028 Acum=Acum+M(X(I.))
2030 GOTO 3158 !TOMA EL SIGUIENTE MEMONCO
2054 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2056 REM " SUBROUTINA SALTA SI C=1"
2058 IF C=0 THEN 3158
2059 Pc=K-1
2060 K=X(K)+1
2061 PRINT "SOBREFLUJO"
2062 GOTO 1660 !TOMA EL MEMONICO ASIGNADO POR X(K)
2068 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2070 REM"CARGA UNA LOCALIDAD DE MEMORIA CON ACUM"
2071 C=0
2072 G=X(K)
2074 M(G)=Acum
2075 GOSUB 2500
2076 GOTO 3158 !TOMA EL SIGUIENTE MEMONCO
2082 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2084 REM"REGRESO DE UNA SUBROUTINA A PROGRAMA PRINCIPAL"
2085 Mu=0
2086 Mu=Mu+1
2087 IF E(Mu)<>0 THEN 2086
2088 Pc=K-1
2089 K=E(Mu-1)+1
2090 C=0
2091 GOSUB 2500
2092 GOTO 1660 ! TOMA EL MEMONICO ASIGNADO POR K
2094 REM"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2096 !*****
2097! SUBROUTINA DE ENTRADA AL STACK. (PUSH)
2098 L=L+1
2099 C=0
2100 L=L+1
2102 IF Ns(L)<>0 THEN 2100
210

```

```

4  Ns(L)=E
2105 GOSUB 2500
2106 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
2112 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2114 REM "SUBROUTINA DE SALIDA DEL STACI. (POP)"
2115 C=0
2116 L=0
2118 L=L+1
2120 IF Ns(L)<>0 THEN 2118
2122 L=L-1
2124 B=Ns(L)
2126 Ns(L)=0
2127 GOSUB 2500
2128 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
2130 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2132 REM "CARGA AL REGISTRO B CON EL CONTENIDO DEL ACUMULADOR"
2133 C=0
2134 B=Acum
2135 GOSUB 2500 !SUBROUTINA DE BANDERAS
2136 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
2142 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2144 REM "CARGA EL REGISTRO B CON UN OPERANDO"
2145 C=0
2146 B=X(K)
2147 GOSUB 2500 !SUBROUTINA DE BANDERAS
2148 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
2154 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2156 REM "SUBROUTINA LECTURA DE DATO DE ENTRADA"
2158 INPUT "DATO EN DECIMAL DE [0,255]",As
2159 C=0
2160 Acum=As
2161 GOSUB 2500 !SUBROUTINA DE BANDERAS
2162 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
2170 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2172 REM "ESCRITURA DE DATO DEL ACUMULADOR EN DECIMAL"
2174 PRINT "ACUM=",Acum
2176 GOTO 3158 !TOMA EL SIGUIENTE NEMONCO
2178 REM "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
2300 REM "SUBROUTINAS DE OPERACIONES LOGICAS"
2302 A=Acum
2308 FOR L=1 TO B !INICIALIZA LOCALIDADES DE BITS
2310 S(L)=0
2312 T(L)=0
2314 NEXT L
2315 FOR L=1 TO B !CONVERSION DE NUMERO DECIMAL A 6 BITS
2316 A=INT(A)/2
2318 W=A-INT(A)
2320 V=W*2
2322 S(L)=V
2330 NEXT L
2335 A=X(K)

```

```

2334 FOR L=1 TO 8
2337 A=INT(A)/2
2338 W=A-INT(A)
2340 V=W*2
2342 T(L)=V
2354 NEXT L
2355 IF Y$(K)="XOR" THEN 2780
2356 IF Y$(K)="AND" THEN 2366
2358 IF Y$(K)="OR" THEN 2386
2360 IF Y$(K)="NEG" THEN 2406
2362 IF Y$(K)="SLA" THEN 2422
2364 IF Y$(P)="SRA" THEN 2446
2365 REM"*****"
2366 REM"FUNCION AND"
2368 FOR L=1 TO 8
2370 S(L)=S(L)*T(L)
2372 IF S(L)<>1 THEN 2376 !TOMA EL VALOR DE CERO LOGICO
2374 S(L)=1
2375 GOTO 2377 !TOMA EL SIGUIENTE BIT
2376 S(L)=0
2377 NEXT L
2382 GOTO 2470 !CONVERSION E-D
2384 !*****
2386 !FUNCION OR
2388 FOR L=1 TO 8
2390 S(L)=S(L)+T(L)
2392 IF S(L)<>0 THEN 2398 !TOMA EL VALOR DE UNO LOGICO
2394 S(L)=0
2396 GOTO 2400 !TOMA EL SIGUIENTE BIT
2398 S(L)=1
2400 NEXT L
2402 GOTO 2470 ! CONVERSION B-D
2404 !*****
2406 !FUNCION NEG
2408 FOR L=1 TO 8
2410 IF S(L)=1 THEN 2416 !TOMA EL VALOR DE CERO LOGICO
2412 S(L)=1
2414 GOTO 2418 !TOMA EL SIGUIENTE BIT
2416 S(L)=0
2418 NEXT L
2419 GOTO 2470 !CONVERSION B-D
2420 !*****
2422 !DESPLAZAMIENTO HACIA LA IZQUIERDA"
2424 C=S(8)
2426 S(8)=S(7)
2428 S(7)=S(6)
2430 S(6)=S(5)
2432 S(5)=S(4)
2434 S(4)=S(3)
2436 S(3)=S(2)
2438 S(2)=S(1)
2440 S(1)=0
2442 GOTO 2470 !CONVERSION B-I
2444 !*****
2446 !DESPLAZAMIENTO HACIA LA DERECHA"

```

```

244B C=S(1)
2450 S(1)=S(2)
2452 S(2)=S(3)
2454 S(3)=S(4)
2456 S(4)=S(5)
2458 S(5)=S(6)
2460 S(6)=S(7)
2462 S(7)=S(8)
2464 S(8)=0
2466 GOTO 2470 ! CONVERSION B-D
246B !*****
2470 !CONVERSION BINARIA A DECIMAL
2472 FOR L=1 TO 8
2474 S(L)=S(L)*(2^(L-1))
2476 NEXT L
247B A=S(1)+S(2)+S(3)+S(4)+S(5)+S(6)+S(7)+S(8)
2480 Acum=A
2481 GOSUB 2500 !SUBROUTINA DE BANDERAS
2484 GOTO 315B !TOMA EL SIGUIENTE MEMORCO
248C !*****
2500 !SUBROUTINA MODIFICACION DE BANDERAS
2510 IF Acum>255.5 THEN 2549 !AFECTA LA BANDERA C
2520 IF Acum>0 THEN 2620 !AFECTA LA BANDERA F
2530 IF Acum=0 THEN 2670 !AFECTA LA BANDERA Z
2531 IF Acum<-255 THEN 2593 !AFECTA LA BANDERA C NEGATIVO
2540 IF Acum<0 THEN 2730 !AFECTA LA BANDERA P
2541 PRINT "ACUM=";Acum
2549 !SUBROUTINA DE SOBREFLUJO
2550 C=1
2570 P=0
2580 Z=0
2590 GOSUB 4190 !SUBROUTINA DE SOBREFLUJO
2591 RETURN ! VA A LA SUBROUTINA CORRESPONDIENTE
2593 !BANDERA DE SOBREFLUJO NEGATIVO
2594 C=1
2595 P=1
2597 Z=0
259B GOSUB 4190 !SUBROUTINA DE SOBREFLUJO
2599 Acum=-Acum
2600 RETURN !VA A LA SUBROUTINA CORRESPONDIENTE
2620 P=0
2640 C=C
2650 Z=0
2660 RETURN !VA A LA SUBROUTINA CORRESPONDIENTE
2670 Z=1
2680 P=0
2700 C=C
2720 RETURN !VA A LA SUBROUTINA CORRESPONDIENTE
2730 Z=0
2740 P=1
2750 C=C
2760 RETURN !VA A LA SUBROUTINA CORRESPONDIENTE
2770 !*****
2780 !SUBROUTINA ADR

```

```

2790 FOR L=1 TO 8
2800 S(L)=5(L)+T(L)
2810 IF S(L)<>1 THEN 2840 'TOMA EL VALOR LOGICO DE CERO
2820 S(L)=1
2830 GOSUB 2850 'TOMA EL SIGUIENTE BIT
2840 S(L)=0
2850 NEXT L 'TOMA EL SIGUIENTE BIT
2860 GOTO 2470 'CONVERSION B-D
2870 !*****
2872 REM A LA MEMORIA M(K) SE LE RESTA EL ACUMULADOR
2873 C=0
2874 M(K(K))=M(K(K))-Acum
2875 GOSUB 2500
2876 GOTO 3158 'TOMA EL SIGUIENTE MEMONICO
2878 !*****
2880 REM SUBROUTINA DE INCREMENTO AL REGISTRO F
2881 C=0
2882 B=B+1
2883 GOSUB 2500
2884 GOTO 3158 'TOMA EL SIGUIENTE MEMONICO
2886 !*****
2888 REM SUBROUTINA QUE DECREMENTA AL REGISTRO F EN 1
2889 C=0
2890 B=B-1
2891 GOSUB 2500
2892 GOTO 3158 'TOMA EL SIGUIENTE MEMONICO
2894 !*****
2896 REM SUBROUTINA COMPARA EL ACUMULADOR CON REGISTRO B
2898 Acum=Acum-B
2899 C=0
2900 GOSUB 2500 'SUBROUTINA DE BANDERAS
2902 Acum=Acum+B
2904 GOTO 3158 'TOMA EL SIGUIENTE MEMONICO
2906 !*****
2908 REM SUBROUTINA DE INTERCAMBIO ACUM-LOC DE MEMORIA
2910 Z=Acum
2912 C=M(X(K))
2914 Acum=C
2916 M(X(K))=Z
2917 C=0
2918 GOSUB 2500 'SUBROUTINA DE BANDERAS
2920 GOTO 3158 'TOMA EL SIGUIENTE MEMONICO
2922 !*****
3158 NEXT F

```

```

3159 IF Aa$="DECI" THEN 4400 ! ESTADO FINAL EN DECIMAL
3160 !COMIENZA CONVERSION DECIMAL A HEXADECIMAL
3161 FOR I=1 TO 18
3162 PRINT " "
3163 NEXT I
3165 PRINT TAB(20); "===== ESTADO FINAL DEL PROCESADOR ====="
3166 X1=ABS(Acum)
3167 GOSUB 3304 !CONVERSION D-H
3168 PRINT Y$(K),X$(K), TAB(20); "ACUM =" ;RS(2);RS(1);RS(3);
3171 X1=B
3172 GOSUB 3304 !CONVERSION D-H
3173 PRINT TAB(35); "REG. B=" ;RS(2);RS(1);RS(3)
3174 X1=PC
3176 GOSUB 3304 !CONVERSION B-D
3178 PRINT TAB(35); "PC =" ;RS(2);RS(1);RS(3)
3186 X1=Z
3188 GOSUB 3304 !CONVERSION B-D
3190 PRINT TAB(35); "ACUM. CERO Z=" ;RS(2);RS(1);RS(3)
3192 X1=P
3194 GOSUB 3304 !CONVERSION D-H
3196 PRINT TAB(35); "ACUM. SIGNO S=" ;RS(2);RS(1);RS(3)
3198 X1=C
3200 GOSUB 3304 !CONVERSION B-D
3201 PRINT TAB(35); "SOBREFLUJO C=" ;RS(2);RS(1);RS(3)
3292 PRINT TAB(20); "=====
3204 GOTO 4850 !TERMINA SIMULACION
3205 !COMIENZA SUBROUTINA CONVERSION DE D-H
3304 !COMIENZA SUBROUTINA CONVERSION DE D-H
3324 FOR Ks=1 TO 2
3334 X1=INT(X1)/16
3344 Y=X1-INT(X1)
3354 Y=Y*16
3364 R(Ks)=Y
3374 NEXT Ks
3384 FOR Ks=1 TO 2
3394 IF R(Ks)=0 THEN 3554
3404 IF R(Ks)=1 THEN 3574
3414 IF R(Ks)=2 THEN 3594
3424 IF R(Ks)=3 THEN 3614
3434 IF R(Ks)=4 THEN 3634
3444 IF R(Ks)=5 THEN 3654
3454 IF R(Ks)=6 THEN 3674
3464 IF R(Ks)=7 THEN 3694
3474 IF R(Ks)=8 THEN 3714
3484 IF R(Ks)=9 THEN 3734
3494 IF R(Ks)=10 THEN 3754

```

```

3504 IF R(Ks)=11 THEN 3774
3514 IF R(Ks)=12 THEN 3794
3524 IF R(Ks)=13 THEN 3814
3534 IF R(Ks)=14 THEN 3834
3544 IF R(Ks)=15 THEN 3854
3554 R$(Ks)="0"
3564 GOTO 3864
3574 R$(Ks)="1"
3584 GOTO 3864
3594 R$(Ks)="2"
3604 GOTO 3864
3614 R$(Ks)="3"
3624 GOTO 3864
3634 R$(Ks)="4"
3644 GOTO 3864
3654 R$(Ks)="5"
3664 GOTO 3864
3674 R$(Ks)="6"
3684 GOTO 3864
3694 R$(Ks)="7"
3704 GOTO 3864
3714 R$(Ks)="8"
3724 GOTO 3864
3734 R$(Ks)="9"
3744 GOTO 3864
3754 R$(Ks)="A"
3764 GOTO 3864
3774 R$(Ks)="B"
3784 GOTO 3864
3794 R$(Ks)="C"
3804 GOTO 3864
3814 R$(Ks)="D"
3824 GOTO 3864
3834 R$(Ks)="E"
3844 GOTO 3864
3854 R$(Ks)="F"
3864 NEXT Ks
3874 R$(3)="H"
3884 RETURN
4000 X1=ABS(ACUM)
4010 GOSUB 3304 !CONVERSION D-H
4020 PRINT Pp$,Qq$,TAB(20);"ACUM =" ;R$(2);R$(1);R$(3);
4030 X1=B
4040 GOSUB 3304 !CONVERSION D-H
4050 PRINT TAB(35); "REG. B=" ;R$(2);R$(1);R$(3)
4060 X1=Pc
4070 GOSUB 3304 !CONVERSION D-H
4080 PRINT TAB(35);"PC =" ;R$(2);R$(1);R$(3)
4090 X1=Z
4100 GOSUB 3304 !CONVERSION D-H
4110 PRINT TAB(35);"ACUM. CERO Z=" ;R$(2);R$(1);R$(3)
4120 X1=P
4130 GOSUB 3304 !CONVERSION D-H

```

```
4140 PRINT TAB(35);"ACUM. SIGNO          S=";R$(2);R$(1);R$(3)
4150 X1=C
4160 GOSUB 3304 !CONVERSION D-H
4170 PRINT TAB(35); "SOBREFLUJO          C=";R$(2);R$(1);R$(3)
4175 GOSUB 4950 !IMPRESION DE LOC. MEMORIA
4180 RETURN
4190 !SUBROUTINA DE SOBREFLUJO
4200 A=ABS(Acum)
```

```

4210 FOR L=1 TO 8
4220 S(L)=0
4230 NEXT L
4240 FOR L=1 TO 8
4250 A=INT(A)/2
4260 W=A-INT(A)
4270 V=W*2
4280 S(L)=V
4290 NEXT L
4300 FOR L=1 TO 8
4310 S(L)=S(L)*(2(L-1))
4320 NEXT L
4330 A=S(1)+S(2)+S(3)+S(4)+S(5)+S(6)+S(7)+S(8)
4340 Acum=A
4350 RETURN
4359 !*****
4360 !SUBROUTINA DE DESPLIEGUE DE DATOS EN DECIMAL
4362 PRINT Pp$, Cr$, TAB(20); "ACUM="; ABS(Acum)
4363 PRINT TAB(35); "REG. B="; B
4364 PRINT TAB(35); "PC ="; Pc
4366 PRINT TAB(35); "ACUM. CERO Z="; Z
4368 PRINT TAB(35); "ACUM. SIGNO S="; P
4369 PRINT TAB(35); "SOBREFLUJO C="; C
4370 GOSUB 5200 !IMPRESION DE LOC MEMORIA EN DECI
4371 RETURN
4372 !*****
4400 !ESTADO FINAL DEL PROCESADOR EN DECIMAL
4402 FOR Ii=1 TO 14
4404 PRINT " "
4406 NEXT Ii
4408 PRINT "=====ESTADO FINAL DEL PROCESADOR=====
4410 PRINT Y$(K), X(K), TAB(20); "ACUM ="; ABS(Acum)
4412 PRINT TAB(35); "REG. B="; B
4414 PRINT TAB(35); "PC ="; Pc
4416 PRINT TAB(35); "ACUM. CERO Z="; Z
4418 PRINT TAB(35); "ACUM. SIGNO S="; P
4420 PRINT TAB(35); "SOBREFLUJO C="; C
4421 GOSUB 5200 ! IMPRESION DE LOC MEMO EN DECI
4422 PRINT "=====
4424 GOTO 4850 ! EJECUTAR DE NUEVO EL PROGRAMA ENSAMBLADOR
4846 PRINT " "
4850 INPUT "EJECUTAR DE NUEVO EL PROGRAMA SI DE LO CONTRARIO NO", Cb$
4860 IF Cb$ "SI" THEN 5890 !FINAL DEL PROGRAMA SIMULADOR
4870 FOR Wk=246 TO 255
4880 M(Wk)=0
4890 NEXT Wk
4900 Acum=0
4905 B=0
4910 GOTO 1654 ! INICIO DEL PROGRAMA ENSAMBLADOR

```

```

4950 !SUBROUTINA DE IMPRESION DE LOC MEMORIA EN HEXA
4952 X1=M(246)
4954 GOSUB 3304 !CONVERSION D-H
4956 PRINT "M(F6)=";R$(2);R$(1);R$(3)
4958 X1=M(247)
4960 GOSUB 3304 !CONVERSION D-H
4962 PRINT "M(F7)=";R$(2);R$(1);R$(3)
4964 X1=M(248)
4966 GOSUB 3304 !CONVERSION D-H
4968 PRINT "M(F8)=";R$(2);R$(1);R$(3)
4970 X1=M(249)
4972 GOSUB 3304 !CONVERSION D-H
4974 PRINT "M(F9)=";R$(2);R$(1);R$(3)
4976 X1=M(250)
4978 GOSUB 3304 !CONVERSION D-H
4980 PRINT "M(FA)=";R$(2);R$(1);R$(3)
4982 X1=M(251)
4984 GOSUB 3304 !CONVERSION D-H
4986 PRINT "M(FB)=";R$(2);R$(1);R$(3)
4988 X1=M(252)
4990 GOSUB 3304 !CONVERSION D-H
4992 PRINT "M(FC)=";R$(2);R$(1);R$(3)
4994 X1=M(253)
4996 GOSUB 3304 !CONVERSION D-H
4998 PRINT "M(FD)=";R$(2);R$(1);R$(3)
5000 X1=M(254)
5002 GOSUB 3304 !CONVERSION D-H
5004 PRINT "M(FE)=";R$(2);R$(1);R$(3)
5008 X1=M(255)
5010 GOSUB 3304 !CONVERSION D-H
5012 PRINT "M(FE)";R$(2);R$(1);R$(3)
5014 RETURN
5200 !SUBROUTINA DE IMPRESION LOC MEMO EN DECI
5202 PRINT TAB(8);"M(";246;")=";M(246),"M(";247;")=";M(247)
5204 PRINT TAB(8);"M(";248;")=";M(248),"M(";249;")=";M(249)
5206 PRINT TAB(8);"M(";250;")=";M(250),"M(";251;")=";M(251)
5208 PRINT TAB(8);"M(";252;")=";M(252),"M(";253;")=";M(253)
5210 PRINT TAB(8);"M(";254;")=";M(254),"M(";255;")=";M(255)
5212 RETURN
5890 PRINT " "
5900 PRINT " "
5910 PRINT TAB(14); "GRACIAS POR USAR EL SIMULADOR DE UN PROCESADOR"
5920 PRINT TAB(20); "ELABORADO EN LA PES CUAUITLAN"
5930 END

```

## APENDICE 3

Características de la máquina empleada.

Equipo Hewlett Packard HP 9816.

El equipo HP9000 serie 216, dispone de una impresora modelo HP822906A con capacidad de gráficos y una unidad de disco modelo HP2191 con dos compartimientos (drives) para disco de 3 1/2", con una capacidad de memoria de 270000 bytes/disco.

La capacidad de la memoria de la máquina (sin haber sido cargado el sistema) es de 524128 bytes, una vez cargado y configurado el sistema se cuenta con 67184 bytes de memoria disponibles.

La configuración del sistema consiste en cargar el lenguaje del sistema basic.

Para la configuración son necesarios los siguientes tres discos:

SYSTEM DISK

DRIVERS DISK

LANGUAGE EXTENSIONS DISK

El System Disk contiene el lenguaje del sistema Basic 3,0 llamado System BA3.

El Driver Disk contiene varios archivos binarios (BIN). - Cada archivo BIN abre un canal requerido para un particular interface.

El Language Extensions Disk contiene también varios archivos BIN. Cada archivo BIN de extensión del lenguaje, provee comandos o instrucciones no contenidas en el System BA3.

La máquina HP9000, cuenta con un teclado con diversidad de teclas de caracteres, de un arreglo similar al de una máquina de escribir, con algunas diferencias, como que posee una sección exclusivamente numérica.

(CAPS LOCK).-

Define el modo de letras mayúsculas o minúsculas. Al activar la computadora indica el modo en que se encuentra.

(SHIFT).-

La mayoría de las teclas cuentan con una doble función, la cual se obtiene presionando la tecla deseada conjuntamente -- con la (SHIFT).

(ENTER).-

Tiene varias funciones, como:

- a) Para dar entrada a un dato esperando por una instrucción INPUT.
- b) Para almacenar cada línea de un programa.

c) Puede ser usada para ejecutar comandos o cálculos,

(CLR I/O).-

Detiene momentáneamente la ejecución de una operación de entrada y salida (por ejemplo la salida por impresora). En conjunto con (SHIFT), para la ejecución de un programa inmediatamente después de la ejecución de la línea de curso.

(RUN).-

Inicializa la ejecución de un programa, desde el inicio de éste.

(PAUSE).-

Detiene la ejecución de un programa, pudiendo continuarlo con (SHIFT), para cualquier actividad que esté realizando la computadora.

(EXECUTE).-

Tiene las mismas funciones que (ENTER). Se emplea con el teclado numérico.

En la máquina también se encuentran las teclas programables que permiten hacer más funcional el teclado, ya que éstas pueden ser definidas con una función específica. Cuenta con 10 teclas programables, etiquetadas desde la (K0) a la (K9), las cuales también pueden ser activadas conjuntamente con (SHIFT) y accesar de la (K10) a la (K19).

La computadora dispone de los siguientes comandos básicos:

**LOAD.-**

Este comando se emplea para cargar archivos (programas) a la memoria de la máquina, borrando el existente.

**SCRATCH.-**

Este comando borra el programa que se encuentra en la memoria de la computadora.

**LIST.-**

Se emplea para listar un programa, que se encuentra contenido en la memoria de la máquina.

**CAT.-**

Al ejecutar este comando, la máquina dará una lista de -- programas contenidos en el disco.

**RE-STORE.-**

Este comando almacena un programa en un archivo y es cargado a la memoria de la máquina por medio del comando LOAD.