

Revista de la *Universidad Politécnica* de Puerto Rico

Publicado semestralmente por la Universidad Politécnica de Puerto Rico para difundir los hallazgos de la investigación científica que en ella se hace.

VOL. 1

Diciembre 1991

NUM. 2

Programa emulador del microprocesador MC6800 para correrse en sistemas de computadoras personales basadas en procesadores Intel

Juan Arroyo Suárez
Candidato a graduación

Sinopsis

En este artículo se desarrolla un programa para emular el entrenador ET-3400 de Heathkit y el microprocesador MC6800 en una computadora personal IBM o compatible. El entrenador ET-3400 es la herramienta disponible en la Universidad Politécnica de Puerto Rico para experimentar con el microprocesador MC6800; su uso se limita solamente a las horas de laboratorio. Por ser más accesible una microcomputadora IBM, se decidió usar esta plataforma para el desarrollo del programa. Para la emulación del microprocesador MC6800 se usó el código del microprocesador Intel 8086.

Abstract

This paper presents the development of a program to emulate the Heathkit ET-3400 trainer and the MC6800 microprocessor on an IBM or compatible personal computer. The ET-3400 trainer is the only available tool at the Universidad Politécnica de Puerto Rico for experimenting with the MC6800 microprocessor, but its use is limited during lab hours only. Due to the availability of the IBM personal computer, this platform was used for the implementation of the software. In order to emulate the MC6800 microprocessor, the instruction code of the Intel 8086 microprocessor was used.

Introducción

Al tomar el curso de Laboratorio de Microprocesadores resultaba evidente la falta de conocimiento que tenían los estudiantes sobre la programación del microprocesador MC6800. No estaban seguros de lo que muchas de las instrucciones significaban, de cómo se utilizaban los distintos modos de acceso a la memoria, de cómo se afectaban las banderas del registro del código de condición y qué consecuencias tenían. Esta situación, se debe a la falta de experiencia de los estudiantes al momento de poner en práctica la teoría aprendida en un curso teórico sobre la programación del microprocesador MC6800 que tienen que haber tomado.

Por consiguiente, las sesiones de laboratorio se convertían en sesiones de tanteo, ya que muchos de los experimentos requerían modificaciones a los programas contenidos en el manual o diseñar otros programas, dadas las especificaciones del microprocesador MC6800. Esta falta de práctica ha causado el que los experimentos hayan tenido que mantenerse a un nivel inferior al deseado. Tampoco se han incluido experimentos de aplicaciones del microprocesador, por ejemplo, aplicaciones de control de procesos.

La única herramienta disponible en la Universidad Politécnica para la experimentación con el microprocesador MC6800 es el entrenador ET-3400 de Heathkit. Sin embargo, su uso está reservado para el horario del curso de laboratorio, por lo que surge la idea de modelar dicho entrenador en un programa que pudiera correrse en sistemas de microcomputadoras IBM y compatibles, las cuales son más accesibles al estudiante. Dicho programa está orientado hacia la fase de programación del microprocesador MC6800

Programa emulador del microprocesador MC6800

exclusivamente y no es su propósito el suplantar el entrenador ET-3400, sino complementarlo.

Se recomienda usar el programa emulador como parte del curso de programación, de forma tal que el estudiante ya sepa programar el microprocesador MC6800 al tomar el curso de laboratorio. De esta forma, el curso de laboratorio puede reestructurarse para incluir experimentos que cubran la interconexión y programación de controladores de dispositivos externos (interfasing) y aplicaciones del microprocesador exclusivamente. Así se enriquece y se aprovecha mejor el curso de laboratorio.

Filosofía de diseño

La filosofía de diseño del entrenador se basa en la facilidad con la cual el usuario puede operar el sistema. Tomando como modelo el entrenador ET-3400, el programa emulador incluye todas las características relevantes a la fase de programación del microprocesador MC6800. En algunos casos esas características se han mejorado, por ejemplo, el que se tenga acceso en todo momento al contenido de todos los registros del microprocesador MC6800 simultáneamente. Por otro lado, se han añadido facultades como la capacidad para almacenar el contenido de la memoria del microprocesador MC6800 en archivos de disco y la habilidad de mover bloques de memoria de una localización a otra, lo que permite insertar instrucciones o datos sin la necesidad de volver a entrar porciones del programa actual. El programa emulador incluye una calculadora para computar localizaciones de memoria relativas, un aspecto de la programación que resulta tedioso y propenso a errores.

Estructura del programa

El programa se divide en dos unidades fundamentales: la consola y el microprocesador MC6800 virtual. La figura 1 muestra el diagrama de bloque del programa y la relación entre las diferentes unidades y subunidades que componen el programa.

A continuación se presentan las funciones de las dos unidades fundamentales que componen el programa emulador.

1. La consola

La consola es una combinación de pantalla y teclado que sirve de interfase entre el usuario y el programa. Esta lleva a cabo las siguientes tareas:

- a) Despliega información sobre el sistema, como, por ejemplo, las teclas de control del entrenador, el contenido de registros y las localizaciones de memoria, el estado del microprocesador MC6800, el punto de detención (break-point) del programa, el conjunto de instrucciones del microprocesador MC6800 y otros.

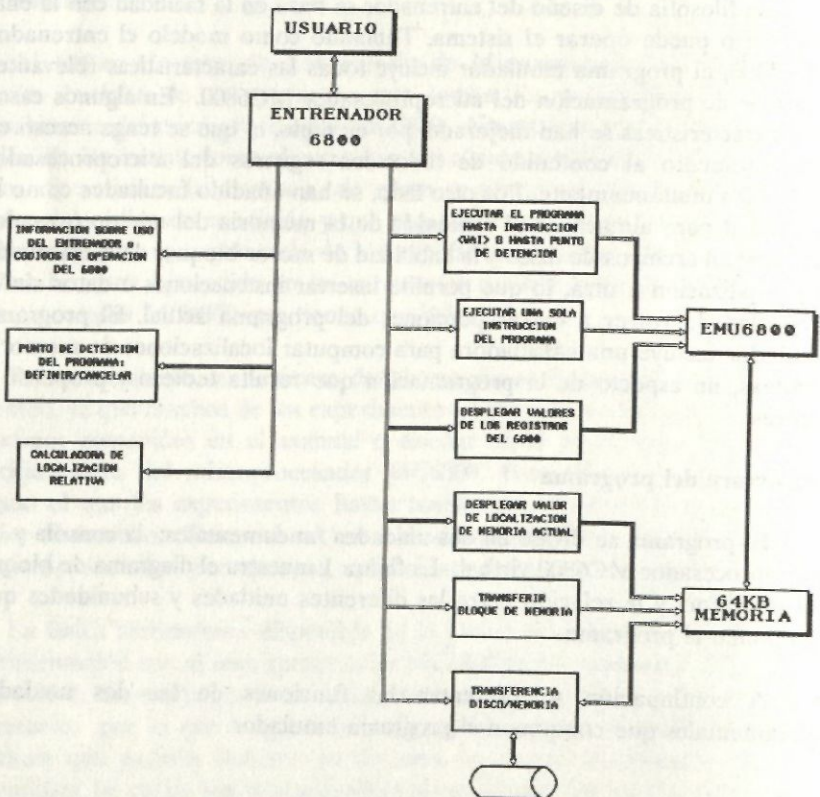


Figura 1. Estructura del entrenador

Programa emulador del microprocesador MC6800

- b) Controla la entrada y salida de datos, por ejemplo, la entrada del código y los datos de un programa para el microprocesador MC6800 y el cambio en el contenido de sus registros.
- c) Manipula el sistema basándose en la información que el usuario especifica. Establece el punto de detención; ejecuta el programa a base de una instrucción a la vez o de corrido; controla la calculadora de localización relativa, la transferencia entre archivos de disco y la memoria y el despliegue del tipo de información sobre el sistema.

2. El microprocesador MC6800 virtual

Debido a que el programa emulador está diseñado para sistemas basados en microprocesadores Intel, en este caso se usó el código del microprocesador 8086 para emular el microprocesador MC6800. Para lograr este objetivo se concentró el esfuerzo en tratar de estructurar el emulador de tal forma que se asemejara lo más posible en cuanto a su funcionamiento a un microprocesador real. El microprocesador MC6800 virtual se divide en dos unidades principales:

a) Unidad de control

La unidad de control decodifica el código de operación para obtener información sobre la localización de los datos necesarios para ejecutar una instrucción, la unidad de ejecución, las banderas que han de actualizarse tan pronto la instrucción se ejecute y la actualización del contenido del contador del programa del microprocesador MC6800.

b) Unidades de ejecución

Las unidades de ejecución son sub-rutinas que ejecutan las instrucciones. Estas sub-rutinas se basan en la información que provee la unidad de control. Para ejecutar las instrucciones los 197 códigos de operación del microprocesador MC6800 se dividieron en términos del tipo de operación que ejecutan, lo que resultó en las siguientes ocho unidades (sub-rutinas):

1. Unidad de transferencia de datos
Transfiere el contenido de una localización de memoria a un registro, o viceversa, y entre un registro y otro.
2. Unidad de transferencia del "stack"
Realiza las transferencias de datos relacionados con las operaciones que envuelven el "stack"
3. Unidad aritmética
Ejecuta las instrucciones que envuelven las operaciones aritméticas, tales como suma y resta, incrementos y decrementos y otras.
4. Unidad lógica
Ejecuta operaciones booleanas y de desplazamiento de dígitos.
5. Unidad de comparación
Ejecuta comparaciones y pruebas.
6. Unidad de cambio de estado de bandera
Cambia el estado de la bandera del registro del código de la condición especificada.
7. Unidad de flujo del programa
Ejecuta instrucciones que cambian, condicional o incondicionalmente, el contenido del contador de programa.
8. Unidad de conversión al sistema BCD (Binary Coded Decimal)
Convierte el resultado de una suma hexadecimal al

formato BCD.

Aunque esta división en unidades se hizo arbitrariamente, la misma responde a una separación más o menos lógica de las operaciones que el microprocesador MC6800 puede ejecutar. En adición a las responsabilidades de la unidad de control antes mencionadas, se le añadió la facultad para verificar las siguientes condiciones inherentes al entrenador: la existencia de un punto de detención del programa y una orden de detención asincrónica y la operación por instrucción. Estas condiciones no forman parte de la arquitectura del microprocesador MC6800 y se llevan a cabo mediante ciertas banderas del sistema que la unidad de control verifica cada vez que ejecuta una instrucción. Además, hay ciertas sub-unidades que se encargan del intercambio de información entre el entrenador y el microprocesador MC6800 virtual que establecen el estado de dichas banderas. La figura 2 muestra el flujograma de la unidad de control.

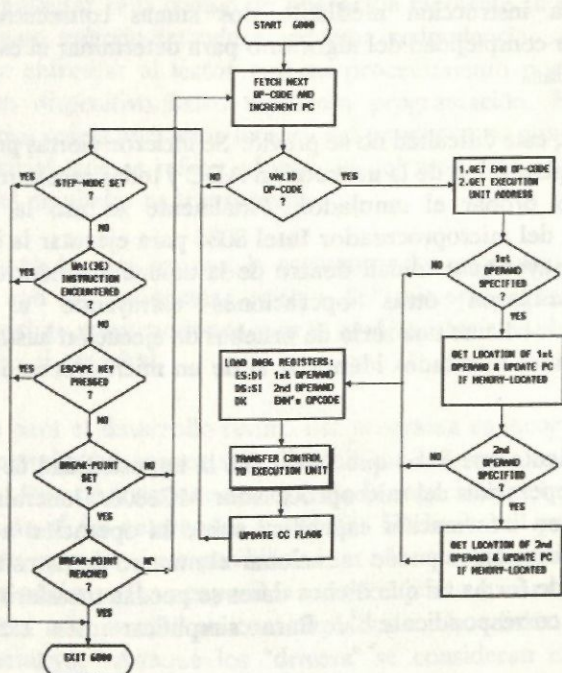


Figura 2. Flujograma de la unidad de control del MC6800 virtual

Dificultades en la implantación del emulador

El problema fundamental en la implantación del programa emulador del microprocesador MC6800 virtual fue la falta de información en sus manuales técnicos. Específicamente la información que se refiere a la arquitectura del microprocesador MC6800 y los algoritmos usados para la ejecución de la instrucción y para la actualización de las banderas del registro de código de condición. Se entiende por algoritmo el procedimiento exacto que sigue el procesador al ejecutar una instrucción. Por ejemplo, la instrucción ADC realiza la suma de dos valores, añadiendo el "carry" a esta suma, con lo cual resultan efectivamente tres los valores a sumarse. Si bien es cierto que la suma es conmutativa, no lo es del todo en términos del microprocesador, ya que el orden en que se efectúe la suma determina, no el resultado numérico, sino el estado de las banderas del registro de código de condición. O sea, que el microprocesador realiza esta suma en forma simultánea. Por esta razón, el emular dicha instrucción mediante dos sumas consecutivas resulta impráctico por la complejidad del algoritmo para determinar el estado de las banderas afectadas.

Ahora bien, esta dificultad no se previó. Se hicieron ciertas presunciones respecto a la implantación de la instrucción ADC y todas resultaron erróneas al momento de probar el emulador. Finalmente se usó la instrucción correspondiente del microprocesador Intel 8086 para ejecutar la instrucción, a expensas de mayor generalidad dentro de la unidad aritmética. En forma similar, se emularon otras operaciones extrayendo el algoritmo correspondiente mediante una serie de pruebas de ejecución hasta lograr que el emulador tuviera resultados idénticos al de un microprocesador MC6800 real.

Otro obstáculo que hubo que salvar fue la imposibilidad de decodificar los códigos de operación del microprocesador MC6800. Generalmente estos códigos contienen información específica sobre la operación a ejecutarse. Entre esta información se puede mencionar el número de datos necesarios y su localización, de forma tal que dichos datos se puedan transferir a la unidad de ejecución correspondiente ¹. Para simplificar esta extracción de

¹ Booth, Taylor L., 1984, *Introduction to Computer Engineering Hardware and Software Design*, John Wiley & Sons Inc., New York, p. 451.

información se ideó una estructura particular en la cual ciertos grupos de dígitos se definieron en términos de la información que portaban. Por ejemplo, el dígito número 12 de los códigos ideados siempre representa el tamaño de los datos envueltos, por ejemplo, 8 o 16 dígitos binarios.

Una vez codificada la información para todos y cada uno de los 197 códigos del microprocesador MC6800 se colocaron ambos conjuntos de códigos en dos tablas, de forma tal que correspondiesen en cuanto a la posición; dentro de cada tabla que correspondiese el código del microprocesador MC6800 y su correspondiente nuevo código.

Conclusiones y desarrollos futuros

El emulador del microprocesador MC6800 se implantó de la manera más difícil. Se usó una arquitectura lo más parecida a la de un procesador real, en vez de implantar cada código de operación mediante su propia subrutina. Aunque este método introduce extrema redundancia, se logra un segundo objetivo: enfrentar al lector con un procedimiento por el cual se puede emular un dispositivo físico mediante programación. El segundo objetivo se relaciona con la operación interna del emulador, en contraposición con el objetivo inicial, que se refiere al usuario del programa en sí, para el cual el interior del emulador es transparente.

Desde el punto de vista interno, la estructura del emulador permite la experimentación con la arquitectura misma del microprocesador. Se le pueden añadir registros e instrucciones, con lo cual se estaría implantando un procesador distinto al MC6800.

Otra opción para el desarrollo futuro del programa es incorporarle un mecanismo que permita interconectarle dispositivos externos, ya sean éstos reales o virtuales. De esta forma se aumentaría la capacidad del entrenador a un nivel semejante al del entrenador ET-3400. De hecho, el programa se estructuró teniendo en mente esta posibilidad, por lo cual las modificaciones al programa serían mínimas. Esto es posible usando "device drivers", los cuales permiten incorporar aditamentos no contemplados en la edición actual de dicho sistema operativo. Aunque los "drivers" se consideran externos al programa, éstos se incorporan sin la necesidad de rediseñar el código original.

La edición presente del entrenador ha sido producto de innumerables pruebas. Estas pruebas han permitido corregir errores encontrados en las

versiones anteriores del entrenador. En este sentido nos sentimos satisfechos. Pero más agradable es el ver que el producto del esfuerzo dedicado a elaborar el programa emulador lo está aprovechando el estudiantado de esta universidad que, en última instancia, es la fuente de mi inspiración y el propósito de la realización de este trabajo. De hecho, ya se contempla incorporar el programa emulador al curso de Laboratorio de Microprocesadores.

Ejemplo para ilustrar el funcionamiento del entrenador

Para ilustrar el funcionamiento del entrenador y demostrar cuán fiel es la operación del microprocesador MC6800 virtual, se preparó un pequeño programa con una variedad de instrucciones que incluyen pruebas, cambio de flujo condicional o incondicional, sub-rutinas, operaciones aritméticas y lógicas, desplazamiento de dígitos y otros. El ejemplo contiene un programa principal y tres sub-rutinas cuyo propósito es multiplicar dos valores y convertir el resultado al formato BCD.

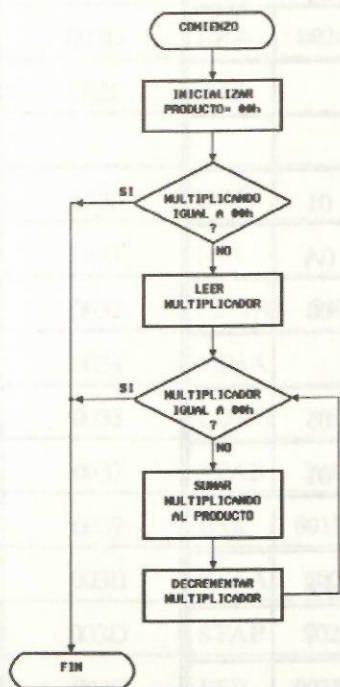
Debido a que el microprocesador MC6800 no posee una instrucción para multiplicar, dicha operación se ejecuta mediante una suma iterativa. La conversión al formato BCD se logra mediante el algoritmo de conversión entre bases numéricas, por ejemplo, dividiendo el valor en la base hexadecimal por la base decimal o 10. El residuo de dicha división constituye el dígito decimal menos significativo (el correspondiente a las unidades). El cociente se convierte entonces en el dividendo y se repite la operación de división por 10. Entonces el segundo residuo constituye el dígito decimal más significativo (el correspondiente a las decenas).

Las divisiones antes mencionadas se llevan a cabo mediante resta iterativas hasta que el dividendo original es menor que el divisor. Se inicializa un contador a cero, de forma tal que por cada resta que se ejecuta se incrementa el valor de dicho contador, el cual, al final del proceso, se convierte en el cociente, mientras que el dividendo original disminuido por las operaciones de resta se convierte en el residuo.

Ambas sub-rutinas, multiplicación y división, suponen que la localización de los valores sobre los cuales han de operar está almacenada en el registro índice (IX) y retornan sus resultados en los acumuladores. Los flujogramas

para ambas sub-rutinas se muestran en la figura 3 y el código correspondiente en las páginas subsiguientes.

SUBROUTINA DE MULTIPLICACION



SUBROUTINA DE DIVISION

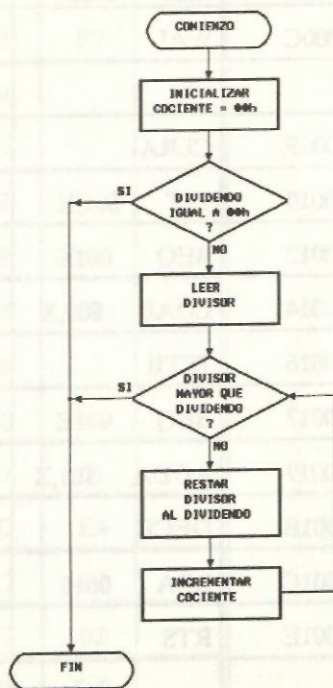


Figura 3. Flujoigramas de sub-rutinas de multiplicación y división

Arroyo Suárez

MEMORIA	MNEMONICO	CODIGO DE OPERACIÓN		
0003	LDX #0050	CE	00	50
0006	BSR 000F	8D	07	
0008	STAA \$02,X	A7	02	
000A	BSR 0030	8D	24	
000C	WAI			
000F	CLRA	4F		
0010	TST \$01,X	6D	01	
0012	BEQ 001E	27	0A	
0014	LDAB \$00,X	E6	00	
0016	TSTB	5D		
0017	BEQ 001E	27	05	
0019	ADDA \$01,X	AB	01	
001B	DECB	5A		
001C	BRA 0016	20	F8	
001E	RTS	39		
001F	CLRA	4F		
0020	TST \$01,X	6D	01	
0022	BEQ 002F	27	0B	
0024	LDAB \$00,X	E6	00	

Programa emulador del microprocesador MC6800

MEMORIA	MNEMONICO	CODIGO DE OPERACIÓN
0026	CMPB \$01,X	E1 01
0028	BMI 002F	2B 05
002A	SUBB \$01,X	E0 01
002C	INCA	4C
002D	BRA 0026	20 F7
002F	RTS	39
0030	INX	08
0031	INX	08
0032	LDAA \$00,X	A6 00
0034	PSHA	36
0035	LDAB # \$0A	C6 0A
0037	STAB \$01,X	E7 01
0039	BSR 001F	8D E4
003B	STAA \$00,X	A7 00
003D	STAB \$02,X	E7 02
003F	BSR 001F	8D DE
0041	ASLB	58
0042	ASLB	58
0043	ASLB	58
0044	ASLB	58
0045	ORAB \$02,X	EA 02

Arroyo Suárez

MEMORIA	MNEMONICO	CODIGO DE OPERACIÓN		
0047	PULA	32		
0048	RTS	39		
0050	PRIMER VALOR			
0051	SEGUNDO VALOR			