

### 1.7 PROGRAMACIÓN DEL MICROCONTROLADOR – CPU08 CPU08 DE LOS MICROCONTROLADORES DE LA FAMILIA HC08

Preparado por: Rangel Alvarado  
Estudiante Graduando de Lic. en Ing. Electromecánica  
Universidad Tecnológica de Panamá  
Panamá, Panamá  
“e-mail”: [issaiass@cwpanama.net](mailto:issaiass@cwpanama.net)  
“web site”: <http://www.geocities.com/issaiass/>

#### ÍNDICE

1.7.1 <i>Introducción</i>	105
1.7.2 <i>¿Qué es un Microcontrolador?</i>	106
1.7.3 <i>CPU08</i>	108
1.7.4 <i>Diagrama de Flujo</i>	110
1.7.5 <i>Código</i>	111
1.7.6 <i>Conclusión</i>	113
1.7.7 <i>Referencias</i>	114
1.7.8 <i>Problemas Propuestos</i>	114

#### 1.7.1 Introducción

---

La lógica de programación es básica para entender el microcontrolador; pero en sí, ¿qué es un microcontrolador?. Primero hay que definir algo que siempre entra en confusión: los microcontroladores y microprocesadores.

Un microprocesador, es un circuito electrónico que funciona como la unidad central de procesamiento o CPU de un computador, proviendo de control computacional,. Los microprocesadores son utilizados en sistemas electrónicos avanzados, como automóviles, jets, computadoras impensoras, etc.

Un microprocesador no es una computadora completa. Esta no contiene grandes cantidades de memoria o tiene la habilidad de comunicarse con dispositivos de entrada, como teclados, joysticks, o con dispositivos de salida, como lo son monitores e impresoras. Un circuito integrado diferente, el microcontrolador, es una computadora completa en una sola pastilla, conteniendo todos los elementos básicos de un microprocesador y rodeado con otras funciones especiales. Los microcontroladores son utilizados en los video juegos, grabadoras de video cassettes (VCRs), automóviles y otras máquinas.<sup>1</sup>

En primera instancia, se explora el concepto del microcontrolador un poco más a nivel macro, luego se hace mención de los registros del microcontrolador y como pueden ser usados comunmente, para luego entrar a un programa ejemplo empleando las instrucciones básicas, en donde el software interactúa con el CPU.

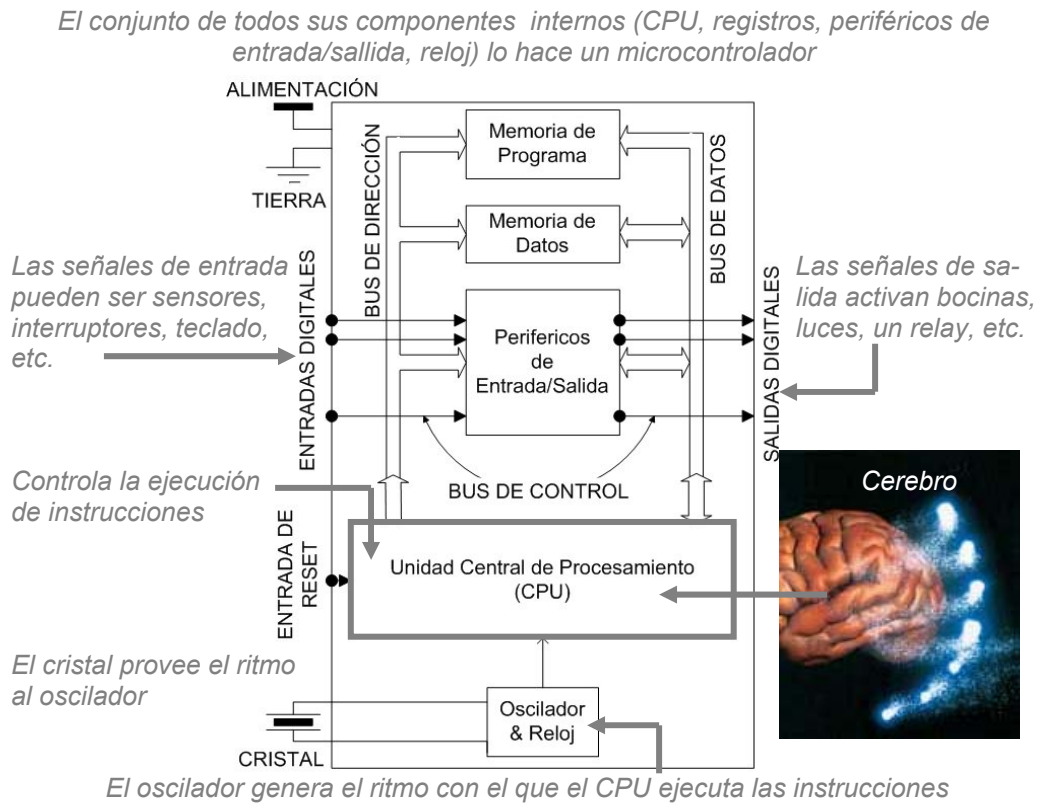
---

<sup>1</sup> “Microsoft® Encarta® Encyclopedia 2003. © 1993-2002 Microsoft Corporation. All rights reserved”.

### 1.7.2 ¿Qué es un microcontrolador?

En esta sección, referirse a las figuras 73 hasta 75.

El *microcontrolador* es un dispositivo que se utiliza para controlar máquinas, por medio de programas para luego dejarlo trabajando autónomamente. En un lenguaje más técnico, el *microcontrolador* es una computadora la cual posee: CPU, memoria, un oscilador y registros de entrada/salida; si el sistema carece de alguno de los componentes, se le llama *microprocesador*. En una computadora casera, el CPU es el *microprocesador* de nuestro *microcontrolador* y es el cerebro el cual comanda todas las acciones.



*Figura 73. Diagrama de Bloques de un Microcontrolador. Cuando en un mismo integrado se encuentran unidos el CPU, memoria, oscilador, registros de entrada/salida, se dice que es un microcontrolador. El CPU, es el cerebro o la parte del microcontrolador que controla la ejecución de instrucciones.*

A la aplicación final del microcontrolador se le llama sistema embebido, el cual es el producto final como un “mouse”, una calculadora, refrigeradora, celular, palm, etc.



Figura 74. Ejemplos Comunes de Sistemas Embebidos. Los sistemas embebidos son la aplicación final de un sistema con microcontrolador. (a) Equipo de medición. (b) Equipo casero. (c) Consola de videojuego. (d) Equipo de uso personal.

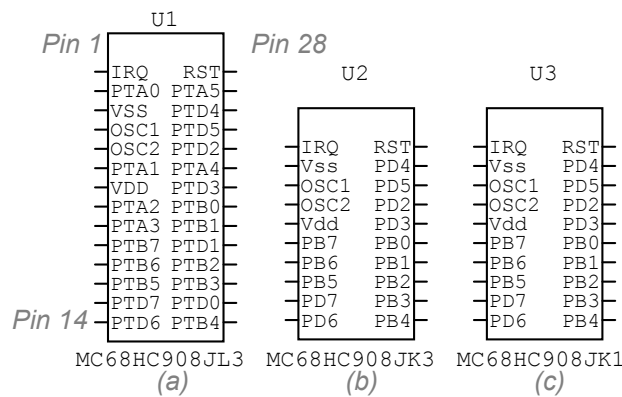


Figura 75. Esquemáticos de los Encapsulados de los Microcontroladores de la Familia HC08 de Motorola. Microcontrolador: (a) 68HC908JL3 (28 pines). (b) 68HC908JK3 (20 pines). (c) 68HC908JK1 (20 pines).

Los microcontroladores de la Familia HC08, se han caracterizado por ser los más famosos por su versatilidad y costo, algunos de estos son el 68HC908JL3, 68HC908JK3 y 68HC908JK1 (figura 75).

### 1.7.3 CPU08

#### 1.7.3.1 Analogía del CPU08 con el Cerebro Humano

En esta sección referirse a la figura 76

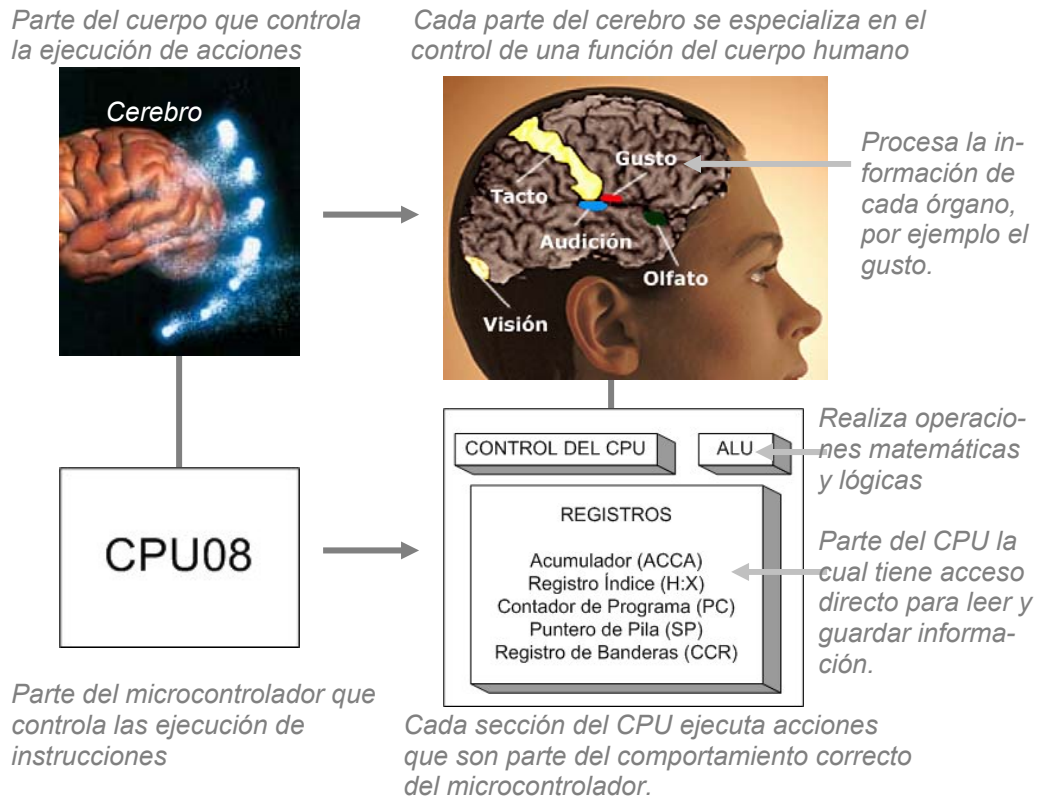
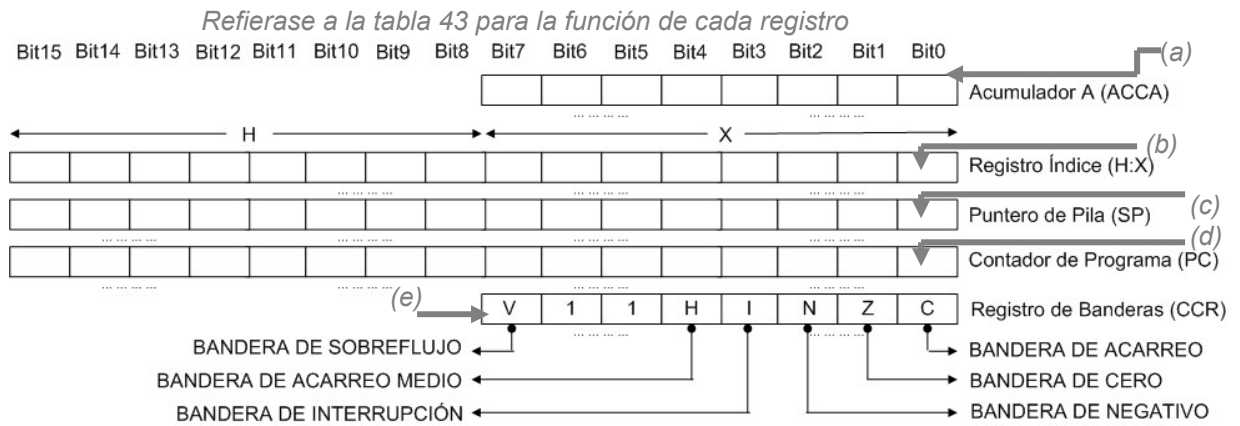


Figura 76. Analogía Entre el Cerebro y el CPU de un Microcontrolador. El CPU es el cerebro de todo sistema con microcontrolador este controla las acciones de las instrucciones; mientras que los registros son interna del CPU al cual se tiene acceso directo que mezcla acciones.

El CPU al igual que el cerebro, es la parte fundamental de cualquier cuerpo humano, la cual se encarga de tomar desiciones en un momento dado de que se debe de hacer. A su vez, internamente el cerebro está dividido en secciones o localidades que tiene un vínculo estrecho para acceder a toda la información proveniente de los sentidos.

**1.7.3.2 Registros del CPU08**

En esta sección refiérase a la figura 77 y la tabla 43.



*Figura 77. Registros del CPU08. (a) El registro A es un registro usado para operaciones lógicas y aritméticas. (b) El registro H:X comunmente se utiliza para moverse alrededor de la memoria. (c) El SP contiene la dirección del posible siguiente dato a almacenarse en la pila. (d) El PC contiene la siguiente dirección de la instrucción a usarse. (e) El CCR indica el resultado de las operaciones efectuadas por el CPU.*

Los registros son localidades que se encuentran “alambradas” con la lógica del CPU en el cual tiene acceso directo a la información que se guarde en ellos. Los registros del CPU se muestran resumidos en la tabla 43.

*Tabla 43. Resumen de Registros del CPU08*

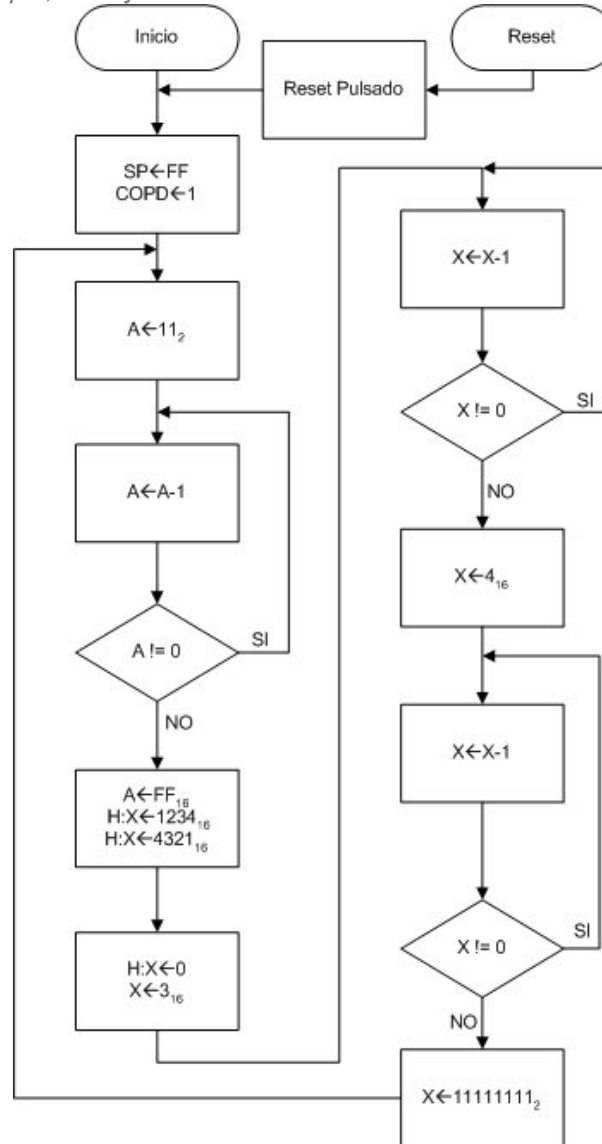
<i>Registros</i>	<i>Abreviatura</i>	<i>Bits</i>	<i>Usado comunmente para:</i>
Acumulador	A ó ACCA	8	Operaciones aritmeticas y lógicas.
Reg. Índice	H:X	16 (H=X=8)	Movimiento sobre la memoria del microcontrolador.
Puntero de Pila	SP	16	Contener la dirección del dato siguiente en una memoria apilada.
Contador de Programa	PC	16	Contener la dirección de la instrucción siguiente a procesarse.
Reg. de Banderas	CCR	8	Contiene seis (6) bits de estado, los cuales indican el resultado de operaciones, p.e. si se da un cero (0).

Para la siguiente sección, cargue y simule el archivo *NT0007 – CPU08 – 21 02 04.asm* siguiendo los pasos de la secciones 1.7.4 y 1.7.5.

### 1.7.4 Diagrama de Flujo

El siguiente programa ejemplifica el manejo de las instrucciones básicas y asignación.

*Rutina Principal, manejo de instrucciones básicas Reinicia el sistema*



*Figura 78. NT0007 – CPU08. (a) Rutina principal de manejo de instrucciones básicas. Saltos y asignaciones. (b) Vectores de Interrupción. Reinicia el sistema.*

## 1.7.5 Código

---

```
=====
;
; ARCHIVO      : NT0007 - CPU08 - 21 02 04
; PROPÓSITO   : Decrementar, Comparar y Saltar - Uso de DECA, DECX y DBNZX
;              1. FAMILIARIZARSE CON:
;                  - LA ARQUITECTURA INTERNA DEL MICRO
;                  - EVALUAR INSTRUCCIONES BASICAS
;
; PROCED.     :
;
; 1. COMPILAR EL ARCHIVO *.ASM
; 2. ACTICAR EL SIMULADOR - SIN LA TARJETA
; 3. CARGAR EL ARCHIVO *.S19 + RESET
; 4. EN EL MEMORY WINDOW 1 REVISAR
;   - MEMORIA $EC00 = INICIO DEL PROGRAMA
;   - MEMORIA $FFFE = VECTOR DE INICIO
; 5. EJECUTAR PASO A PASO
; 6. OBASERVAR EL REGISTRO "A", "H" Y "X"
;
; REFERENCIA:
;
; 1. ICS08 Operators Manual.PDF
; 2. ICS08 Addendum - Operators Manual.PDF
; 3. Technical Data of 68HC908JK3 - 9j13r1.PDF
; 4. Advanced Information of 68HC908JK3 - JI3r3.PDF
; 5. M68HC08 CPU Reference Manual - Cpu08r2.PDF
;
; LENGUAJE    : IN-LINE ASSEMBLER
;
;-----
; HISTORIAL
; DD MM AA
; 22 01 02   Creado.
; 25 08 04   Modificado.
;
;=====

;-----
;
;                               Definiciones del Usuario
;-----

COPD                equ $0          ; (3-31)

;-----
;
;                               Mapa de Memoria del Microcontrolador
;-----
;-----
;
;                               Registro de Configuraciones
;-----
CONFIG1             equ $001F       ; (3-31)
```

```

=====
;
;                               Memoria FLASH
;
=====
FLASH_START      equ $EC00          ; Puntero - Mem.FLASH

;
;                               Vectores de Usuario
;
=====
RESET_VEC        equ $FFFE          ; Puntero del RESET

;
;                               =====
; OBJETIVO   : Inicio de Codif. del Ensambla-
;                               : dor en la Memoria FLASH.
;                               :
;                               =====
;                               org FLASH_START          ; Inicio Mem. FLASH

;
;                               =====
; OBJETIVO   : Inicio del programa
;                               : SP = STACK POINTER (3-31)
;                               : COPD = WATCHDOG
;                               :
;                               =====

START
                rsp                ; inic.Stack = $00ff
                bset COPD,CONFIG1   ; desactiva watchdog

;
;                               =====
; PROPÓSITO : Ciclo interminable
;
;
; Objetivo   :
; 1. Inicializa A = $03
; 2. Decrementa A hasta 0 usando "DECA"
; 3. Carga el Registro H:X
; 4. Decrementa X hasta 0 usando "DEC"
; 5. Decrementa X hasta 0 usando "DBNZX"
; 6. Repite el ciclo
;
;
; Referencia: (5-26) para registros
;
;                               =====
; ETIQUETA OPERACION_OPERANDO      ; COMENTARIO ( REFERENCIA )

LOOP0          LDA #%00000011        ; A = %0000 0011 = 3
LOOP1          DECA                   ; A = A - 1 (5-185)
               CMP #0                 ; A = 0 ???
               BNE LOOP1              ; Subr. de RETARDO
               LDA #$FF               ; A = $FF = 255
               LDHX #$1234            ; H:X = $1234
               LDHX #$4321            ; H:X = $4321
               LDHX #0                ; H:X = $0000
               LDX #$03               ; X = $03
LOOP2          DECX                   ; X = X - 1 (5-185)
               CPX #0                 ; X = 0 ???
               BNE LOOP2              ; Subr. de RETARDO

```

NT0007

Rev. 1 del 06.08.05



```

LDX #$04 ; X = $04

LOOP3 DBNZX LOOP3 ; X = X - 1 ->LOOP3 (5-136/185)
LDX #%11111111 ; X = $FF
JMP LOOP0 ; REPITE EL CICLO

```

```

;=====
; OBJETIVO : Búsqueda del Vector de Reset
;           Arranque del programa en la memo-
;           ria FLASH
;=====
;===== Vector de Reinicio de Sistema =====
org RESET_VEC ; Puntero Vec - RESET
dw START ; al darse reset salta a Start

```

*Listado 2. NT0007 – CPU08. Exploración de las instrucciones condicionales de control de flujo (CMP, BNE), incondicionales de control de flujo (JMP), de postincremento (INC) y carga a registros (LDA, LDHX, LDX) y de decremento (DEC, DBNZ).*

*Nota: DECA, DECX decremantan en uno cada registro  
La nueva instrucción DBNZX decrementa y salta si no encuentra un resultado de cero.  
El prefijo % se utiliza para escribir un número binario.*

## 1.7.6 Conclusión

---

Se estudiaron aspectos básicos como, ¿qué es un microcontrolador?, la diferencia entre un microcontrolador y un microprocesador.

Se mencionó una breve analogía entre el CPU y el cerebro humano, además de vincular varios registros del CPU al uso habitual de este y finalmente se estudió un programa ejemplo sobre el aspecto de cómo interactúa el CPU para ejecutar acciones de control.

## 1.7.7 Referencias

---

### 1.7.7.1 Data Técnica sobre el microcontrolador JL3 y JK3

(a) [http://www.freescale.com/files/microcontrollers/doc/data\\_sheet/MC68HC08JL3.pdf](http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC08JL3.pdf)

*Pág. 25, Asignación de pines del microcontrolador.*

*Pág. 26, Descripción de la función de cada pin del microcontrolador.*

*Pág. 45, Sección del CPU del microcontrolador.*

*Pág. 46 a 52, Breve aclaratoria de los registros del microcontrolador.*

*Pág. 184, Especificaciones eléctricas a 5 V<sub>DC</sub>.*

### 1.7.7.2 Juego de Instrucciones en español del microcontrolador

(a) <http://bairesrobotics.com.ar/data/instrucc0508.pdf>

### 1.7.7.3 Guía didáctica en español de los microcontroladores de Motorola

(a) <http://bairesrobotics.com.ar/data/guia68hc08.pdf>

### 1.7.7.4 Cursos en línea y publicaciones para HC08 de Electrocomponentes

(a) <http://www.electrocomponentes.com.ar/index1.html>

### 1.7.7.5 Manual de Referencia del CPU

(a) [http://www.freescale.com/files/microcontrollers/doc/ref\\_manual/CPU08RM.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/CPU08RM.pdf)

*Pág. 98, Instrucción ADD.*

*Pág. 107, Instrucción BEQ.*

*Pág. 112, Instrucción BHI.*

*Pág. 118, Instrucción BLO.*

*Pág. 126-127, Instrucción BRA.*

*Pág. 143, Instrucción DBNZ.*

*Pág. 144, Instrucción DEC.*

### 1.7.7.6 Página “web” sobre esta Nota Técnica

(a) <http://www.geocities.com/issaiass/>

## 1.7.8 Problemas Propuestos

---

Para esta sección básiase en el listado 2 y la referencia 1.7.7.5

1.7.8.1 Saltar a LOOP1 seis (6) veces utilizando la instrucción BHI.

1.7.8.2 Modificar el lazo LOOP2 utilizando la instrucción BLO.

1.7.8.3 Reemplace el lazo LOOP3 y haga que esta sección sea un lazo de quince (15) saltos utilizando las instrucciones LDA, ADD, BEQ, BRA.