NT0008

1.8 PROGRAMACIÓN DEL MICROCONTROLADOR – REGISTROS, LOCALIDADES DE MEMORIA Y MAPA DE MEMORIA

MAPA DE MEMORIA DE LOS MICROCONTROLADORES JK3/JL3

Preparado por: Rangel Alvarado
Estudiante Graduando de Lic. en Ing. Electromecánica
Universidad Tecnológica de Panamá
Panamá, Panamá
"e-mail": issaiass@cwpanama.net
"web site": http://www.geocities.com/issaiass/

ÍNDICE

1.8.1 Introducción

Un mapa de memoria es una representación pictórica de las regiones internas (registros) en las que se divide en este caso, nuestro microcontrolador.

Los registros son partes más pequeñas del mapa de memoria (localidades de memoria) las cuales el usuario puede alterar para conseguir interacción con los módulos, escribir un programa, controlar dispositivos, etc.

Las localidades de memoria son aquellas que se pueden leer y/o escribir (*estas son variables*), y son capaces de poseer data de un (1) byte almacenada en ella.

En esta sección trataremos de explicar en detalle que consiste una mapa de memoria para cualquier sistema computacional; también de establecer la relación entre mapa de memoria, registros y bits para luego finalizar por un programa ejemplo que utiliza regiones del mapa de memoria para leer y escribir datos.

1.8.2 Mapa de Memoria

1.8.2.1 El Concepto de Mapa de Memoria

Como se habló anteriormente, el mapa de memoria es la representación de forma gráfica de cómo se divide internamente un sistema. Cada región representa un pedazo del microcontrolador, concéntrese en el mapa de memoria como si fuese un elevador al cual podemos ir al piso que queramos.

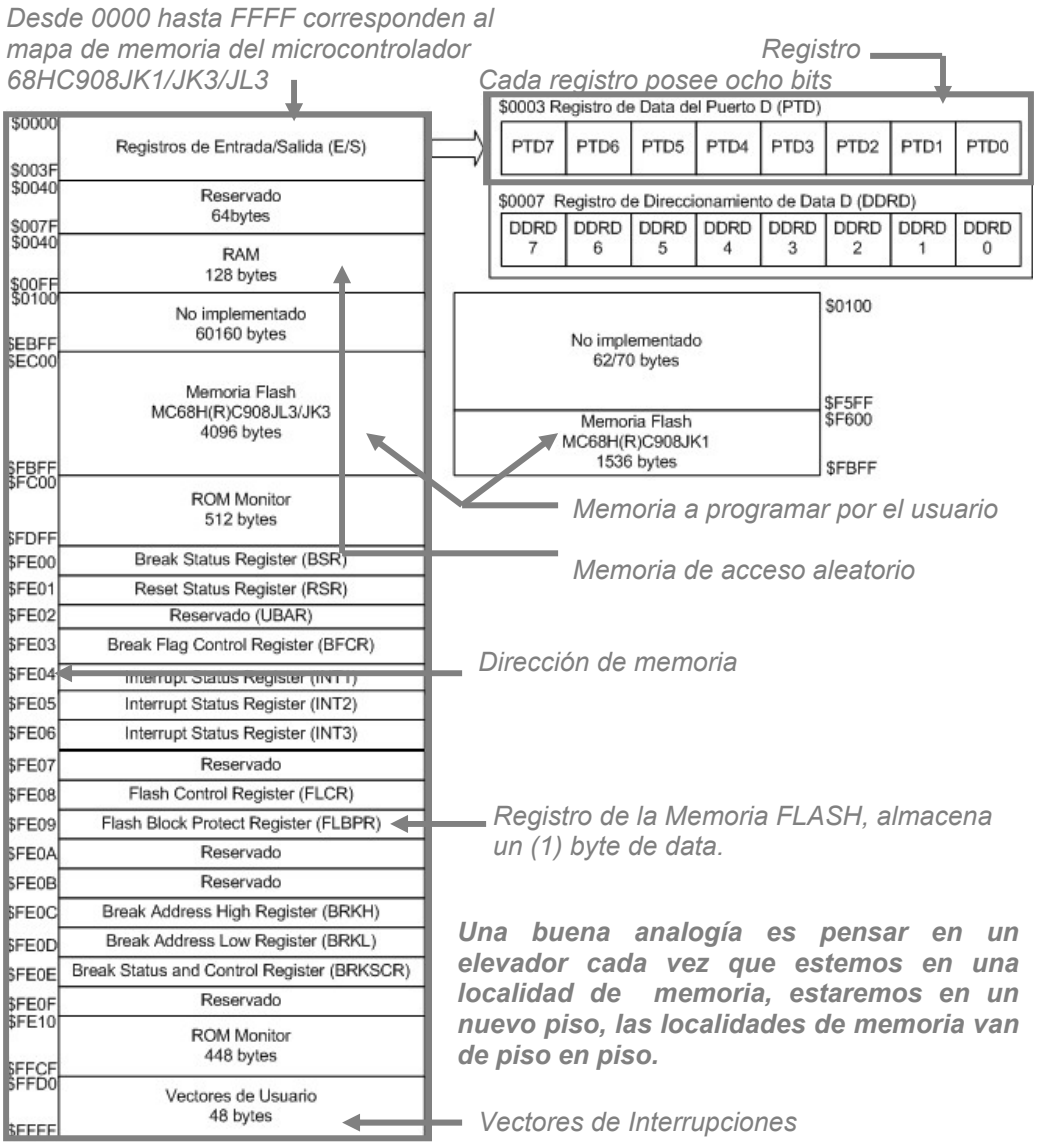

Figura 79. Mapa de Memoria de los microcontroladores JK1/JK3/JL3. En un mapa de memoria se puede encontrar de manera rápida el contenido de los dispositivos o periféricos de control, zonas no accesibles, memoria de programación, etc.

1.8.2.2 Concepto de Registro y Byte

Imagínese a un registro como una casa o "suite" que existe en ese piso, al entrar en esa "suite", hay ocho (8) habitaciones (8 bits) en las cuales puede o no estar la luz encendida (está encendida = 1, apagada = 0). La cantidad máxima de data la cual se puede almacenar en un registro consta de un (1) byte

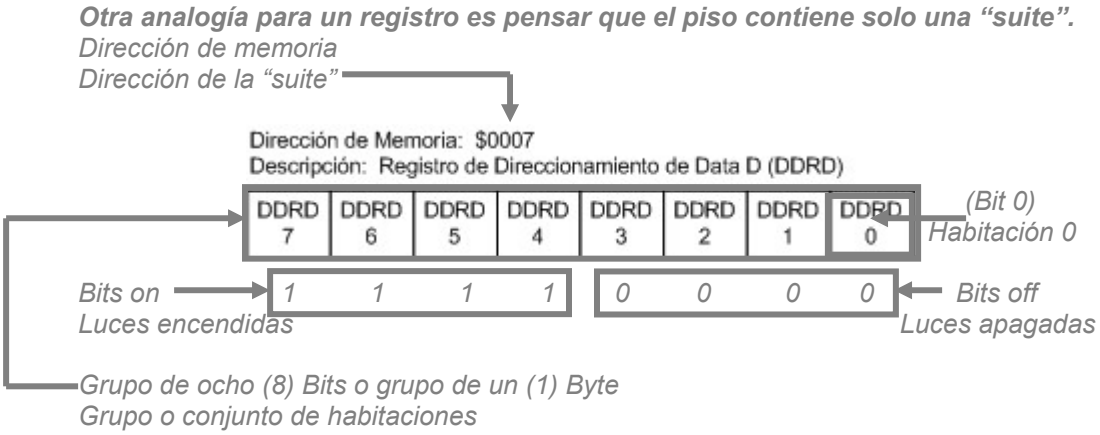

Figura 80. Concepto de Registro. Un registro es una localidad de memoria la cual posee una dirección y está dividida en grupos de ocho (8) bits, las cuales conforman un (1) byte y puede almacenarse algún dato o leer el dato almacenado en él.

1.8.2.3 Tipos de Memorias Existentes en el Microcontrolador HC908[1]

1.8.2.3.1 RAM – Memoria de Acceso Aleatorio

La RAM es un tipo de memoria volátil que puede ser leída o escrita por la CPU y se puede acceder a las posiciones de la RAM en cualquier orden. En un microcontrolador la memoria RAM es pequeña pues el área que ocupa en el chip es extensa.

Figura 81. Diagrama de Bloques de la Memoria RAM. La memoria RAM existente en un microcontrolador es pequeña pues ocupa mucho espacio en silicio, esta memoria es volátil, es decir, al desconectarse la alimentación pierde su contenido.

[1] Iniciación a los microcontroladores de 8-bits, Autor: Jordi Mayné.

1.8.2.3.2 ROM – Memoria de Solo Lectura

La ROM es la memoria más típica de memoria no volátil, el CPU puede leerla pero no se puede modificar. El contenido de la memoria lo proporciona el fabricante del dispositivo. Esta ROM trae incorporado el programa "Monitor" que es el que permite conectarse a la PC con un programa como WinIDE.

$1 + (FDFF_{16} – FC00_{16}) = 512$ $1 + (FFCF_{16} – FE10_{16}) = 448$

Figura 82. Diagrama de Bloques de la Memoria ROM. La memoria ROM es una memoria no volátil, de solo lectura y no modificable por el CPU. En especial, esta ROM trae rutinas incorporadas que son las que permiten conectarse al programa WinIDE o rutinas de programación de la memoria FLASH.

1.8.2.3.3 Memoria FLASH

Es una memoria que se puede borrar y programar eléctricamente por bloques. Esta en especial se llama FLASH "split-gate", es una memoria ultra rápida, eficiente y permite hasta 10000 ciclos de programado-borrado.

Flash para los MC JK3 Y JL3 (4kbytes) *Flash para el MC JK1 (1.5kbytes)*

Figura 83. Diagrama de Bloques de la Memoria FLASH. La memoria FLASH permite una alta velocidad, programación de una página o varios bytes y además asegura un ciclo confiable de programado-borrado de alrededor de 10000 ciclos.

1.8.2.3.4 Entrada/Salida (E/S) Como un Tipo de Memoria

El estado y el control de las Entradas/Salidas, es un tipo de posición de memoria que permite al sistema microcontrolador conseguir la información del mundo exterior. Este tipo de posición de memoria es inusual porque la información puede detectarse y/o puede cambiarse por otra cosa diferente a la CPU.

Figura 84. Diagrama de Bloques de un Dispositivo de Entrada/Salida. Un dispositivo de entrada salida común son los puertos, de estos se pueden adquirir o configurar para leer datos booleanos o para leer datos análogos como temperatura.

1.8.3 Diagrama de Flujo

El siguiente flujograma muestra el manejo de instrucciones de rotación de bits (ROR y ROL) además de borrado (CLC) y salto (BCC) dependiendo del estado del estado del bit de carry (C).

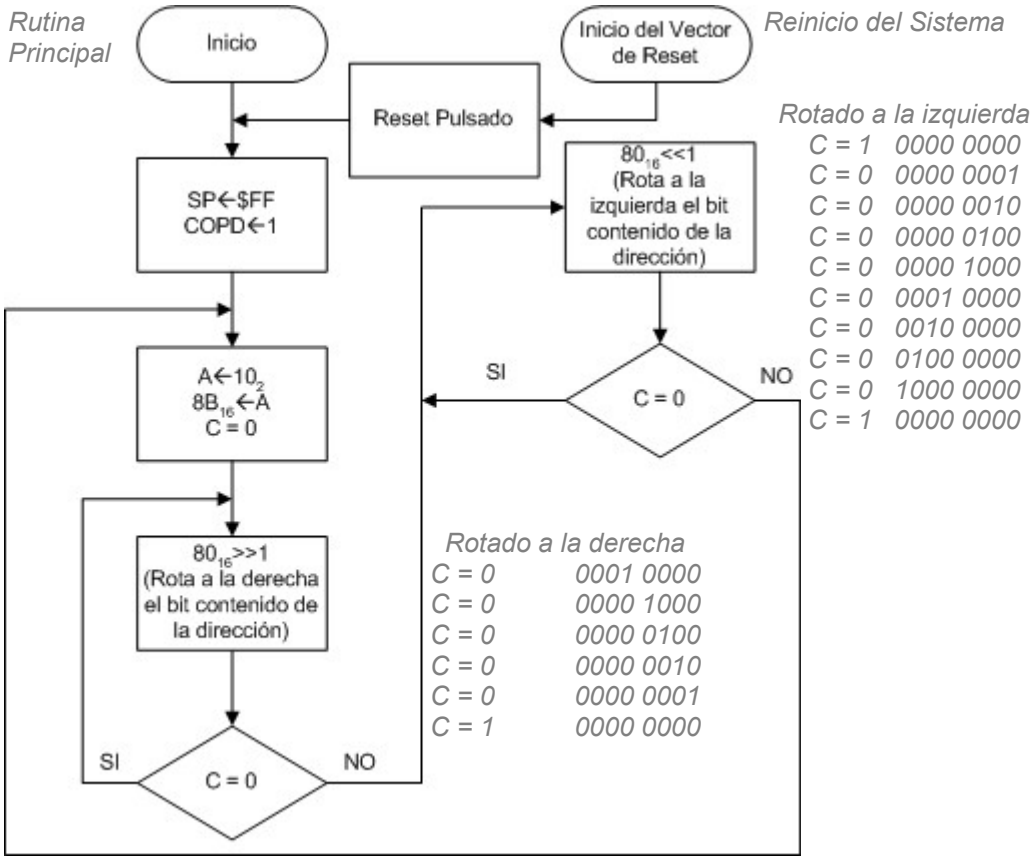

Figura 85. Diagrama de flujo de la NT0008 – Mapa de Memoria. En este programa se manejan instrucciones de salto dependiendo del estado del bit de llevando (C), manejo de variables en RAM además de borrado (CLC) y rotado (ROL, ROR) del bit de llevando .

1.8.4 Código

```
;*****************************************************************
; ARCHIVO      : NT0008 - Mapa de Memoria - 23 02 04
; PROPÓSITO  : Rotar a la Izquierda y a la Derecha a traves
;              del Carry (C) un Bit
;              1. FAMILIARIZARSE CON:
;              - LA ARQUITECTURA INTERNA DEL MICRO
;              - EVALUAR INSTRUCCIONES "ROL" Y "ROR"
;
; PROCED.      :
;
;  1. COMPILAR EL ARCHIVO *.ASM
;  2. ACTIVAR 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. AL EJECUTAR OBSERVAR LA DIRECCION $0080
;
; REFERENCIA:
;
;  1. ICS08 Operators Manual.PDF
;  2. ICS08 Addendum - Operators Manual.PDF
;  3. Technical Data of 68HC908JK3 - 9jl3r1.PDF
;  4. Advanced Information of 68HC908JK3 - Jl3r3.PDF
;  5. M68HC08 CPU Reference Manual - Cpu08r2.PDF
;
; LENGUAJE     : IN-LINE ASSEMBLER
; ----------------------------------------------------------------------
; HISTORIAL
;   DD MM AA
;   22   01   02   Creado.
;   26   08   04   Modificado.
;*****************************************************************
;*****************************************
;                  Definiciones del Usuario
;*****************************************
COPD            equ $0                  ; Bit 0 del registro de Configuración 1

;*****************************************
;            Mapa de Memoria del Microcontrolador
;*****************************************
;*****************************************
;                       Registro de E/S
;*****************************************
PORTD           equ $0003               ; PUERTO D
DDRD            equ $0007               ; REG.CONTROL DE DIR
```

```
;=============================================
;                  Registro de Configuraciones
;=============================================
CONFIG1     equ $001F           ; Registro de Configuración 1

;=============================================
;                       Memoria RAM
;=============================================
RAM_START   equ $0080           ; Puntero - Mem.RAM

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

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

;=============================================
; OBJETIVO    :  Inicio de Codif. del Ensam-
;                blador en 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
;=============================================
            LDA #%00010000      ; A =  %0001 0000
            STA $8B             ; A -> $008B
            CLC                 ; C = 1
LOOP1
            ROR $8B             ; ROTACION CIRCULAR DE
                                ; C Y $008B - A LA DERECHA
            BCC LOOP1           ; C = 0 ..? SI. CONTINUA ROTANDO
                                ; NO - C=1 - INVIERTE LA ROTACION
LOOP2
            ROL $8B             ; ROTACION CIRCULAR DE
                                ; C Y $008B - A LA IZQUIERDA
            BCC LOOP2           ; C = 0 ..? SI. CONTINUA ROTANDO
            JMP LOOP1           ; NO - C=1 - INVIERTE LA ROTACION
```

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

Listado 3. NT0008 – Mapa de Memoria. Se utilizan comandos básicos de salto BCC, JMP; manejo de bits como rotar a la derecha ROR y a la izquierda ROL para demostrar el manejo de variables en la RAM y limpiado del bit de llevando ("carry bit") y asignación STA.

Tabla 44. Instrucciones Utilizadas bajo el Programa de la NT0008

Instrucción	Complemento Ejemplo		Acción
sta	$8B	RAM+11	Almacena el byte de "ACCA"en una localidad.
clc			Borra el bit de carry en el CCR (C = 0).
bcc	LOOP0		Si C = 0, entonces salta a la etiqueta.
rol	$8B	RAM+11	Rota a la izquierda (ver fig. 85).
ror	$8B	RAM+11	Rota a la derecha (ver fig. 85).

Nota: (RAM = $80). RAM+11 simboliza que su dirección efectiva es $80 + $0B = $8B

1.8.5 Registros y Direcciones en el Simulador

En esta sección refiérase a la figura 86.
Inicie el simulador cargando el programa ejemplo NT0008 – Mapa de Memoria – 23 02 04.asm y concéntrese en la ventana de la memoria.

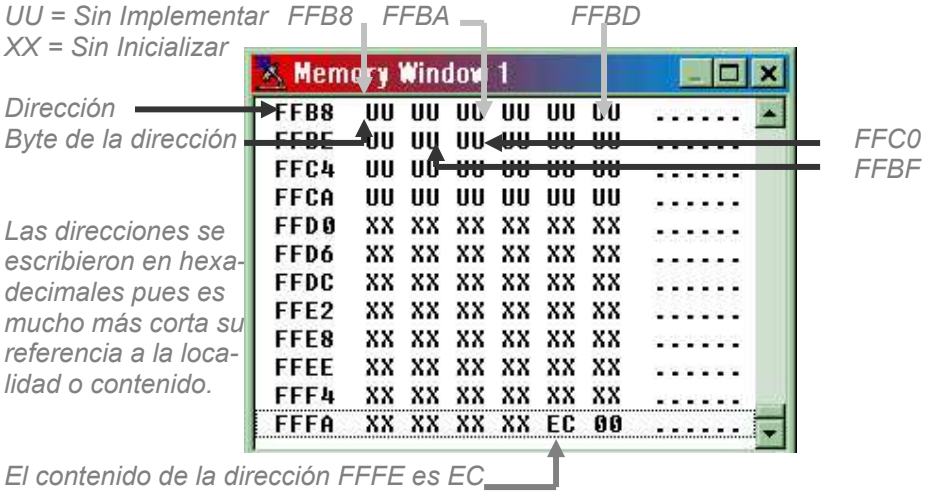

Figura 86. Ventana de Memoria del Simulador. Cada dirección de memoria está constituido por un número hexadecimal de dos (2) bytes de longitud en donde cada localidad puede acceder a un byte de capacidad.

1.8.6 Conclusión

El mapa de memoria fue estudiado a grandes rasgos para identificar las partes importantes del microcontrolador, cuyo "cuerpo" está dividido en registros y estos a su vez fueron diferenciados en zonas donde se puede leer o escribir una cantidad máxima de un (1) byte de data.

Por otro lado, se mencionan de las diferentes memorias que contiene un microcontrolador HC08, específicamente un MC JK1/JK3/JL3, y muy ligado a la nota técnica se explora un programa ejemplo que aplica nuevas instrucciones de rotado de bits, asignación en la memoria, limpiado de la bandera de llevando, etc.

Finalmente, se pudo ver en la pantalla del simulador, como se procede a vincular el mapa de memoria con los registros y localidades; como moverse a través del mapa por medio de direcciones efectivas y que contienen cada una de las direcciones.

1.8.7 Referencias

1.8.7.1 Información Avanzada sobre el Microcontrolador
(a) http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC08JL3.pdf

Pág. 28, se encuentra el mapa de memoria del microcontrolador
Págs. 30 a 35, todos los registros del microcontrolador incluyendo Vectores de Interrupción

1.8.7.2 Manual de Referencia del CPU
(a) http://www.freescale.com/files/microcontrollers/doc/ref_manual/CPU08RM.pdf

Pág. 104, Instrucción BCC
Pág. 134, Instrucción CLC
Pág. 167, Instrucción ROL
Pág. 168, Instrucción ROR
Pág. 175, Instrucción STA
Pág. 223, un glosario de palabras para mejor entendimiento del microcontrolador

1.8.7.3 Página "web" sobre esta Nota Técnica
(a) http://www.geocities.com/issaiass/

1.8.8 Problemas Propuestos

1.8.8.1 Realice un contador de 0 a 65535. Nota: Utilice dos variables en la RAM.

1.8.8.2 Realice un programa que cada vez que "PORTA" sea 1 incremente una variable en RAM. Utilice el listado 3.

1.8.8.3 Mediante un código propio, genere un programa que rote a la izquierda cada vez que se roten dos veces un bit a la derecha. Utilice el listado 3.