

SET de Instrucciones CPU08



WWW.EDUDEVICES.COM.AR



*Curso de Microcontroladores
Familia HC908 Flash de Freescale*

Parte II

ING. DANIEL DI LELLA



WWW.EDUDEVICES.COM.AR

SET DE INSTRUCCIONES

- **Movimiento de Datos**
- **Aritméticas**
- **Lógicas**
- **Manipulación de Datos**
- **Manipulación de Bits**
- **Control del Programa**
- **Operaciones BCD**
- **Especiales**

Compatibilidad de Código entre familias HC705 / HC908 / HC9S08 !!



Facilita la migración de aplicaciones a familias de MCUs más modernas sin perder horas de ingeniería en re - escritura de programas.

El set de instrucciones de la familia HC908, **es una versión muy mejorada y ampliada** del set de instrucciones de la flia. HC705. Esta característica hace que los usuarios de HC705 puedan migrar rápidamente a HC908, sin grandes cambios en sus rutinas y luego mejorar las aplicaciones existentes con el uso de las nuevas instrucciones.

Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II



ING. DANIEL DI LELLA



Movimiento de Datos

- Carga registros de CPU -

WWW.EDUDEVICES.COM.AR

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
LDA #opr LDA opr LDA opr LDA opr,X LDA opr,X LDA ,X LDA opr,SP LDA opr,SP	Load Accumulator from Memory	$A \leftarrow (M)$	0	-	-	3	3	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5
LDX #opr LDX opr LDX opr LDX opr,X LDX opr,X LDX ,X LDX opr,SP LDX opr,SP	Load Index Register X from Memory	$X \leftarrow (M)$	0	-	-	3	3	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5
LDHX #opr LDHX opr	Load Index Register H:X from Memory	$H:X \leftarrow (M:M + 1)$	0	-	-	3	3	-	IMM DIR	3 4



WWW.EDUDEVICES.COM.AR

Movimiento de Datos - Almacenamiento de registros del CPU -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles	
			V	H	I	N	Z	C			
STA opr STA opr STA opr,X STA opr,X STA ,X STA opr,SP STA opr,SP	Store Accumulator in Memory	M ← (A)							DIR	3	
									EXT	4	
					0	-	-	3	3	IX2	4
										IX1	3
										IX	2
										SP1	4
										SP2	5
STX opr STX opr STX opr,X STX opr,X STX ,X STX opr,SP STX opr,SP	Store Index Register X in Memory	M ← (X)							DIR	3	
									EXT	4	
					0	-	-	3	3	IX2	4
										IX1	3
										IX	2
										SP1	4
										SP2	5
STHX opr	Store Index Register H:X in Memory	M:M+1 ← (H:X)	0	-	-	3	3	-	DIR	4	



WWW.EDUDEVICES.COM.AR

Movimiento de Datos - Operaciones con el Stack -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
PSHA	Push Accumulator onto Stack	Push (A); $SP \leftarrow (SP - \$01)$	-	-	-	-	-	-	INH	2
PSHH	Push Index Register H onto Stack	Push (H); $SP \leftarrow (SP - \$01)$	-	-	-	-	-	-	INH	2
PSHX	Push Index Register X onto Stack	Push (X); $SP \leftarrow (SP - \$01)$	-	-	-	-	-	-	INH	2
PULA	Pull Accumulator from Stack	$SP \leftarrow (SP + \$01)$; Pull (A)	-	-	-	-	-	-	INH	2
PULH	Pull Index Register H from Stack	$SP \leftarrow (SP + \$01)$; Pull (H)	-	-	-	-	-	-	INH	2
PULX	Pull Index Register X from Stack	$SP \leftarrow (SP + \$01)$; Pull (X)	-	-	-	-	-	-	INH	2



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA



WWW.EDUDEVICES.COM.AR

Movimiento de Datos - Registro a Registro -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
TAP	Transfer Accumulator to CCR	$CCR \leftarrow (A)$	3	3	3	3	3	3	INH	2
TPA	Transfer CCR to Accumulator	$A \leftarrow (CCR)$	-	-	-	-	-	-	INH	1
TAX	Transfer Accumulator to Index Register X	$X \leftarrow (A)$	-	-	-	-	-	-	INH	1
TXA	Transfer Index Register X to Accumulator	$A \leftarrow (X)$	-	-	-	-	-	-	INH	1
TXS	Transfer Index Register to SP	$SPH:SP \leftarrow (H:X) - \0001	-	-	-	-	-	-	INH	2
TSX	Transfer SP to Index Register	$H:X \leftarrow (SPH:SP) + \0001	-	-	-	-	-	-	INH	2



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA



WWW.EDUDEVICES.COM.AR

Movimiento - Memoria a Memoria -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
MOV opr,opr MOV opr,X+ MOV #opr,opr MOV X+,opr	Move	$(M)Destination \leftarrow (M)Source$ $H:X \leftarrow (H:X) + 1$ in X+ modes	0	-	-	3	3	-	DD DIX+ IMD IX+D	5 4 4 4



Movimientos de Datos:

Las instrucciones de movimientos de datos, han sido ampliadas notoriamente en la familia HC908 en comparación con la HC705.

El agregado de instrucciones que involucran al nuevo registro concatenado H:X de 16 bits como ser LDHX, STHX, otorgan gran flexibilidad en el manejo de tablas y rutinas de acceso indexado, ahorrando código y aumentando la velocidad de ejecución de las mismas.

Además se puede apreciar que por cada tipo de instrucción, se agrega un nuevo modo de direccionamiento, basado en el uso del **Stack Pointer "SP"** (puntero de pila) como "segundo registro índice", lo que facilita el uso de lenguajes de alto nivel como el "C" y otros.

Las instrucciones PUSH y PULL permiten resguardar y rescatar el contenido del Acumulador (ACC) y del puntero índice H:X en espacio de RAM, ante sub-rutinas e interrupciones al programa (externas / internas), en forma más rápida y transparente.

Las instrucciones "MOV" en sus diferentes versiones, facilitan el movimiento de datos SIN AFECTAR LOS REGISTROS del CPU, de esta forma se consiguen operaciones más rápidas y algoritmos más sencillos. Estas instrucciones son útiles en rutinas de RX / TX en la SCI (UART) de los distintos MCUs de la flia. , o bién en movimientos de datos de una tabla a otra.





Aritméticas - Adición -

WWW.EDUDEVICES.COM.AR

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
ADD #opr ADD opr ADD opr ADD opr,X ADD opr,X ADD ,X ADD opr,SP ADD opr,SP	Add Memory to Accumulator without Carry	$A \leftarrow (A) + (M)$							IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5
ADC #opr ADC opr ADC opr ADC opr,X ADC opr,X ADC ,X ADC opr,SP ADC opr,SP	Add Memory to Accumulator with Carry	$A \leftarrow (A) + (M) + (C)$							IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA



Aritméticas - Sustracción -

WWW.EDUDEVICES.COM.AR

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
SUB #opr	Subtract Memory from Accumulator without Carry	$A \leftarrow (A) - (M)$							IMM	2
SUB opr									DIR	3
SUB opr									EXT	4
SUB opr,X			3	-	-	3	3	3	IX2	4
SUB opr,X									IX1	3
SUB ,X									IX	2
SUB opr,SP									SP1	4
SUB opr,SP									SP2	5
SBC #opr	Subtract Memory from Accumulator with Carry	$A \leftarrow (A) - (M) - (C)$							IMM	2
SBC opr									DIR	3
SBC opr									EXT	4
SBC opr,X			3	-	-	3	3	3	IX2	4
SBC opr,X									IX1	3
SBC ,X									IX	2
SBC opr,SP									SP1	4
SBC opr,SP									SP2	5



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA



WWW.EDUDEVICES.COM.AR

Aritméticas - Multiplicación & División -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
MUL	Unsigned 8-bit x 8-bit Multiply	$X:A \leftarrow (X) \times (A)$	-	0	-	-	-	0	INH	5
DIV	Unsigned 16-bit x 8-bit Divide	$A \leftarrow (H:A) \div (X)$ $H \leftarrow \text{Remainder}$	-	-	-	-	3	3	INH	7

MUL

- **X** contendrá el MSB del producto
- **A** contendrá el LSB del producto

DIV

- **H** es el MSB del dividendo
- **A** es LSB del dividendo
- **X** no es afectado



MULTIPLICACION Y DIVISION:

La familia HC908 contiene instrucciones de Multiplicación y de División, a diferencia de la familia HC705 que solo contenía la Multiplicación.

La instrucción de multiplicación en el CPU08, es del tipo No signado (sin signo) de 8 x 8 bits. Se ejecuta en 5 ciclos de Clock versus los 11 ciclos necesarios en HC705. En los registros "A" y "X" se cargan los valores a multiplicar, el resultado de la operación, se obtiene en los mismos registros "A" y "X", en "A" se encontrará la parte menos "pesada" del resultado, mientras que en "X" se encontrará la parte más pesada del resultado.

La instrucción División es nueva para aquellos usuarios de HC705, en el CPU08 es del tipo No signado (sin signo) de 16 / 8 bits. En el registro "H" se carga la parte más pesada del valor a dividir, en el registro "A" se carga la parte menos pesada de dicho valor, mientras el divisor se carga en el registro "X".

El resultado de la operación, se carga en el registro "A", mientras que el resto o remanente se carga en el "H". Si el resultado de la operación es mayor que "\$ FF", entonces se activará el flag de "CARRY" (C) en el CCR y el valor en el registro "H" será **indeterminado**.



WWW.EDUDEVICES.COM.AR





WWW.EDUDEVICES.COM.AR

Aritméticas - Incremento & Decremento -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
INC opr INCA INCX INC opr,X INC ,X INC opr,SP	Increment	$M \leftarrow (M) + 1$							DIR	4
		$A \leftarrow (A) + 1$							INH	1
		$X \leftarrow (X) + 1$	3	-	-	3	3	-	INH	1
		$M \leftarrow (M) + 1$							IX1	4
		$M \leftarrow (M) + 1$							IX	3
		$M \leftarrow (M) + 1$							SP1	5
DEC opr DECA DECX DEC opr,X DEC ,X DEC opr,SP	Decrement	$M \leftarrow (M) - 1$							DIR	4
		$A \leftarrow (A) - 1$							INH	1
		$X \leftarrow (X) - 1$	3	-	-	3	3	-	INH	1
		$M \leftarrow (M) - 1$							IX1	4
		$M \leftarrow (M) - 1$							IX	3
		$M \leftarrow (M) - 1$							SP1	5



Aritméticas

- Complemento & Negación -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
COM opr COMA COMX COM opr,X COM ,X COM opr,SP	Complement (One's Complement)	$M \leftarrow \$FF - (M)$							DIR	4
		$A \leftarrow \$FF - (A)$							INH	1
		$X \leftarrow \$FF - (X)$	0	-	-	3	3	1	INH	1
		$M \leftarrow \$FF - (M)$							IX1	4
		$M \leftarrow \$FF - (M)$							IX	3
		$M \leftarrow \$FF - (M)$							SP1	5
NEG opr NEGA NEGX NEG opr,X NEG ,X NEG opr,SP	Negate (Two's Complement)	$M \leftarrow \$00 - (M)$							DIR	4
		$A \leftarrow \$00 - (A)$							INH	1
		$X \leftarrow \$00 - (X)$	3	-	-	3	3	3	INH	1
		$M \leftarrow \$00 - (M)$							IX1	4
		$M \leftarrow \$00 - (M)$							IX	3
		$M \leftarrow \$00 - (M)$							SP1	5



www.edudevices.com.ar

Complemento a uno

- Operación no signada

Complemento a dos

- Operación signada



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA



Aritméticas - Comparación -

WWW.EDUDEVICES.COM.AR

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
CMP #opr CMP opr CMP opr CMP opr,X CMP opr,X CMP ,X CMP opr,SP CMP opr,SP	Compare Accumulator with Memory	A - (M)							IMM DIR EXT	2 3 4
			3	-	-	3	3	3	IX2 IX1	4 3
									IX	2
									SP1	4
									SP2	5
CPX #opr CPX opr CPX opr CPX opr,X CPX opr,X CPX ,X CPX opr,SP CPX opr,SP	Compare Index Register X with Memory	X - (M)							IMM DIR EXT	2 3 4
			3	-	-	3	3	3	IX2 IX1	4 3
									IX	2
									SP1	4
									SP2	5
CPHX #opr CPHX opr	Compare Index Register H:X with Memory	H:X - (M:M + 1)							IMM DIR	3 4
			3	-	-	3	3	3		



WWW.EDUDEVICES.COM.AR

Aritméticas - Miscelaneos

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
CLR opr CLRA CLR X CLR H CLR opr,X CLR ,X CLR opr,SP	Clear	M ← \$00 A ← \$00 X ← \$00 H ← \$00 M ← \$00 M ← \$00 M ← \$00	0	-	-	0	1	-	DIR INH INH INH IX1 IX SP1	3 1 1 1 3 2 4
TST opr TSTA TST X TST opr,X TST ,X TST opr,SP	Test for Negative or Zero	(M) - \$00 (A) - \$00 (X) - \$00 (M) - \$00 (M) - \$00 (M) - \$00	0	-	-	3	3	-	DIR INH INH IX1 IX SP1	3 1 1 3 2 4
AIS #opr	Add Immediate Value (Signed) to Stack Pointer	SPH:SP ← (SPH:SP) + (16 << M)	-	-	-	-	-	-	IMM	2
AIX #opr	Add Immediate Value (Signed) to Index Register H:X	H:X ← (H:X) + (16 << M)	-	-	-	-	-	-	IMM	2

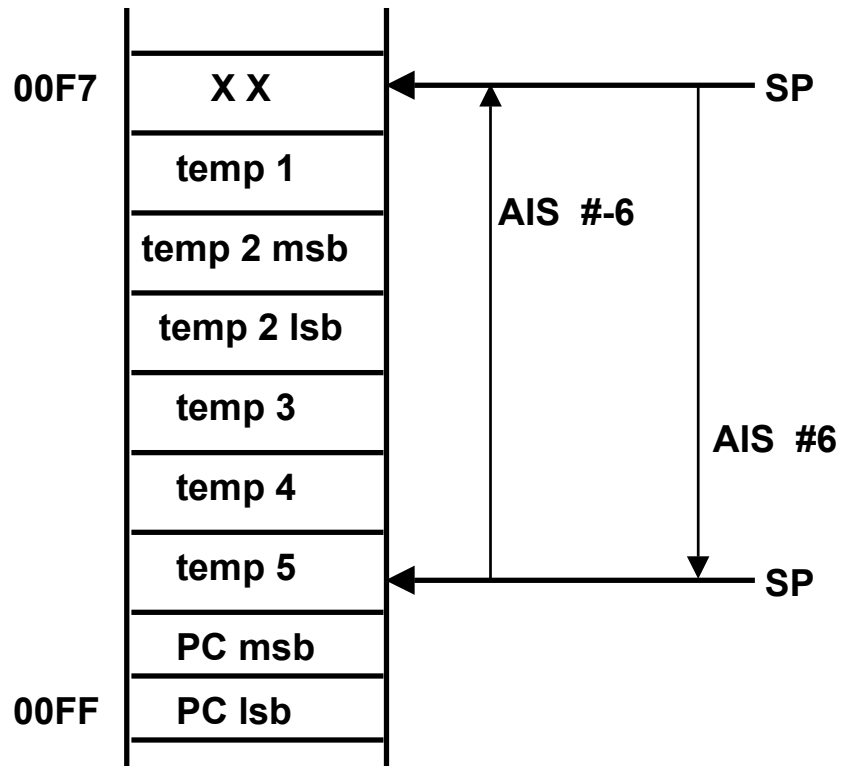
Mas sobre la instrucción AIS

AIS puede usarse para un rápido alojamiento o desalojo de espacio del Stack

- Variables Temporales
- Procesos en “trama”



WWW.EDUDEVICES.COM.AR



```
SUB1    AIS    #-6    ; Aloja 6 bytes
        .
        .
        AIS    #6     ; Desaloja 6 bytes
        RTS
```



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA

Mas sobre la instrucción AIX



WWW.EDUDEVICES.COM.AR

AIX puede usarse:

- Eficiente incremento o decremento del registro **H:X**
 - INCX / DECX solo afecta al registro X
 - INCX / DECX afecta el CCR, AIX No lo afecta
- Bucles (loops) alrededor de un bloque de memoria
 - Direccionamiento indexado con post incremento solo para incrementos
 - Solamente disponible para instrucciones MOV y CBEQ

La instrucción AIX permite “adicionar” en forma inmediata un número signado (positivo o negativo) al registro indice H:X, de esta forma pueden lograrse manejos de tablas más eficientes, búsquedas ascendentes o descendentes a partir de un punto, “saltos” discretos mayores a “1” en una tabla, tanto positivos como negativos. La instrucción AIX no afecta el CCR (registro de condiciones), y permiten incrementar / decrementar a registro H:X en forma amplia (16bits), y no reducida como las instrucciones INCX / DECX que solo afectan al registro “X”.





Ejemplo AIX

WWW.EDUDEVICES.COM.AR

```
* Calcula un 8 bit checksum para una tabla de 512 bytes
      ORG    $0100
TABLE RMB    512      ;Tabla de Datos

      ORG    $8000

      LDHX   #511     ;Inicialización del cdor de byte

      CLRA                   ;Inicializa el checksum

ADDLOOP ADD   Table,X     ;Calcula el checksum
        AIX   #-1        ;Decrementa el cdor de bytes
*Con DECX no hay "carry" desde X a través de H. AIX si !!!

        CPHX   #0        ;terminó ?
* CPHX setea bits CCR .

        BPL   ADDLOOP    ;en Loop si no se completó
```

Operadores Lógicos

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
AND #opr AND opr AND opr AND opr,X AND opr,X AND ,X AND opr,SP AND opr,SP	Logical AND Accumulator and Memroy	$A \leftarrow (A) \wedge (M)$	0	-	-	3	3	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5
ORA #opr ORA opr ORA opr ORA opr,X ORA opr,X ORA ,X ORA opr,SP ORA opr,SP	Inclusive OR Accumulator and Memory	$A \leftarrow (A) + (M)$	0	-	-	3	3	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5
EOR #opr EOR opr EOR opr EOR opr,X EOR opr,X EOR ,X EOR opr,SP EOR opr,SP	Exclusive OR Accumulator and Memory	$A \leftarrow (A) \oplus (M)$	0	-	-	3	3	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5



WWW.EDUDEVICES.COM.AR

Manipulación de Bits

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
BIT #opr BIT opr BIT opr BIT opr,X BIT opr,X BIT ,X BIT opr,SP BIT opr,SP	Bit test (AND) Accumulator with Memory	$A \wedge (M)$	0	-	-	3	3	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	2 3 4 4 3 2 4 5
BCLR n,opr	Clear bit n in Memory	$M_n \leftarrow 0$	-	-	-	-	-	-	DIR	4
BSET n,opr	Set bit n in Memory	$M_n \leftarrow 1$	-	-	-	-	-	-	DIR	4
CLC	Clear Carry Bit	$C \leftarrow 0$	-	-	-	-	-	0	INH	1
SEC	Set Carry Bit	$C \leftarrow 1$	-	-	-	-	-	1	INH	1
CLI	Clear Interrupt Mask	$I \leftarrow 0$	-	-	0	-	-	-	INH	2
SEI	Set Interrupt Mask	$I \leftarrow 1$	-	-	1	-	-	-	INH	2

Manipulación de Datos

- Shifts -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
ASL opr ASLA ASLX ASL opr,X ASL ,X ASL opr,SP	Arithmetic Shift Left (Same as LSL)		3	-	-	3	3	3	DIR INH INH IX1 IX SP1	4 1 1 4 3 5
ASR opr ASRA ASRX ASR opr,X ASR ,X ASR opr,SP	Arithmetic Shift Right		3	-	-	3	3	3	DIR INH INH IX1 IX SP1	4 1 1 4 3 5
LSL opr LSLA LSLX LSL opr,X LSL ,X LSL opr,SP	Logical Shift Left		3	-	-	3	3	3	DIR INH INH IX1 IX SP1	4 1 1 4 3 5
LSR opr LSRA LSRX LSR opr,X LSR ,X LSR opr,SP	Logical Shift Right		3	-	-	0	3	3	DIR INH INH IX1 IX SP1	4 1 1 4 3 5



WWW.EDUDEVICES.COM.AR

Manipulación de Datos - Rotates -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
ROL opr ROLA ROLX ROL opr,X ROL ,X ROL opr,SP	Rotate Left through Carry		3	-	-	3	3	3	DIR INH INH IX1 IX SP1	4 1 1 4 3 5
ROR opr RORA RORX ROR opr,X ROR ,X ROR opr,SP	Rotate Right through Carry		3	-	-	3	3	3	DIR INH INH IX1 IX SP1	4 1 1 4 3 5





WWW.EDUDEVICES.COM.AR

Control de Programas - Branches (saltos) -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
Bcc rel	Branch if condition is true (CC, CS, HCC, HCS, HI, HS, LO, LS, PL, MI, EQ, NE, GE, GT, LE, LT, IH, IL, MC, MS)	$PC \leftarrow (PC) + \$0002 + rel ? cc$	-	-	-	-	-	-	REL	3
BRA rel	Branch Always	$PC \leftarrow (PC) + \$0002 + rel$	-	-	-	-	-	-	REL	3
BRN rel	Branch Never	$PC \leftarrow (PC) + \$0002$	-	-	-	-	-	-	REL	3
BRCLR n,opr,rel	Branch if Bit n in Memory is Clear	$PC \leftarrow (PC) + \$0003 + rel$? Mn = 0	-	-	-	-	-	3	DIR/ REL	5
BRSET n,opr,rel	Branch if Bit n in Memory is Set	$PC \leftarrow (PC) + \$0003 + rel$? Mn = 1	-	-	-	-	-	3	DIR/ REL	5



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA

Control de Programa - Saltos Especiales -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
CBEQ opr,rel	Compare and Branch if Equal	PC ← (PC) + \$0003 + rel ? (A) - (M) = \$00							DIR	5
CBEQA #opr,rel		PC ← (PC) + \$0003 + rel ? (A) - (M) = \$00							IMM	4
CBEQX #opr,rel		PC ← (PC) + \$0003 + rel ? (X) - (M) = \$00							IMM	4
CBEQ X+,rel		PC ← (PC) + \$0003 + rel ? (A) - (M) = \$00	-	-	-	-	-	-	IX+	4
CBEQ opr,X+,rel		PC ← (PC) + \$0002 + rel ? (A) - (M) = \$00							IX1+	5
CBEQ opr,SP,rel		PC ← (PC) + \$0004 + rel ? (A) - (M) = \$00							SP1	6
DBNZ opr,rel	Decrement and Branch if not Zero	M ← (M) - \$01 PC ← (PC) + \$0003 + rel ? (M) ° 0							DIR	5
DBNZA rel		A ← (A) - \$01 PC ← (PC) + \$0002 + rel ? (A) ° 0							INH	3
DBNZX rel		X ← (X) - \$01 PC ← (PC) + \$0002 + rel ? (X) ° 0							INH	3
DBNZ X,rel		M ← (M) - \$01 PC ← (PC) + \$0002 + rel ? (M) ° 0	-	-	-	-	-	-	IX	4
DBNZ opr,X,rel		M ← (M) - \$01 PC ← (PC) + \$0003 + rel ? (M) ° 0							IX1	5
DBNZ opr,SP,rel		M ← (M) - \$01 PC ← (PC) + \$0004 + rel ? (M) ° 0							SP1	6

“CBEQ” y “DBNZ”



WWW.EDUDEVICES.COM.AR

CBEQ -- Compare and Branch if Equal

DBNZ -- Decrement and Branch if Not Zero

CBEQ combina las instrucciones **CMP** y **BEQ**

- Operaciones más rápidas de Búsqueda / acceso a tablas

DBNZ combina las instrucciones **DEC** y **BNE**

- Loop's más rápidos y eficientes

Con estas dos nuevas instrucciones, se consigue un manejo más sencillo y eficiente de operaciones repetitivas como “loops” , búsqueda / acceso a tablas.

Se siguen conservando las instrucciones “tradicionales” para mantener la compatibilidad con la flia. HC705.

A continuación, veremos algunos ejemplos que utilizan estas instrucciones, con notables ventajas a las instrucciones tradicionales.



Ejemplo instrucción CBEQ



WWW.EDUDEVICES.COM.AR

- * Subrutina que busca en un string el próximo carácter en blanco
- * y luego apunta el registro H:X al caracter inmediatamente seguido al blanco.
- * H:X se asume que siempre apunta a la localización del string.

String	ORG	\$00A0	
	RMB	50	;The character string
Search	ORG	\$8000	
	LDA	#\$20	;Load search character
Loop	CBEQ	X+,Out	;Match?
	BRA	Loop	;No match, do it again.

- *El post incremento de X ocurrirá independientemente por donde tome el Branch. Por otra parte cuando una coincidencia es encontrada H:X estará listo para apuntar próximo caracter. *

Out RTS

Ejemplo instrucción DBNZ

* Time delay routine

* Delay = $N \times (160.0 + 0.375) \mu\text{s}$ for an 8 MHz CPU clock

* For example, for delay = 10ms $N = 63$

```
N          EQU      63          ;Loop counter for 10 ms delay

Count      ORG      $50
           RMB      1          ;Loop counter

Delay      ORG      $6E00
Loop      LDA      #N          ;Set delay constant
           DBNZ    Count,Loop  ;Inner loop, Count starts at $00
           DBNZA   Loop
           RTS
```



WWW.EDUDEVICES.COM.AR



Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA

Control de Programa

- Jumps (saltos) y Subrutinas -



WWW.EDUDEVICES.COM.AR

JMP -- Salto "largo" (16 bits) (mapa de 64K)

JSR - Salto a sub rutina "largo" (16 bits) (mapa de 64K)

BSR - Salto a sub rutina "corto" (8 bits) (256 posiciones)

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
JMP opr JMP opr JMP opr,X JMP opr,X JMP ,X	Jump to location	PC ← Jump Address							DIR	2
									EXT	3
									IX2	4
									IX1	3
									IX	2
JSR opr JSR opr JSR opr,X JSR opr,X JSR ,X	Jump to subroutine	PC ← (PC) + n (n=1, 2 or 3) Push (PCL); SP ← (SP) - 1 Push (PCH); SP ← (SP) - 1 PC ← Unconditional Address							DIR	4
									EXT	5
									IX2	6
									IX1	5
									IX	4
BSR rel	Branch to subroutine	PC ← (PC) + 2 Push (PCL); SP ← (SP) - 1 Push (PCH); SP ← (SP) - 1 PC ← (PC) + rel							REL	4
RTS	Return from subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)							INH	4

Control de Programa

- Interrupciones -

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
SWI	Software Interrupt	$PC \leftarrow (PC) + 1$ Push (PCL); $SP \leftarrow (SP) - 1$ Push (PCH); $SP \leftarrow (SP) - 1$ Push (X); $SP \leftarrow (SP) - 1$ Push (A); $SP \leftarrow (SP) - 1$ Push (CCR); $SP \leftarrow (SP) - 1$ $I \leftarrow 1$ $PCH \leftarrow$ Interrupt Vector High Byte $PCL \leftarrow$ Interrupt Vector Low Byte	-	-	1	-	-	-	INH	9
RTI	Return from Interrupt	$SP \leftarrow (SP) + 1$; Pull (CCR) $SP \leftarrow (SP) + 1$; Pull (A) $SP \leftarrow (SP) + 1$; Pull (X) $SP \leftarrow (SP) + 1$; Pull (PCH) $SP \leftarrow (SP) + 1$; Pull (PCL)	3	3	3	3	3	3	INH	7





WWW.EDUDEVICES.COM.AR

Operaciones BCD

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
DAA	Decimal Adjust Accumulator	$(A)_{10}$	U	-	-	3	3	3	INH	2
NSA	Nibble Swap contents of Accumulator	$A \leftarrow (A[3:0]:A[7:4])$	-	-	-	-	-	-	INH	3



Instrucciones Especiales

Source Forms	Description	Operation	Effect on CCR						Address Modes	Bus Cycles
			V	H	I	N	Z	C		
RSP	Reset Stack Pointer	$SPL \leftarrow \$FF$	-	-	-	-	-	-	INH	1
NOP	No Operation	None	-	-	-	-	-	-	INH	1
STOP	Stop Processor and wait for interrupt	$I \leftarrow 0$ Stop Oscillator	-	-	0	-	-	-	INH	1
WAIT	Halt Processor and wait for interrupt	$I \leftarrow 0$	-	-	0	-	-	-	INH	1



WWW.EDUDEVICES.COM.AR

WAIT:

- El CPU08 **detiene** el procesamiento de instrucciones
- Espera por una interrupción para salir de ese estado
- No se detiene el oscilador, Timers, A/D u otro periférico.

STOP:

- El CPU08 **detiene** el procesamiento de instrucciones
- **Detiene el circuito del oscilador**
 - Pone al MCU en estado “low power”
- Espera por una interrupción para salir de ese estado

Fin Capítulo 5 !!

Curso de Microcontroladores

Familia HC908 Flash de Freescale

Parte II

ING. DANIEL DI LELLA