



ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES

I

**TEMA XIII
ARQUITECTURA Y PROGRAMACION DE
UN
PROCESADOR DE 16 BITS (I): MC68000**



TEMA 13. Arquitectura y programación de un procesador de 16 bits (I): MC68000

- 13.1 Introducción
- 13.2 El MC68000 visto por el programador
- 13.3 El programa visto por el procesador MC68000
- 13.4 Ejemplos de utilización de instrucciones
- 13.5 La pila del MC68000
- 13.6 Interrupciones y excepciones

13.1. INTRODUCCIÓN

MC68000

- Bus datos (16)	D0..D15	
- Bus direcciones (24)	A1..A23, A0 es interno apunta a los 2 bytes en memoria A0 a A23 ⇒ direcciones de 000000H a FFFFFFFH	
- Registro de datos (8) de 32 bits		
- Registros direcciones (8) de 32 bits		
- Operandos	Procesadores	Carga-almacenamiento
		Registro memoria
	MC68000 híbrido	Memoria - memoria
- Regular	Las instrucciones no presentan casos especiales	
- Gran ortogonalidad	Todas instrucciones mismos modos direccionamiento	

- **Modelo de memoria**

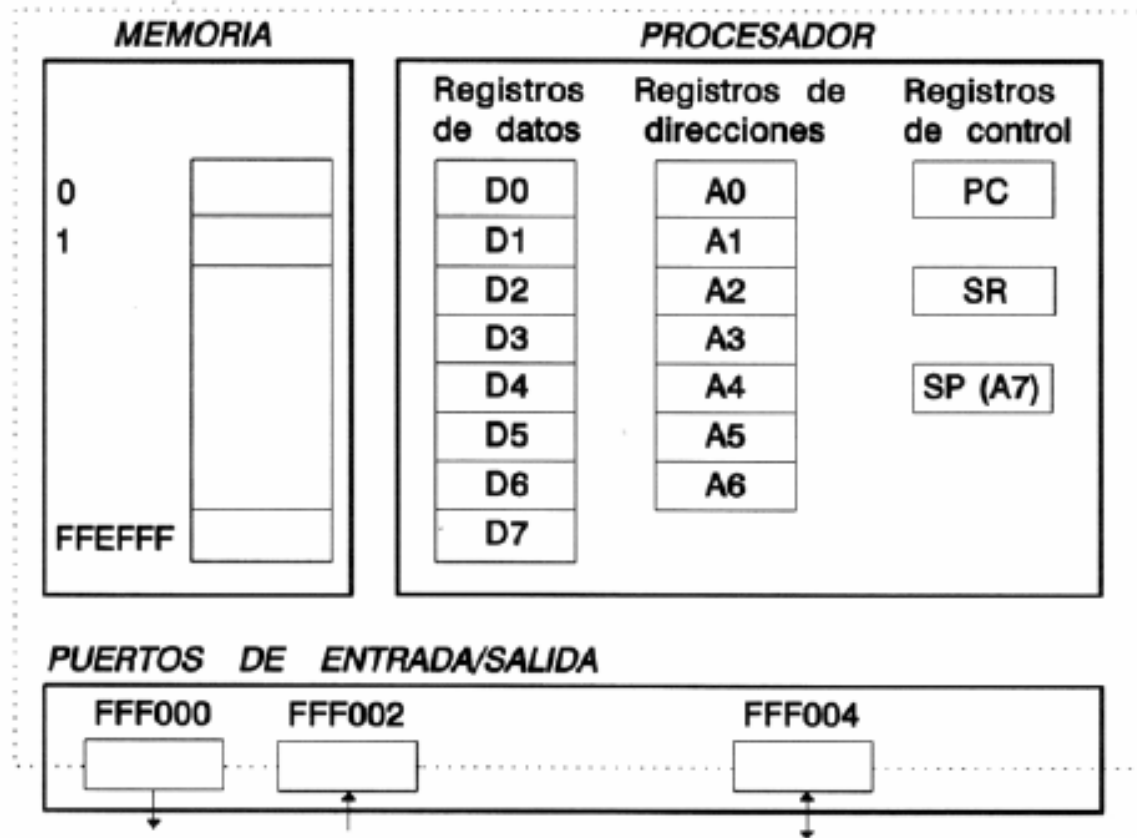
- Memoria principal estructurada en bytes (8 bits)
- Direcciones A0 a A23 (000000H a FFFFFFFH)
- Registros de direcciones de 32 bits A0 a A7 (A7 puntero de pila SP)

- **Modelo de los registros de datos**

- 8 registros de datos D0 a D7 de 32 bits
- Como algunas instrucciones permiten indicar la longitud de palabra B/W/L los registros se cargan con esa longitud de datos desde la parte menos significativa, dejando el resto inalterado.
- Notación para los datos

13.2. EL MC68000 VISTO POR EL PROGRAMADOR

- Al programador interesan
 - Los registros internos,
 - Los modos de direccionamiento,
 - Los tipos de instrucciones



Un modelo funcional del sistema computador basado en el microprocesador 68000 de Motorola.

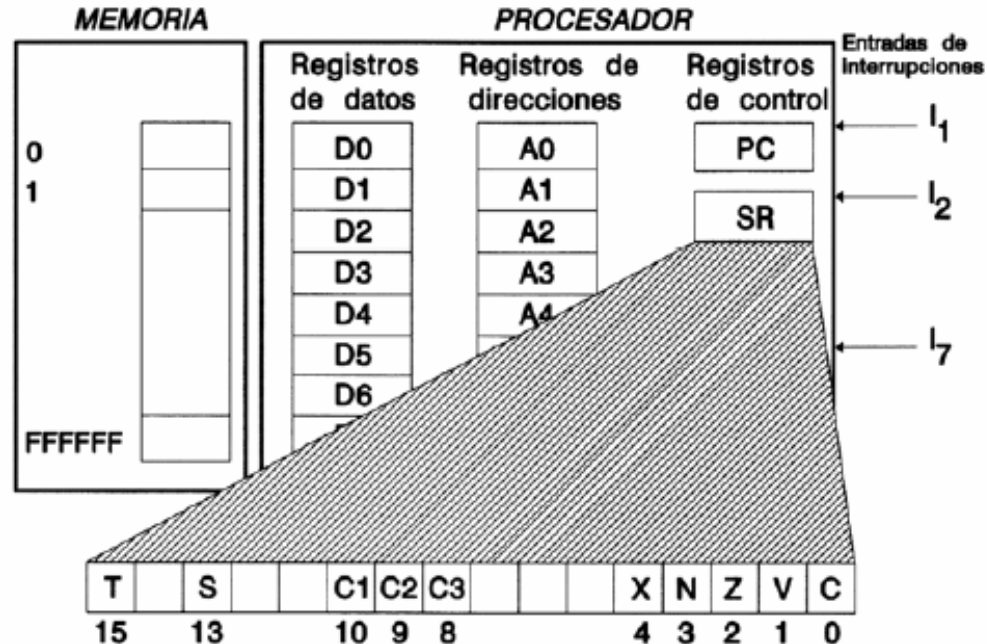
13.2.1. REGISTROS INTERNOS

- Un banco de registros formado por
 - Ocho registros de datos
 - Ocho registros de direcciones
 - Puntero de pila de usuario (USP)
 - Puntero de pila de supervisor (SSP)
 - Dispone de un registro contador de programa, PC, de 32 bits
 - En el MC68000 sólo salen al exterior 24 terminales del bus de direcciones.
 - Un registro de estado de 16 bits.

Registro de estado SR

- Un registro de 16 bits cuyos valores dependen del estado del procesador
- El byte más significativo contiene información del estado del sistema procesador.
 - El bit SR[13] indica si el procesador se encuentra o no en estado de supervisor.
- El byte menos significativo contiene información para el usuario, en concreto el denominado **registro de códigos de condición (CCR)**.

Modelo para instrucciones condicionales Gestión de excepciones del M68000



13.2.2. EL CONJUNTO DE INSTRUCCIONES

- La sintaxis general de las instrucciones en ensamblador es la siguiente:
 - *<mnemotécnico>. <tamaño> <origen>, <destino>*
- Un conjunto de instrucciones es **regular cuando cada instrucción no presenta casos especiales.**
- **Un conjunto** de instrucciones es **ortogonal cuando todas las instrucciones pueden utilizar los** mismos modos de direccionamiento.
- El conjunto de instrucciones del MC68000 puede dividirse en tres grandes grupos:
 - instrucciones de transferencia,
 - instrucciones de control e
 - instrucciones de cálculo o modificación de información

Accesos	B ⇒ Byte ⇒ 1 celda ⇒ dirección i
	W ⇒ Word (2 bytes) ⇒ 2 celdas ⇒ byte menos significativo en i / byte más significativo en i+1
	L ⇒ Long (4 bytes) ⇒ 4 celdas ⇒ byte menos significativo en i / byte más significativo en i+3

13.2.3. LOS MODOS DE DIRECCIONAMIENTO

- indirecto a registro (A_n),
- indirecto a registro con postincremento $(A_n)_+$,
- indirecto a registro con predecremento $-(A_n)$,
- indirecto a registro con desplazamiento $d16(A_n)$ o $(d16, A_n)$,
- indirecto a registro con índice $d8(A_n, R_x.t)$
- absoluto corto
- absoluto largo,
- relativo al contador de programa con desplazamiento $d16(PC)$,
- relativo al contador de programa con índice $d8(PC, R_x.t)$,
- inmediato $\#dato$,
- Implícito
 - el dato se encuentra en una posición que viene determinada por la propia instrucción.
 - El retorno de subrutina, RTS,

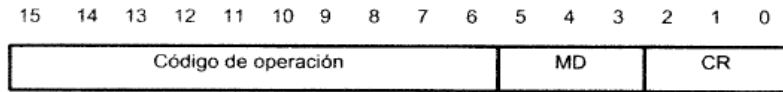
Modo de direccionamiento	Ejemplo	Notación
Inmediato	MOVE.B #1,D0	1 → D0
Absoluto	MOVE.B 1,D0	(1) → D0
Mediante registro de datos	MOVE.B D1,D0	(D1) → D0
Mediante registro de direcciones	MOVE.L A1,D0	(A1) → D0
Relativo a registro	MOVE.B (A1),D0	((A1)) → D0
Relativo a registro con posincremento	MOVE.B (A1)+,D0	((A1)) → D0; (A1) + 1 → A1
Relativo a registro con predecremento	MOVE.B -(A1),D0	(A1) - 1 → A1; ((A1)) → D0
Relativo a registro con desplazamiento	MOVE.B 5(A0),D0	((A0) + 5) → D0
Relativo a registro con índice	MOVE.B 5(A0,D1),D0	((A0) + (D1) + 5) → D0
Relativo a contador de programa con desplazamiento	MOVE.B 5(PC),D0	((PC) + 5) → D0
Relativo a contador de programa con índice	MOVE.B 5(PC,D1),D0	((PC) + (D1) + 5) → D0

MODO DE DIRECCIONAMIENTO INMEDIATO

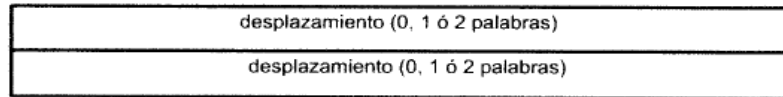
- El dato se encuentra en la propia instrucción. Además se utilizan las siguientes convenciones:
 - <número> o también
 - &<número> designa a ese número en base 10,
 - \$<número> designa a ese número en base 16,
 - @<número> designa a ese número en base 8,
 - %<número> designa a ese número en base 2.
- Ejemplos:
 - 11 = &11 = 11 en base decimal,
 - &11 = 9 en base decimal,
 - \$11 = 17 en base decimal,
 - %11 = 3 en base decimal.
- Un dato inmediato va precedido del símbolo #
 - **MOVE.L #\$FF, D0**

Decimal	Blanco o &
Binario	%
Hexadecimal	\$
Octal	@

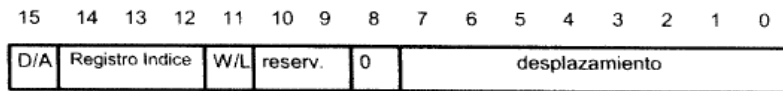
13.3. EL PROGRAMA VISTO POR EL PROCESADOR MC68000



Formato de dirección efectiva única (1 palabra)



Palabra(s) de ampliación tipo 1



Palabra de ampliación tipo 2

DESCRIPCIÓN

Código de operación: Instrucción y posible información sobre modo/registro para el segundo operando

CR: Campo de registro de datos o direcciones (ver Tabla13.3)

MD: Modo de direccionamiento (ver Tabla13.3)

desplazamiento: valor con signo de 8, 16 o 32 bits

D/A: Tipo de registro índice:
 0 registro de datos
 1 registro de direcciones

Registro índice: registro de datos o de direcciones (000 - 111)

W/L: Tamaño de registro índice:
 0 palabra
 1 palabra larga

13.3.1. FORMATOS DE LOS DATOS Y DE LAS INSTRUCCIONES. UBICACIÓN EN MEMORIA.

- Existen dos posibilidades:
 - Situar el byte de menor peso en la posición de memoria con menor dirección de memoria o
 - Situar el de mayor peso en dicha posición de memoria.
- A la primera posibilidad, conocida como "*little endian*",
 - La segunda se la conoce como "*big endian*" y se corresponde con la utilizada por el MC68000.
 - Supongamos que el dato hexadecimal \$F85A

dirección n	byte 1
dirección n +1	byte 2
dirección n +2	byte 3
dirección n +3	byte 4

Figura 13.2. Alineación en memoria de un dato de 32 bits.

<u>Dirección</u>	<u>Contenido</u>
1001	...
1002	\$F8
1003	\$5A
1004	...

- Supongamos que sea el dato \$E704FFA2 el que se encuentra en la posición 1002

1001	...
1002	\$E7
1003	\$04
1004	\$FF
1005	\$A2
1006	...

13.3.2. LOS FORMATOS DE LAS INSTRUCCIONES Y LOS MODOS DE DIRECCIONAMIENTO

- MOVE.W D1, D5.
- El código de operación de esta instrucción es 00xx, donde xx vale 11 para indicar el tamaño W (01 para el tamaño B y 10 para el tamaño L).
- A continuación le sigue el campo que especifica el destino: rrrmmm, donde rrr son los tres bits cuyo valor indican el registro y mmm son los tres bits que indican el modo de operación.
 - En este caso el registro destino es el registro de datos D5 por lo que el binario de 5 es 101, resultando ser rrr = 101.
 - El modo de direccionamiento directo en registro de datos se especifica por el código 000, resultando que es mmm = 000. Por lo tanto el campo destino se codifica por 101000.
- El campo origen se codifica de la misma manera con la única diferencia de ser ahora rrr = 001 (binario de 1).
- 0011101000001000

1000

1001

\$3A
\$08

MD	CR	Modo de direccionamiento efectivo
000	n° reg.	Mediante registro de datos
001	n° reg.	Mediante registro de dirección
010	n° reg.	Relativo a registro de dirección
011	n° reg.	Relativo a registro de dirección con posincremento
100	n° reg.	Relativo a registro de dirección con predecremento
101	n° reg.	Relativo a registro de dirección con desplazamiento
110	n° reg.	Relativo a registro de dirección con índice
111	000	Absoluto corto
111	001	Absoluto largo
111	010	Relativo a contador de programa con desplazamiento
111	011	Relativo a contador de programa con índice
111	100	Dato inmediato
111	101-111	Reservado

Tabla II.6. Codificación combinada de los campos modo/registro (MD/CR) en la *dirección efectiva* de un operando del M68000.

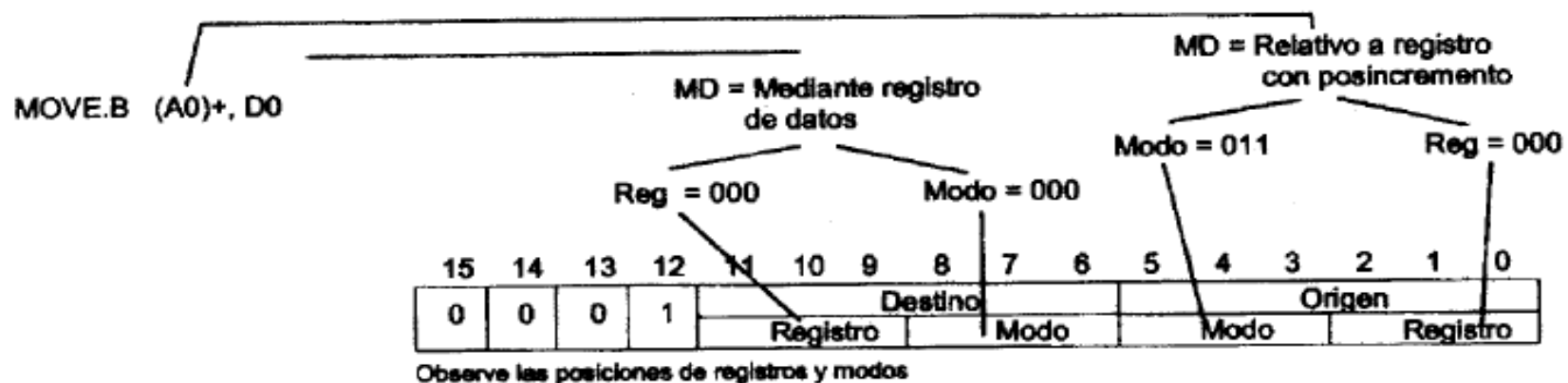


Figura 15.5 Tratamiento de la instrucción *MOVE.B* en la generación de código.

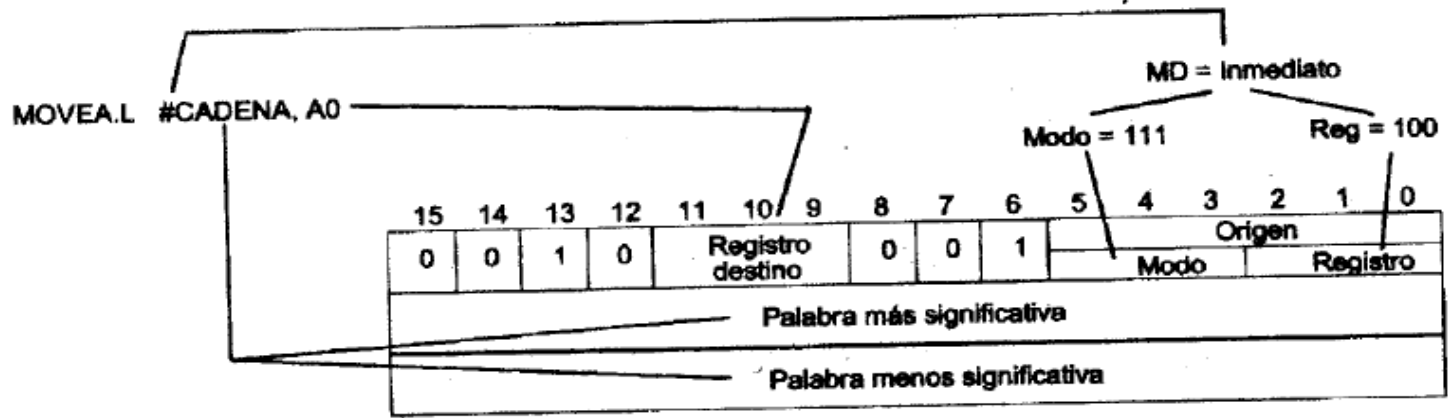


Figura 15.3 Tratamiento de la instrucción *MOVEA* en la generación de código.

total de 4 bytes.

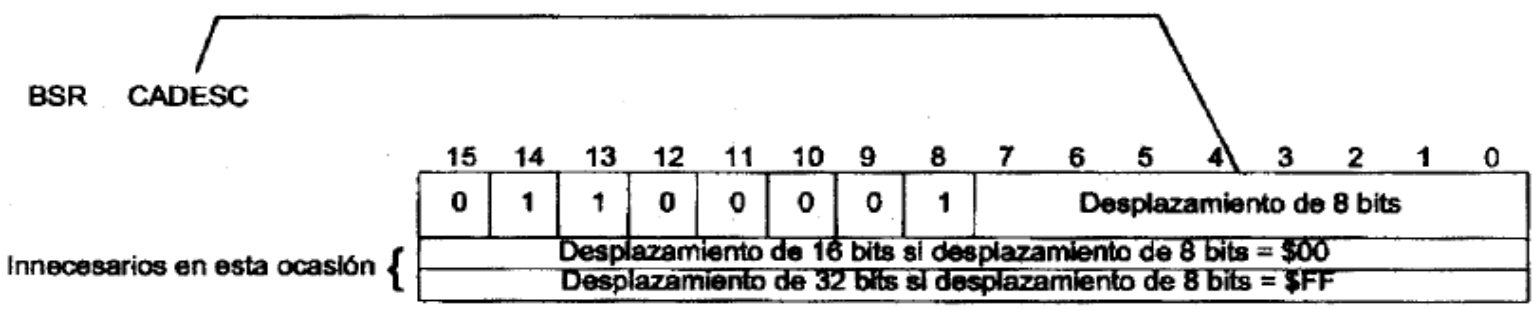


Figura 15.4 Tratamiento de la instrucción *BSR* en la generación de código.

13.3.3 LA EJECUCIÓN DE LAS INSTRUCCIONES. EL ARRANQUE DEL PROCESADOR. LOS VECTORES DE EXCEPCIÓN/INTERRUPCIÓN

- **Vectores de excepción**

- Consiste en una serie de posiciones de memoria que ante ciertas circunstancias (excepciones) se cargan en el contador de programa, dando lugar a un salto a una subrutina que se supone trata adecuadamente la situación que ha dado lugar a la excepción.

\$0000	USS[31:24]	Dirección o puntero de pila de sistema a cargar en el registro SSP del procesador.
\$0001	USS[23:16]	
\$0002	USS[15:8]	
\$0003	USS[7:0]	
\$0004	PC[31:24]	Dirección de inicio del programa a cargar en el registro contador de programa, PC.
\$0005	PC[23:16]	
\$0006	PC[15:8]	
\$0007	PC[7:0]	
...	...	

Figura 13.16. Contenidos de los primeros 8 bytes de memoria.

Ejemplos de instrucciones

D0 = \$00.00.43.21	
D1 = \$12.34.56.78	
D2 = \$87.65.43.21	
D3 = \$00.FF.00.FF	
D4 = \$00.00.00.00	
D5 = \$11.11.11.11	
D6 = \$10.10.10.10	
D7 = \$01.01.01.01	
A0 = \$00.00.40.00	
A1 = \$00.00.40.0F	
...	
A7 = \$00.00.4F.00	

FIGURA 13.19.

Símbolo	Dirección	Contenido
\$CERO	\$4000	\$00
\$UNO	\$4001	\$01
DOS	\$4002	\$02
TRES	\$4003	\$03
CUATRO	\$4004	\$00
	\$4005	\$04
CINCO	\$4006	\$00
	\$4007	\$05
SEIS	\$4008	\$00
	\$4009	\$06
SIETE	\$400A	\$00
	\$400B	\$07
OCHO	\$400C	\$00
	\$400D	\$00
	\$400E	\$00
	\$400F	\$08

FIGURA 13.20.

EJEMPLOS DE UTILIZACIÓN DE INSTRUCCIONES DE TRANSFERENCIA

Ejemplo 14.1.2.

Antes de la ejecución:	D0 = 00.00.43.21
Instrucción a ejecutar:	MOVE.W DOS, D0
Resultado:	D0 = 00.00.02.03

Figura 14.5.

Ejemplo 14.1.3.

Antes de la ejecución:	D0 = 00.00.43.21
Instrucción a ejecutar:	MOVE.L DOS, D0
Resultado:	D0 =02.03.00.04

Ejemplo 14.1.4.

Antes de la ejecución:	D1 = \$12.34.56.78
Instrucción a ejecutar:	MOVE.L #UNO, D1
Resultado:	D1 = 00.00.40.00

Figura 14.7.

Ejemplo 14.1.5.

Antes de la ejecución:	D1 = \$12.34.56.78
Instrucción a ejecutar:	MOVE.L CERO+2, D1
Resultado:	D1 = 02.03.00.04

EJEMPLOS DE UTILIZACIÓN DE INSTRUCCIONES ARITMÉTICAS.

Ejemplo 14.2.1.

Antes de la ejecución:	D0 = \$00.00.43.21
	D1 = \$12.34.56.78
Instrucción a ejecutar:	ADD.L D0, D1
Resultado:	D0 = \$00.00.43.21
	D1 = \$12.34.99.99

Figura 14.9.

Ejemplo 14.2.2.

Antes de la ejecución:	D0 = \$00.00.43.21
	D1 = \$12.34.56.78
Instrucción a ejecutar:	ADD.B D0, D1
Resultado:	D0 = \$00.00.43.21
	D1 = \$12.34.56.99

Ejemplo 14.2.3.

Antes de la ejecución:	D1 = \$12.34.56.78
Instrucción a ejecutar:	ADD.W D1, D1
Resultado:	D1 = \$12.34.AC.F0

Figura 14.11.

Ejemplo 14.2.4.

Antes de la ejecución:	D1 = \$12.34.56.78
Instrucción a ejecutar:	ADD.B D1, D1
Resultado:	D1 = \$12.34.56.F0

Figura 14.12.

EJEMPLOS DE UTILIZACIÓN DE INSTRUCCIONES LÓGICAS

Antes de la ejecución:	D1 = \$12.34.56.78
Instrucción a ejecutar:	ADD.L D1, D1
Resultado:	D1 = \$24.68.AC.F0

Figura 14.13.

Ejemplo 14.3.2.

Antes de la ejecución:	D0 = \$00.00.43.21
	D1 = \$12.34.56.78
Instrucción a ejecutar:	AND.W D0, D1
Resultado:	D0 = \$00.00.43.21
	D1 = \$12.34.42.20

Figura 14.15

Ejemplo 14.3.3.

Antes de la ejecución:	D0 = \$00.00.43.21
	D1 = \$12.34.56.78
Instrucción a ejecutar:	ORL D0, D1
Resultado:	D0 = \$00.00.43.21
	D1 = \$12.34.57.79

Figura 14.16

Ejemplo 3.3.4.

Antes de la ejecución:	D6 = \$10.10.10.10
	D2 = \$87.65.43.21
Instrucción a ejecutar:	OR.L D6, D2
Resultado:	D6 = \$10.10.10.10
	D2 = \$97.75.53.31

Figura 14.17

Ejemplo 3.3.5.

Antes de la ejecución: D6 = \$10.10.10.10

D2 = \$87.65.43.21

Instrucción a ejecutar: EOR.L D6, D2

Resultado:

D6 = \$10.10.10.10

D2 = \$97.75.53.31

Figura 14.18

Ejemplo 14.3.6.

Antes de la ejecución: D1 = \$12.34.56.78

D2 = \$87.65.43.21

Instrucción a ejecutar: EOR.W D1, D2

Resultado:

D1 = \$12.34.56.78

D2 = \$87.65.15.59

Figura 14.19

Ejemplo 14.3.7.

Antes de la ejecución:	D1 = \$12.34.56.78
Instrucción a ejecutar:	EOR.W D1, D1
Resultado:	D1 = \$12.34.00.00

Figura 14.20

Ejemplo 14.3.8.

Antes de la ejecución:	D3 = \$00.FF.00.FF
	D2 = \$87.65.43.21
Instrucción a ejecutar:	EOR.L D3, D2
Resultado:	D3 = \$00.FF.00.FF
	D2 = \$87.9A.43.DE

Figura 14.21.

13.4. LA PILA DEL MC68000

- La pila es un conjunto de posiciones de memoria de tamaño variable que se encuentra en memoria.
- Suele emplearse para recibir información antes de producirse un salto a subrutina para restituirla cuando ha finalizado la ejecución de dicha subrutina



Elementos: Cabeza de la pila \Rightarrow Registro puntero de pila (A7)
Memoria

Instrucciones: Meter \Rightarrow MOVE.x Dn,-(A7)
Sacar \Rightarrow MOVE.x (A7)+,Dn

Interrupciones y excepciones

Excepción = suspensión de la ejecución secuencial de las instrucciones en curso.

Respuesta a interrupción:

1. Termina instrucción en curso
2. Guarda parámetros en pila, PC a pila.
3. Dirección de vector interrupción → PC
4. En la subrutina salvar en pila los registros que se puedan modificar.
5. Terminación de ejecución de subrutina.
6. Recuperación del PC desde la pila para continuar con ejecución interrumpida.

Tipos	Interno	Errores de direccionamiento
		Instrucciones desconocidas
		Instrucciones no permitidas
	Externo	RESET
		Error de BUS
		Interrupciones (periféricos)