

ARQUITECTURA Y PROGRAMACIÓN DEL MC68000

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
			Memoria - memoria
		MC68000 híbrido	Registro + 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 (000000_H a FFFFFFF_H)

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

- 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:

Decimal	Blanco o &
Binario	%
Hexadecimal	\$
Octal	@

Modelo de direccionamiento

- Inmediato
- Absoluto
- Mediante registro
- Relativo a registro:
 - Normal
 - Con posincremento / preincremento
 - Con posdecremento / predecremento
 - Con desplazamiento
 - Con índice
 - Relativo a PC con desplazamiento
 - Relativo a PC con índice

Modelo para instrucciones condicionales

Bits de condición de flags	C Carry
	X eXtended
	Z Zero
	N Negativo
	V oVerflow

Set de instrucciones

Modelo para la gestión de subrutinas

Subrutina= conjunto de instrucciones que realiza una tarea concreta y que debe ser llamado desde el programa principal

Subrutina debe guardar el entorno de trabajo y el PC antes de saltar a ser ejecutada, para posteriormente poder recuperarlos y continuar donde estaba antes del salto.

PILA: Porción de mem. principal que se reserva para almacenar el entorno y que tiene un modo de acceso especial, ya que utiliza el puntero de pila (SP) A7 para direccionarla, no teniendo el programador que proporcionar la dirección de almacenamiento.

Definir la pila \Rightarrow al principio del programa indicar la posición inicial y el tamaño.

Operaciones en la pila:

Meter datos (PUSH)

Sacar datos (POP)

Con estas operaciones, automáticamente se incrementa/decrementa el puntero de pila SP



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

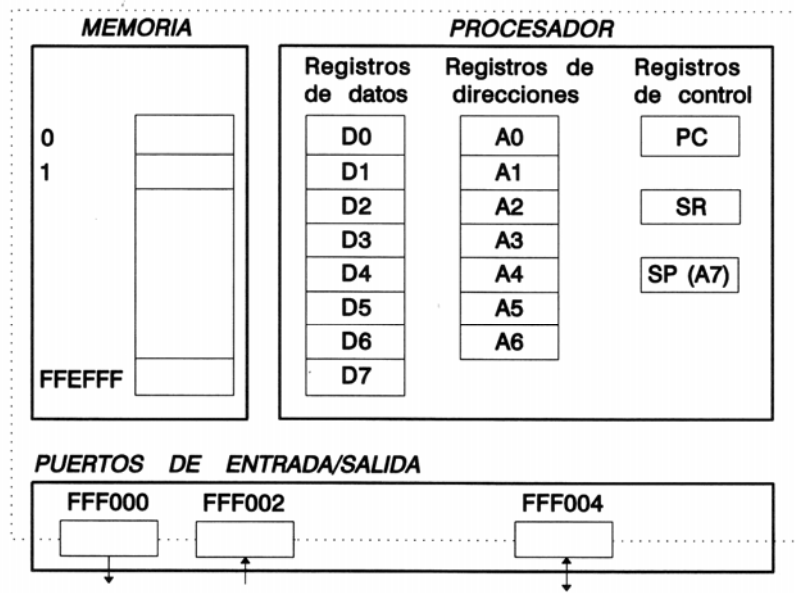
Excepción = suspensión de la ejecución secuencial de las instrucciones en curso.

Respuesta a interrupción:

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

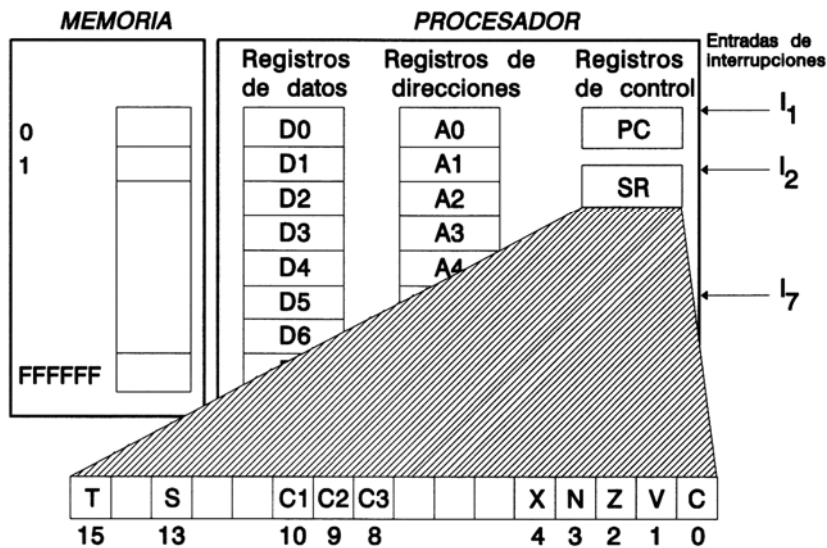
Tipos	Interno	Errores de direccionamiento
		Instrucciones desconocidas
		Instrucciones no permitidas
	Externo	RESET
		Error de BUS
		Interrupciones (periféricos)

MODELO FUNCIONAL DEL MICROPROCESADOR MC68000



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

Modelo para instrucciones condicionales Gestión de excepciones del M68000



ENSAMBLADOR PARA EL MC68000

Sintaxis [<etiqueta>] [<mnemonico_instruc> [<operandos>]] ; [<comentarios>]

Directivas:

ORG	<direcc o expresión>		origen absoluto de las instr. o datos	
END			fin	
EQU	<etiqueta>	EQU	<valor o expresión>	asigna a etiqueta el valor indicado
DS	<etiqueta>	DS.x	<número de variables>	reserva posición de memoria
DC	<etiqueta>	DC.x	<valor o valores>	reserva espacio y mete los valores indicados

Todos los programas deben ser escritos con un editor que no introduzca caracteres de control. Todos los programas han de **contener** la directiva **"ORG"** al principio del programa y la directiva **"END"** al final de éste.

```

00000000 =00000004    1 Longitud    EQU    4           ;longitud tiene valor de 4
00000000 =00000019    2 valor1     EQU    25          ;valor1= 25 osea $19
00000000 =000000A9    3 valor2     EQU    $A9         ;valor2 = $A9
00000000              4
00000000              5 ;aquí empieza el programa principal
00000000              6
00000400              7          ORG    $400       ;el programa empieza a cargarse en la dir $400
00000400              8
00000400 4E71         9          NOP                    ;instrucción que no hace nada
00000402             10
00000402             11                    ;zona de subrutinas
00000402             12
00000C00             13          ORG    $C00
00000C00 4E71         14          NOP
00000C02             15
00000C02             16 ;zona de datos
00000C02             17
00001000             18          ORG    $1000
00001000             19
00001000             20 vector     DS.W   Longitud    ;a "vector" se la asigna la dirección en la
00001008             21                    ;que está colocado y ($1000) además a partir de
00001008             22                    ;dicha dirección reserva "Longitud"(4) Words (16 bits) de la
00001008             23                    ;$1000 a la $1007
00001008             24
00001008 19 A9       25 cont1     DC.B   valor1,valor2 ;se guarda en las posiciones siguientes $1008
0000100A             26                    ;valor1 ($19) y en la $1009 valor2($A9)
0000100A             27
0000100A 00000019 000000A9 28 cont2     DC.L   valor1,valor2 ;igual que la anterior pero en lugar de guardar bytes
00001012             29                    ;se guardan palabras largas "L" esto es en la $100A $00
00001012             30                    ;$100B $00; $100C $00 ;$100D $19;$100E $00; $100F $00
00001012             31                    ;$1010 $00; $1011 $A9
00001012             32
00001012             33
00001012             34          END

```

No errors detected
No warnings generated

Sep 2002 - 19 - 20 B

(D0) = 0000020A
 (D1) = 0000FFFF
 (D2) = 35829EFA
 (D3) = 43503889
 (D4) = 00000000
 (D5) = FFFFFFFF
 (D6) = FFFFFFFF
 (D7) = 00000000

(A0) = 00020000
 (A1) = (A2) = (A3) = (A5) = (A6) = 00000000
 (A4) = 00020004
 (A7) = 00000FF0
 (020000) = 3C
 (1) = 4A
 (2) = 05
 (3) = 13
 (4) = 10

(020005) = A5
 6 = BF
 7 = 38
 8 = FF
 9 = 40

(19) MOVE.L (A4)+, D3; (D3) = 10 A5 BF 38 (A4) = 00030008
 MOVE.W -(A4), D0; (A4) = 00030006 (D0) = 0000BF38
 ADD.B D3, D0; (D0) = 0000BF70 ⇒ a

(20) MOVE.L #0000020A, D0
 MOVE.L A1, D1; (D1) = 00000000
 NOT.L D1; (D1) = FFFFFFFF
 MOVE.L A0, D0; (D0) = 00020000
 EOR.L D1, D0; (D0) = FFFDFFFF
 NOT.L D0; (D0) = 00020000
 MOVE.W D0, A0; (A0) = 00020000
 MOVE.W (A0)+, D6; (D6) = FFFF3C4A ⇒ d

Feb 2002 - 15 - 16.C

(D0) = \$0000020A SUBJ.B #0E, D0; (D0) = 000002FC ⇒ b

¡OJO!

$$\begin{array}{r} 0A \\ 0E \\ \hline FC \end{array}$$

↓
 SUBJ.B #n, a2 ⇒ (a2) - n → a2

Febrero 2000 - 1^{er} S - Tipo D

ORG \$1000
MAXIMO EQU \$5
BUSCAR EQU 'a' ; "a" = \$61

CADENA = \$1032 → c = \$63
CONTADOR = \$1038 a = \$61
 s = \$73
 * = \$2A

```

CLR    CONTADOR    ; (1038) = 0000
LEA    CADENA, A6   ; (A6) = 1032
MOVE.L #MAXIMO, D6 ; (D6) = 00000005

CUENTA MOVE.B (A6)+, D5 ; (D5) = 00000063 (A6) = 1033
CMP.B #BUSCAR, D5 ; (D5) ≠ "a" → (D5) = "a"
BEQ    ENCONTRADO ; Salto a ENCONTRADO

SIGUE  CMPI.B #*, (A6) ; "*" ≠ 61
       DBEQ  D6, CUENTA ; (D6) = 000004
       MOVE.L CONTADOR, D4 ;
       TRAP #1

ENCONTRADO ADDI #1, CONTADOR
           JMP  SIGUE

CODENA DC.B 'CASE*'
CONTADOR DS.L 1

END
    
```

(D5) = ... 61 (D5) = ... 73 (D5) = ... 73
 (A6) = 1034 (A6) = 1035 (A6) = 1036

(D5) ≠ "a" (D5) = "a"
 (D5) ≠ "a" (D5) = "a"
 "*" ≠ 73 (D6) = 0003
 "*" ≠ 73 (D6) = ... 02
 "*" ≠ 2A (D4) = 2
 FIN

(1039) = 1 (1039) = 2

DBEQ D6, CUENTA
 ↓
 No iguales → Decrem y salta
 Si iguales: → Ni decrem. ni salta

El programa CUENTA (CONTADOR) el n° de letras "BUSCAR" que hay en la cadena "CODENA" hasta encontrar "*" o con un n° maximo de "MAXIMO"

- (21) (A6) = 1033 ⇒
- (22) (D4) = 0000 ⇒ b
- (23) (D6) = 0003 ⇒ d
- (24) (D5) = 0073 ⇒
- (25) (D4) = 00020000 ⇒

E. AP 68000.6G

Feb 2000 - 1^{er} S - 16 Δ

ORG 2500
 TAB DS.L \$12
 LIB EQU *+2048
 CAN DC.W 256+LIB
 ST DC.W \$400+CAN
 END

$$\begin{array}{r} 2500 \quad \overline{)16} \quad \Rightarrow \$9C4 \\ 90 \quad \overline{)156} \quad \overline{)16} \\ 100 \quad \overline{)12} \quad \overline{)9} \\ 04 \end{array}$$

2048 ⇒ \$800

256 ⇒ \$100

DIR MEM

TAB 9C4
 → ABC+800 = 120C ⇒ LIB = 120C
 CAN ADC
 ST A0E
 ↓
 C

Reserva \$12 largos ⇒

$$\begin{array}{r} 00010010 \\ \underline{100} \quad \Rightarrow 1001000 = 048 + \frac{48}{ABC} \end{array}$$

120C + 100 = 130C / CAN = \$A0C

$$\begin{array}{r} A0C \\ 400 \\ \hline E0C \end{array}$$

Sep 2000 - 16. Δ

(D0) = \$00000003 (D1) = \$A3B00505 (D2) = \$A3080385

BUCLE	EOR.W	D2,D1	;	A 3 B 0 0 5 0 5	EOR	0101	0000
				A 3 0 B 0 3 8 5		0011	0000
SWAP		D1		A 3 B 0 0 6 8 0		0110	1000
SUBJ.B	#1,D0			(D1) = 0680 A3B0			
BNE		BUCLE		(D0) = 2			

2ª pasada

A 3 0 B	0 3 8 5	EOR	1000
0 6 8 0	A 3 B 0		1011
(D1) =	0 6 8 0	A 0 3 5	0011

SWAP D1 ⇒ (D1) = A035 0680

SUBJ.B #1,D0 ⇒ (D0) = 1

3ª pasada

A 3 0 B	0 3 8 5	EOR	0011
A 0 3 5	0 6 8 0		0110
A 0 3 5	0 5 0 5		0101

SWAP D1 ⇒ (D1) = 0505 A035

SUBJ.B #1,D0 ⇒ D0 = 0 ⇒

(D1) = 0505 A035 ⇒ a

2007 2S G S F16

¿Cuál será el contenido de de D1 y de C tras ejecutar ROXR.W D2,D1 si inicialmente C=X=1 y (D1)=\$A18C49B2 Y (D2)=\$00000005

- a) (D1)=\$A18C5000 b) (D1)=\$00002A4D c) (D1)=\$A18C2A4D d) (D1)=\$A18C924D