

ADDRESS REGISTER INDIRECT with POST-INCREMENT

EA IS THE CONTENT OF A_i .

OPERAND IS THE CONTENT OF $M[EA]$.

$\Rightarrow (A_i) +$

IF $(A_i) +$ IS IN AN INSTRUCTION CONTAINING

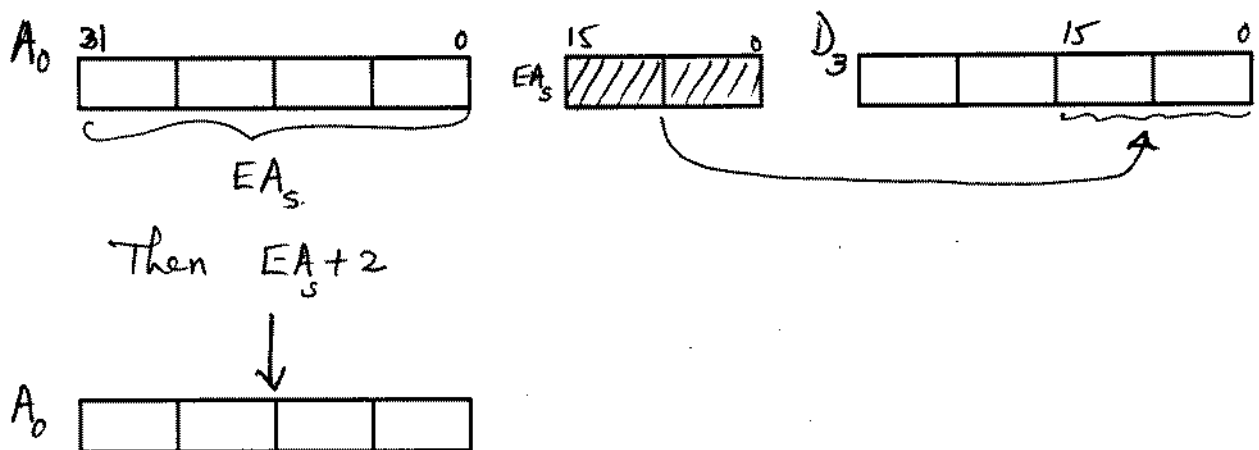
.B THEN $A_i \leftarrow A_i + 1$

.W THEN $A_i \leftarrow A_i + 2$

.L THEN $A_i \leftarrow A_i + 4$

AFTER THE INSTRUCTION IS EXECUTED.

e.g., MOVE.W $(A0) +$, D3



ADDRESS REGISTER INDIRECT with PRE-DECREMENT

⇒ $-(A_i)$

IF $-(A_i)$ IS IN AN INSTRUCTION CONTAINING

.B THEN $A_i \leftarrow A_i - 1$

.W THEN $A_i \leftarrow A_i - 2$

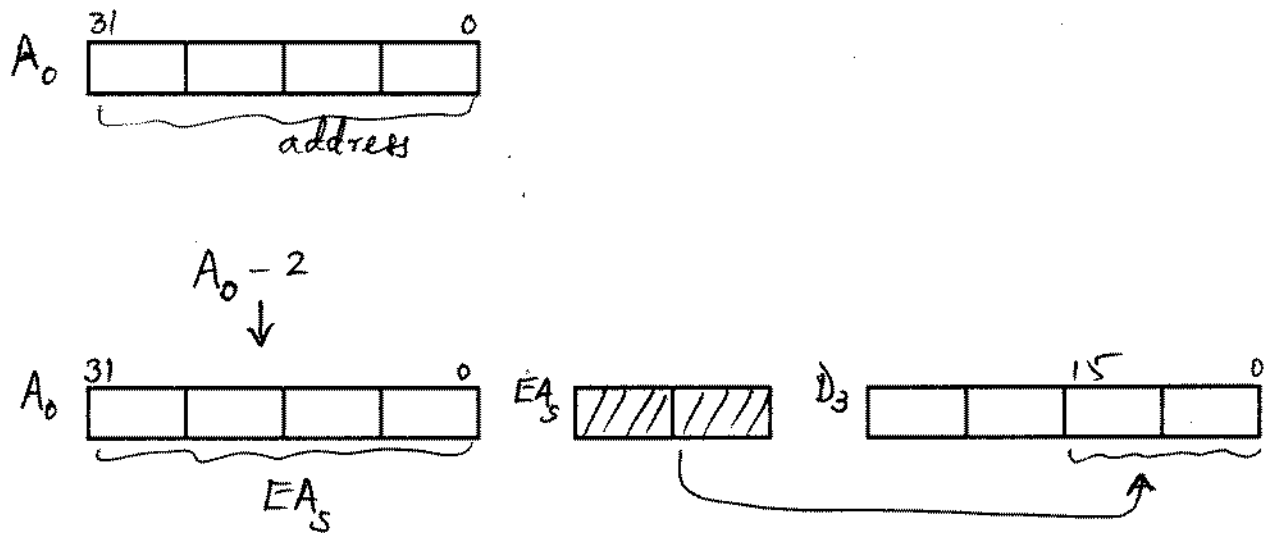
.L THEN $A_i \leftarrow A_i - 4$

BEFORE THE INSTRUCTION IS EXECUTED.

EA IS THE CONTENT OF $A_i - (1, 2, \text{ or } 4)$.

OPERAND IS THE CONTENT OF $M[EA]$.

e.g., MOVE.W $-(A0)$, D3



ADDRESS REGISTER INDIRECT with DISPLACEMENT

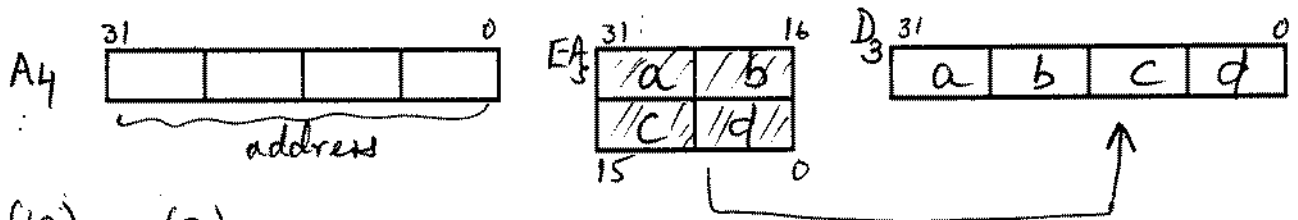
$$EA = d_{16} + A_i$$

(d_{16} is a 16-bit number in 2's CF that will be sign extended to a 32-bit number)

OPERAND IS THE CONTENT OF $M[EA]$.

$$\Rightarrow d_{16}(A_i)$$

e.g., `MOVE.L 12(A4), D3`



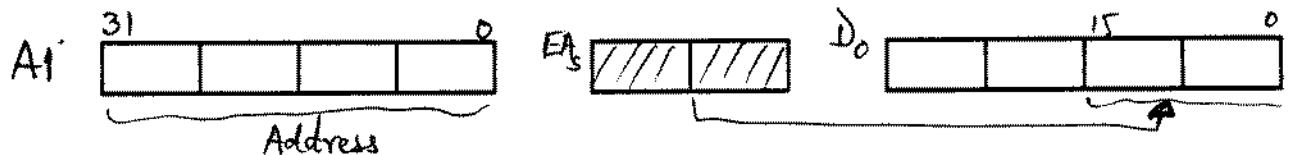
$$(12)_{10} = (C)_{16}$$

$$\Rightarrow (000C)_{16}$$

$$\Rightarrow (0000\ 000C)_{16}$$

$$\therefore EA_s = (0000\ 000C)_{16} + (A4)$$

`MOVE.W -$04(A1), D0`



$$(04)_{16} = (04)_{10} = (0000\ 0000\ 0000\ 0100)_2$$

$$\begin{array}{r} 1111\ 1111\ 1111\ 1011 \leftarrow 1's\ CF \\ + \\ \hline \end{array}$$

$$(-4)_{10} = (1111\ 1111\ 1111\ 1100)_2 = (FFFC)_{16}$$

$$\Rightarrow (FFFF\ FFFC)_{16}$$

$$\therefore EA_s = (FFFF\ FFFC)_{16} + \text{Address}$$

ADDRESS REGISTER INDIRECT with INDEXING

$$EA = d_8 + A_i + X_j$$

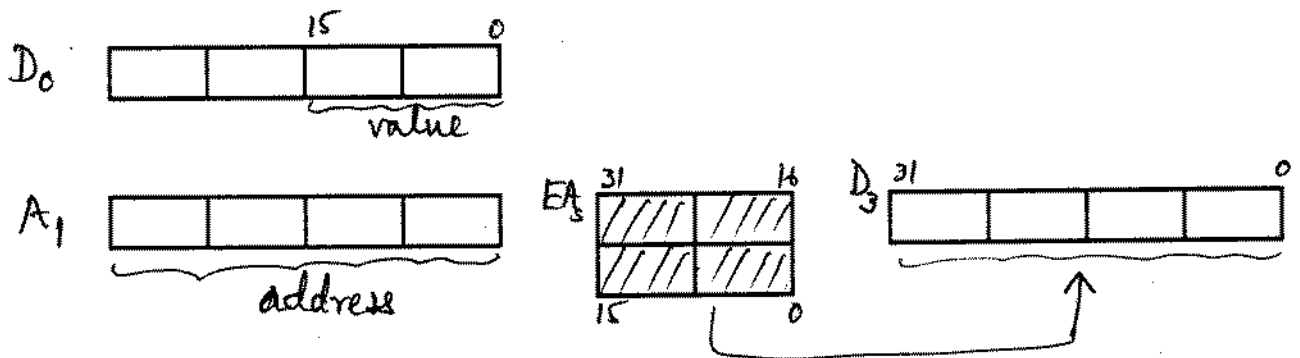
(d_8 is an 8-bit number in 2's CF that will be sign extended to a 32-bit number)

(X_j is the register for indexing)

OPERAND IS THE CONTENT OF $M[EA]$.

$\Rightarrow d_8(A_i, X_j.W)$ where $X_j.W$ will be sign extended to a 32-bit number
or
 $d_8(A_i, X_j.L)$

e.g., MOVE.L 9(A1, D0.W), D3



$$\text{value} = (HHHH)_{16} \Rightarrow \begin{pmatrix} 0000 & HHHH \\ FFFF & \end{pmatrix}_{16}$$

$$(9)_{10} = (9)_{16} = (0000 \ 0009)_{16}$$

$$\therefore EA_s = \text{Address} + (0000 \ 0009)_{16} + \begin{pmatrix} 0000 & HHHH \\ FFFF & \end{pmatrix}_{16}$$

Register (Xi = content):

A0 = \$0000 2000
 A1 = \$0000 2004
 D0 = \$1266 6768
 D1 = \$AAAB B2DD
 D2 = \$100A 0005
 D3 = \$ABCD 573D

Memory (address = content):

\$002000 = \$1234
 \$002002 = \$5678
 \$002004 = \$BEAD
 \$002006 = \$F00D
 :
 \$00200C = \$A110

\$002000

\$002002

12	34
56	78

Absolute

	<u>For .B</u>	<u>For .W</u>	<u>For .L</u>
MOVE.B \$2000, \$2004 \$002004 =	\$12AD	\$1234	\$1234
\$002006 =			\$5678

Immediate

MOVE.B #\$20, D0	D0 = \$1266 6720	\$1266 0020	\$0000 0020
MOVEA.W #\$20, A1	A1 = N/A	\$0000 0020*	\$0000 0020

Direct

MOVE.B D0, D1	D1 = \$AAAB B268	\$AAAB 6768	\$1266 6768
MOVE.W A0, D1	D1 = N/A	\$AAAB 2000	\$0000 2000
MOVEA.W D0, A0	A0 = N/A	\$0000 6768*	\$1266 6768
MOVEA.W D1, A0	A0 = N/A	\$FFFF B2DD*	\$AAAB B2DD
MOVEA.W A1, A0	A0 = N/A	\$0000 2004*	\$0000 2004

Indirect

MOVE.B (A0), D3	D3 = \$ABCD 5712	\$ABCD 1234	\$1234 5678
MOVEA.W (A1), A4	A4 = N/A	\$FFFF BEAD*	\$BEAD F00D

MOVE.B D0, (A0)	\$002000 = \$6834	\$6768	\$1266
	\$002002 =		\$6768

MOVE.B (A0), (A1)	\$002004 = \$12AD	\$1234	\$1234
	\$002006 =		\$5678

Displacement

MOVE.B \$2(A0), D0	D0 = \$1266 6756	\$1266 5678	\$5678 BEAD
--------------------	------------------	-------------	-------------

Post-increment

MOVE.B (A1)+, D3	D3 = \$ABCD 57BE and A1 = \$0000 2005
MOVE.W D3, (A1)+	\$002004 = \$573D and A1 = \$0000 2006

Post-decrement

MOVE.B -(A1), D3	A1 = \$0000 2003 and D3 = \$ABCD 5778
MOVE.W D0, -(A1)	A1 = \$0000 2002 and \$002002 = \$6768

Displacement

MOVE.W \$3(A1, D2.W), D0	D0 = \$1266 A110
--------------------------	------------------

(Note: \$0000 2004 + \$0000 0005 + \$0000 0003 = \$0000 200C).

Relative to PC

MOVE.W 4(PC), D0	D0 = \$1266 BEAD	(given PC = \$002000)
------------------	------------------	-----------------------

* Sign extension to 32 bit for address register