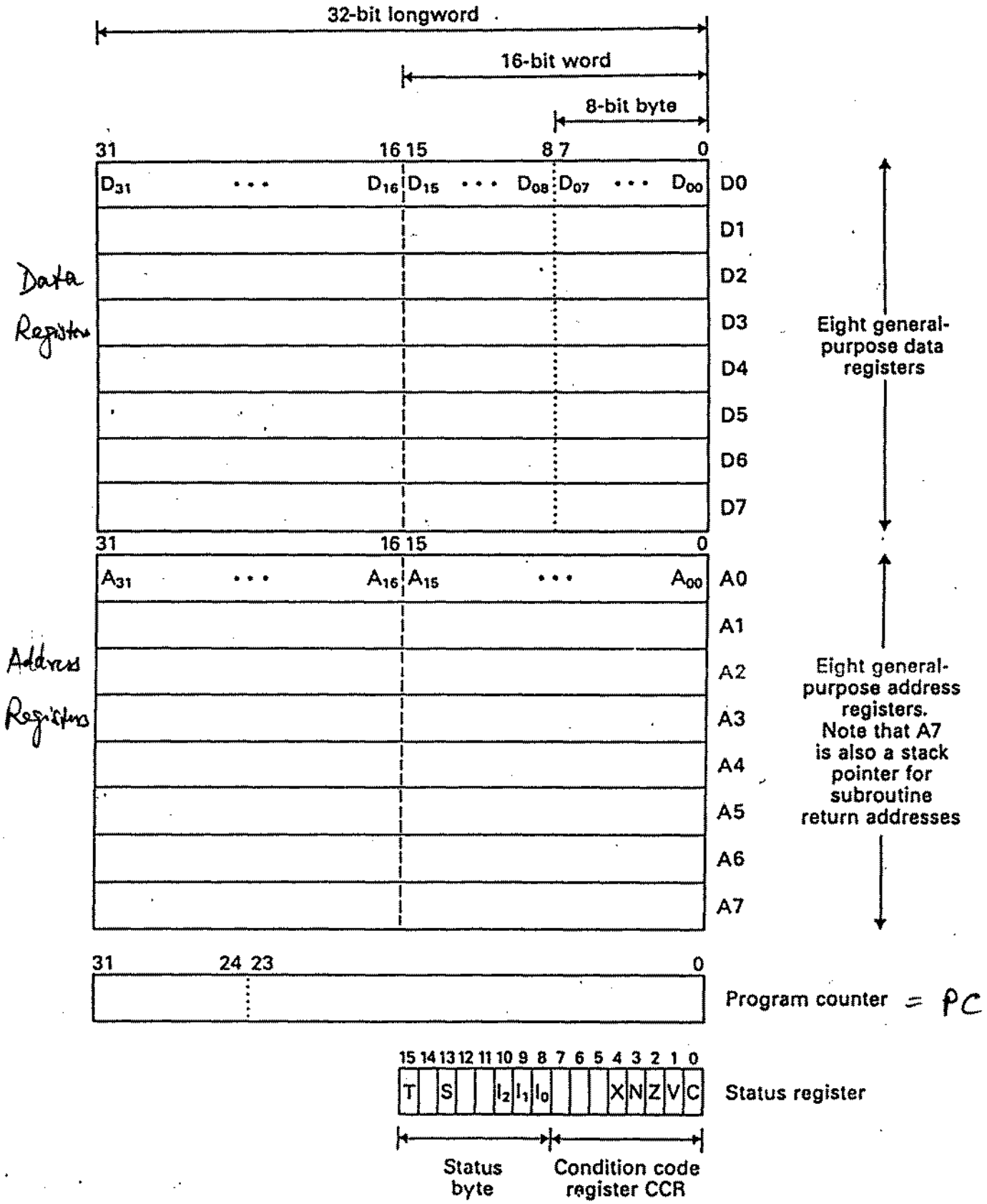
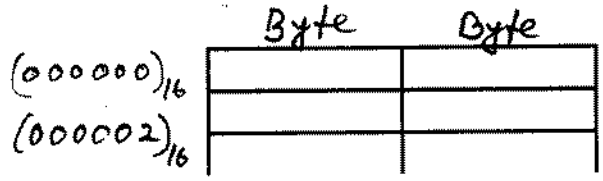
Programming model of the 68000



MC68000

* HAS 24-BIT ADDRESSES
  $\Rightarrow (000000)_{16}$ - $(FFFFFF)_{16}$ ADDRESS SPACE

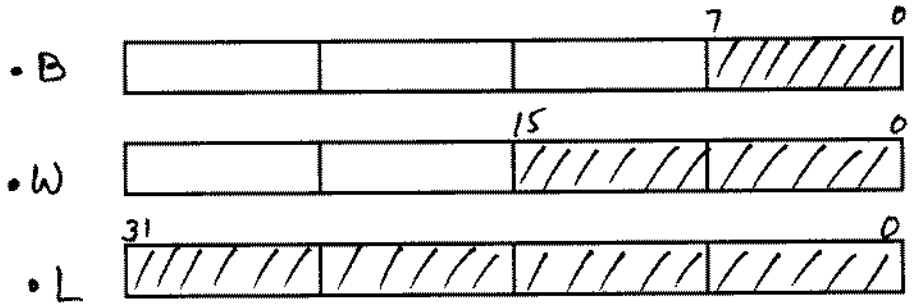
* BYTE-ADDRESSABLE
  $\Rightarrow 2^{24}$ BYTES $\Rightarrow$ 16M BYTES MAIN MEMORY



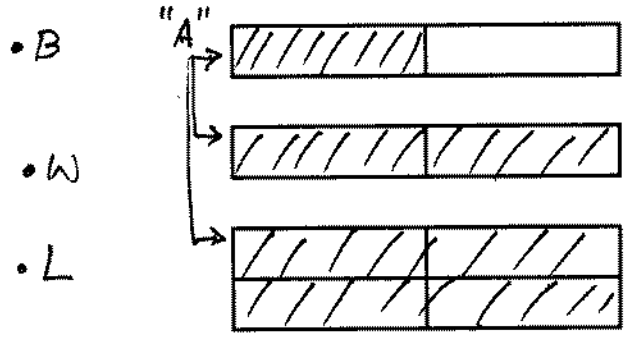
* PERMITS OPERATIONS ON **8-**, **16-**, **32**-BIT OPERANDS BY ALLOWING SUFFIX **.B**, **.W**, **.L** TO BE USED AFTER OPCODE MNEMONICS

- .B for byte
- .W for word
- .L for longword

* IF OPERAND IS IN A REGISTER



* IF OPERAND IS IN MAIN MEMORY



At address "A", pick byte, word or longword.

ADDRESSING MODES

ABSOLUTE
IMMEDIATE
REGISTER DIRECT
REGISTER INDIRECT
REGISTER INDIRECT with POST-INCREMENT
REGISTER INDIRECT with PRE-DECREMENT
REGISTER INDIRECT with DISPLACEMENT
REGISTER INDIRECT with INDEXING

Notation

A_i	Address Register	where $0 \leq i \leq 7$
D_i	Data Register	where $0 \leq i \leq 7$
X_i	Address or Data Register	where $0 \leq i \leq 7$

d_8	8-bit signed offset
d_{16}	16-bit signed offset
d_{32}	32-bit signed offset

$H ... H	Memory Address where H is an hexadecimal digit

N	$(N)_{10}$	where N is a number
$N	$(N)_{16}$	where N is a number
%N	$(N)_2$	where N is a number

'S'	$(S)_{ascii}$	where S is any sequence of alphanumeric characters

RTL (Register Transfer Language)

←	Transfers a copy of the right hand side to the left hand side
comma	Separates simultaneous transfers
M[ ]	Specifies the memory location whose address is given within [ ]
R( )	Specifies the part of a register R which is identified within ( )

ADDRESSING MODES of MC68000

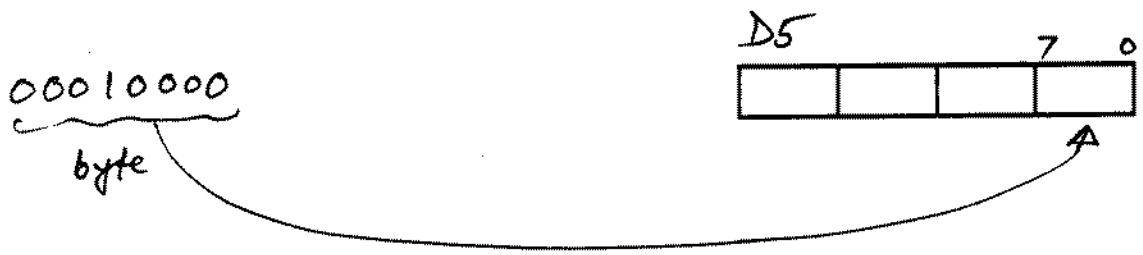
IMMEDIATE

OPERAND IS GIVEN WITHIN THE INSTRUCTION.
OPERAND IS A **CONSTANT.**

⇒ **# CONSTANT** to represent a constant

e.g., #10 ⇒ $(10)_{10}$
#$10 ⇒ $(10)_{16}$
#%10 ⇒ $(10)_2$

e.g., MOVE.B #$10, D5



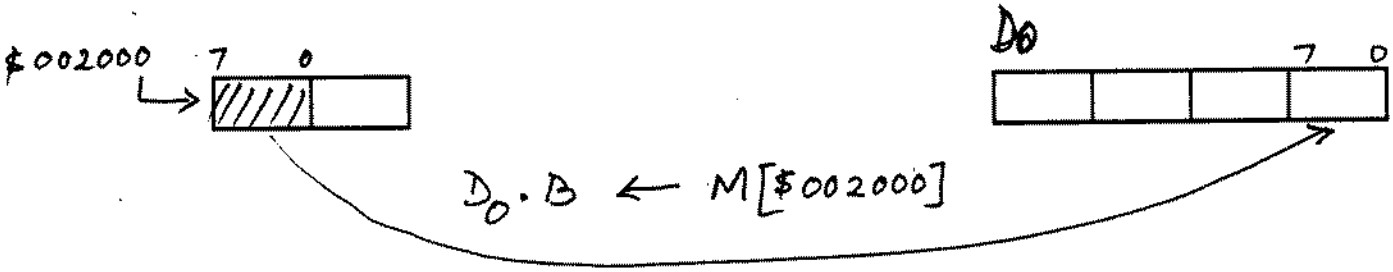
NOTE:
For the remaining address modes
EA ≡ Effective Address
= Address of the operand.

ABSOLUTE

EA IS GIVEN WITHIN THE INSTRUCTION.
OPERAND IS THE CONTENT OF M[EA].

⇒ **$HHHHHH**

e.g., MOVE.B $002000, D0



RELATIVE

EA IS GIVEN WITHIN THE INSTRUCTION.
OPERAND IS THE CONTENT OF M[EA].

⇒ **VARIABLE**

e.g., MOVE.W AVERAGE, D0

Go to the address of Average and copy the content of M[Average] to D0.

D0.W ← M[Average]

A PARTICULAR USE OF ABSOLUTE/RELATIVE ADDRESSING
IS IN A BRANCH OR JUMP INSTRUCTION
WHERE THE ADDRESS TO BRANCH OR JUMP IS INDICATED

- IN ABSOLUTE ADDRESSING BY **$HHHHHH**

  e.g., BRA $HHHHHH

- IN RELATIVE ADDRESSING BY **LABEL**

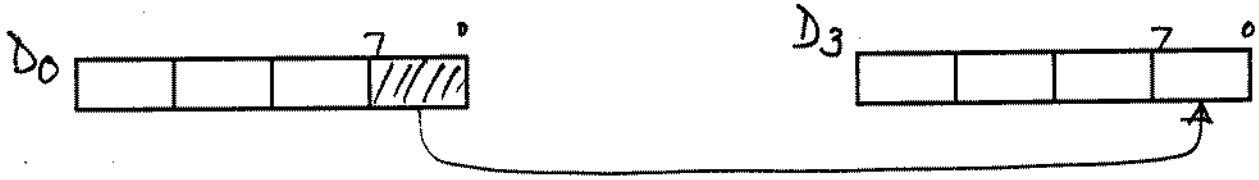
  e.g., BRA DONE

REGISTER DIRECT (operand is in a register)

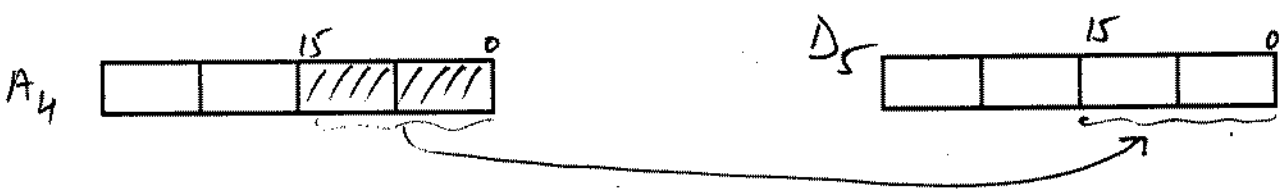
EA IS X_i.
OPERAND IS THE CONTENT OF X_i.

⇒ **A_i**
or
D_i

e.g., MOVE.B D0, D3

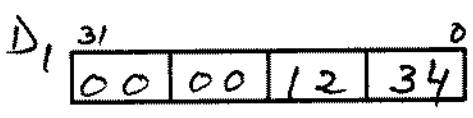


MOVE.W A4, D5



MOVE.L #$1234, D1

$(1234)_{16}$ = $(0000\ 1234)_{16}$



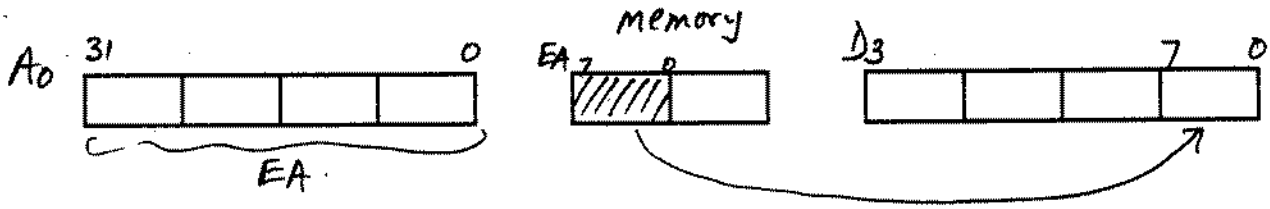
ADDRESS REGISTER INDIRECT

operand is in a memory location whose address is in an address register.

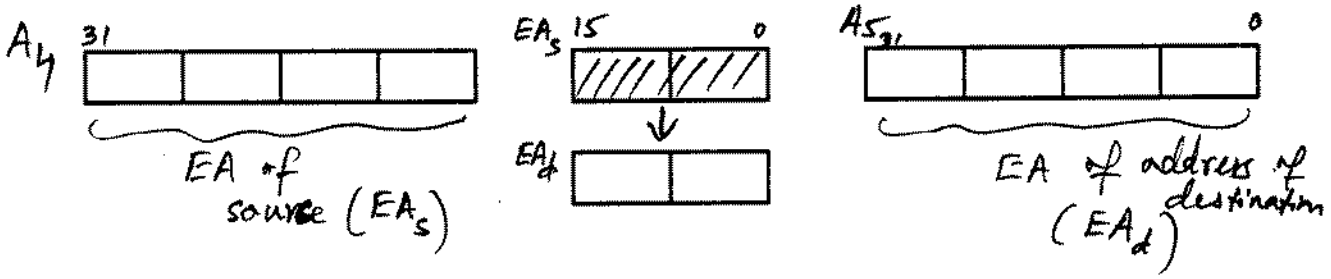
EA IS THE CONTENT OF A_i.
OPERAND IS THE CONTENT OF M[EA].

$\Rightarrow$ (A_i)

e.g., MOVE.B (A0), D3



MOVE.W (A4), (A5)



MOVE.L #$1234, (A1)