ADDRESSING MODES

- REGISTER ADDRESSING
- IMMEDIATE
- DIRECT
- BASED -REGISTER INDIRECT
- INDEXED
- BASED INDEXED
- BASED INDEXED WITH DISPLACEMENT

REGISTER ADDRESSING

① FIRSTLY GOOGLE 8086_family_Users_Manual.pdf
② DOWNLOAD FIRST LINK
③ LOCATE p-171
④ OBSERVE THE BINARY ENCODINGS FOR THE FOLLOWING INSTRUCTIONS

INSTRUCTION	BINARY
INC AX	0100 0000
INC BX	0100 0011
INC CX	0100 0001
INC DX	0100 0010
ETC.	

ALL THE PRECEDING INSTRUCTIONS USE REGISTER ADDRESSING
I.E. THE OPERAND IS IN A REGISTER

IMMEDIATE ADDRESSING

THE OPERAND IS ENCODED AS A BINARY NUMBER SOMEWHERE IN THE SEQUENCE OF BITS THAT ENCODE THE INSTRUCTION

E.G.

INSTRUCTION	BINARY
ADD AX, BX	0000 0011 1100 0011

HEX

ADD AX, BX 03C3

INSTRUCTION	BINARY
ADD AX, 1	0000 0101 0000

HEX

ADD AX, 1 050100

0000 0001 0000 0000

ADD AX, 1 = 0000 0101 0000 0001 0000 0000

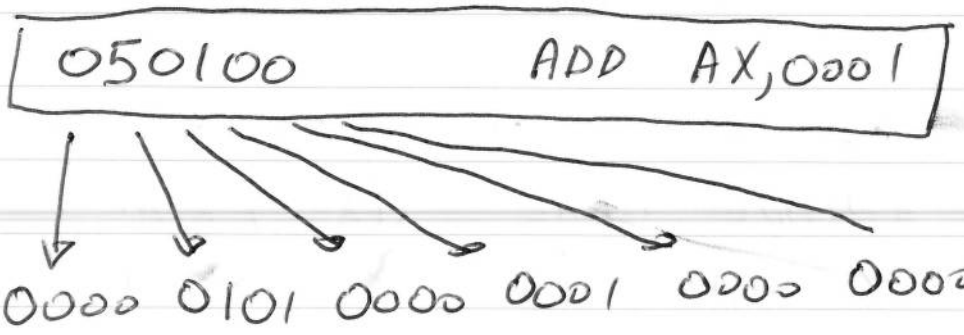
Q WHERE DID WE GET THAT FROM ?

A. ① PUT THE INSTRUCTION
ADD AX, 1 INTO AN ASSEMBLY PROGRAM.

② ASSEMBLE THE PROGRAM
E.G. TASM MYPROG

③ DEBUG THE PROGRAM
E.G. TD MYPROG

④ FIND THE LINE:
ADD AX, 0001
LOOK TO THE LEFT HAND SIDE OF THIS LINE AND YOU WILL SEE THE FOLLOWING:



So:

0000 0101 0000 0001 0000 0000

THE ABOVE IS OUR INSTRUCTION ENCODED IN BINARY

④

Q. WHAT CREATED THE INSTRUCTION ENCODING ?

A. THE ASSEMBLER TASM CREATED THE OBJECT CODE .OBJ
THE LINKER TLINK CREATED THE EXECUTABLE CODE .EXE

LETS LOOK AT THE INSTRUCTION AGAIN

ADD AX,1 0000 0101 | 0000 0001 0000 0000 |

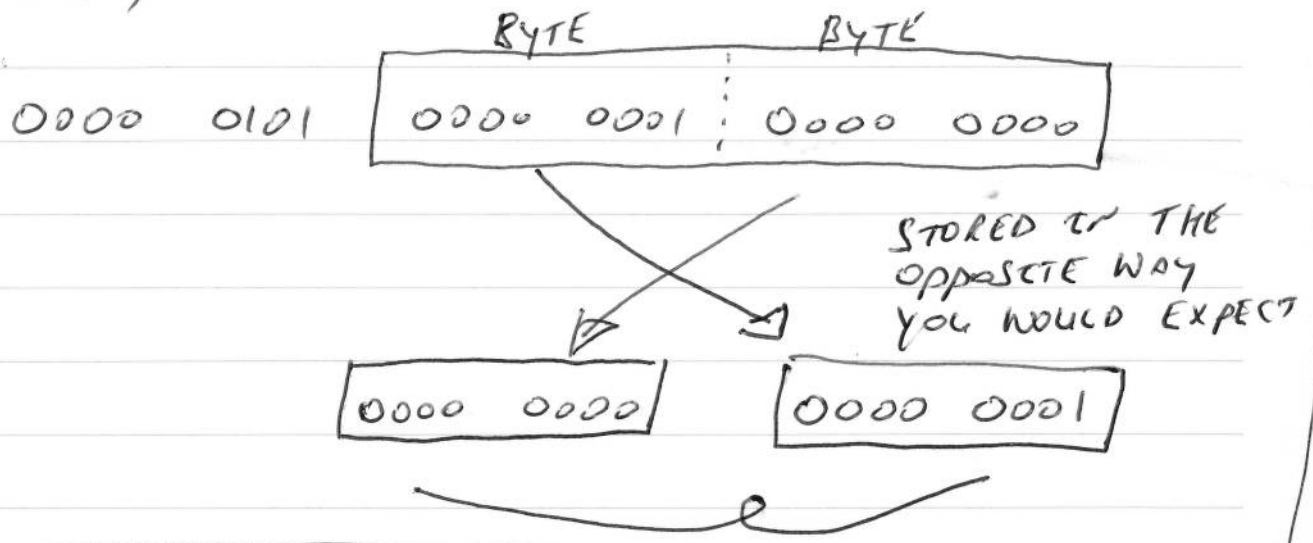
THE OPERAND IS ENCODED AS A BINARY NUMBER SOMEWHERE IN THE SEQUENCE OF BITS THAT ENCODE THE INSTRUCTION.

WHERE ~~ABS~~ IS THE SEQUENCE OF BITS THAT ENCODE THE (1) FROM ADD AX, (1) IN THE ABOVE EXAMPLE ?

ANSWER: THE 16 BITS INSIDE THE BOX

DOES IT LOOK LIKE A 1 ?

ADD AX, 1:



WHAT ABOUT

ADD AX, 371

HOW IS THE ABOVE ENCODED?

TRY TO WORK IT OUT ON PAPER. FIRST. THEN PUT IN AN ASSEMBLY PROGRAM. DEBUG THE ASSEMBLY PROGRAM TO VERIFY.

EXPRESS ANSWER IN HEX AND BINARY

371 IN BINARY → 0000 0001 0111 0011

0 1 7 3

05 73 01