

①

ADDRESSING MODES

OPERAND IS IN A REGISTER

REGISTER ADDRESSING E.G. ADD AX, BX

REGISTER INDIRECT ADDRESSING

MEMORY VARIABLE

.DATA

HRSWORKED ~~DB~~ DW 0

HRLYRATE DW 40

.CODE

MOV AX, @DATA

MOV DS, AX

MOV AX, 0

MOV BX, 2

MOV BX, OFFSET HRLYRATE

BX CONTAINS OFFSET: 2

ADD AX, [BX]

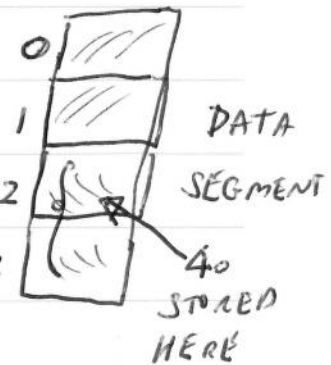
CALL OUTDEC

WHAT IS IN AX?

HRSWORKED

BX: 2 → HRLYRATE → 2

□ = BYTE



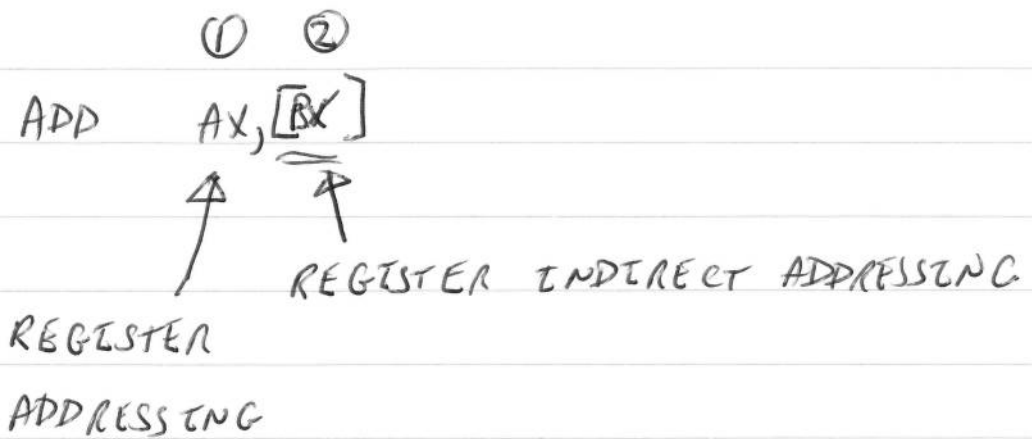
OFFSET OF HRLYRATE IS 2

ADD AX, [BX]

↑
DESTINATION

↑
WHAT BX POINTS TO
WHAT DOES BX POINT TO?

(2)



	<u>WHERE IS THE OPERAND</u>
① REGR. ADDRESSING	IN A REGISTER: AX
② REGR. INDIRECT ADDRESSING	IN MEMORY

Q. WHICH IS FASTER TO ACCESS?

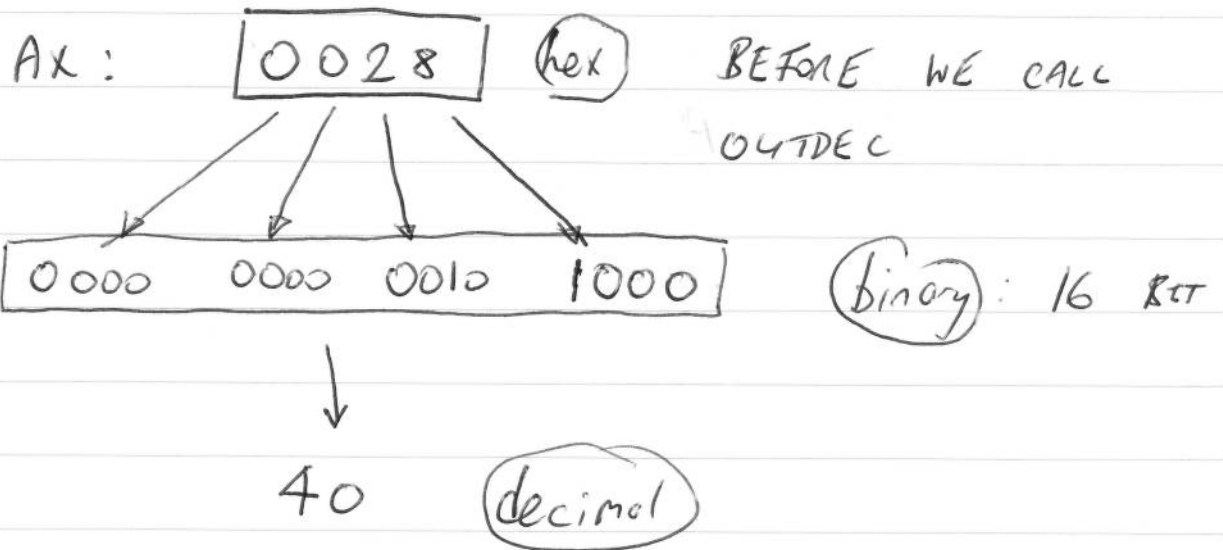
A. REGISTER

ADD AX, [BX]

ADDS ^{THE} 40, WHICH IS STORED IN MEMORY TO AX. AX CONTAINS 40 AFTER EXECUTION.

1. TASM FEB9 ← ASSUME ^{PROGRAM IS IN} ~~FEB9~~ FEB9.ASM
2. TASM OUTDEC
3. TLINK FEB9 OUTDEC → FEB9.EXE PRODUCED
4. FEB9 (RUN PROGRAM)

(3)



BASED ADDRESSING

- CAN USE BP (BASE POINTER)
- OUR EXAMPLE WILL USE THE STACK

• CODE

CALL ADDNUMS

MOV AX, 4C00h
int 21h

ADDNUMS: MOV BP, SP



ret

(RETURN FROM PROCEDURE)

STACK POINTER

CODE TO
ACCESS TWO
PARAMETERS

(4)

```
MOV AX, 2
MOV BX, 7
```

— TWO PARAMETERS

```
PUSH AX
PUSH BX
```

```
POP AX
POP AX
```

STACK : RETURN ADDRESS

PASS PARAMETERS TO PROCEDURES

```
MOV AX, 2
PUSH AX
MOV AX, 7
PUSH AX
```

```
POP AX
POP AX
```

LEAVE
STACK
CONSISTENT

EXAMINE ABOVE BLOCK OF CODE IN
DEBUGGER.

WHEN YOU PUSH A VALUE :

$$SP \text{ REGISTER : } SP = SP - 2$$

WHEN YOU POP A VALUE :

$$SP = SP + 2$$

(5)

• CODE

MOV AX, 2

PUSH AX

MOV AX, 7

PUSH AX
—

CALL ADDNUMS

MOV AX, 4C00h

int 21h

ADD TWO PARAMETERS

MOV BP, SP

ADDNUMS:

MOV BX

WHEN YOU CALL A PROCEDURE: / USING CALL INSTRUCTION

① $SP = SP - 2$

② RETURN ADDRESS IS PUSHED ONTO STACK.

WHEN YOU RETURN FROM A PROCEDURE / USING RET INSTRUCTION

① $SP = SP + 2$

② RETURN ADDRESS IS POPPED OFF STACK AND COPIED INTO IP (INSTRUCTION POINTER).

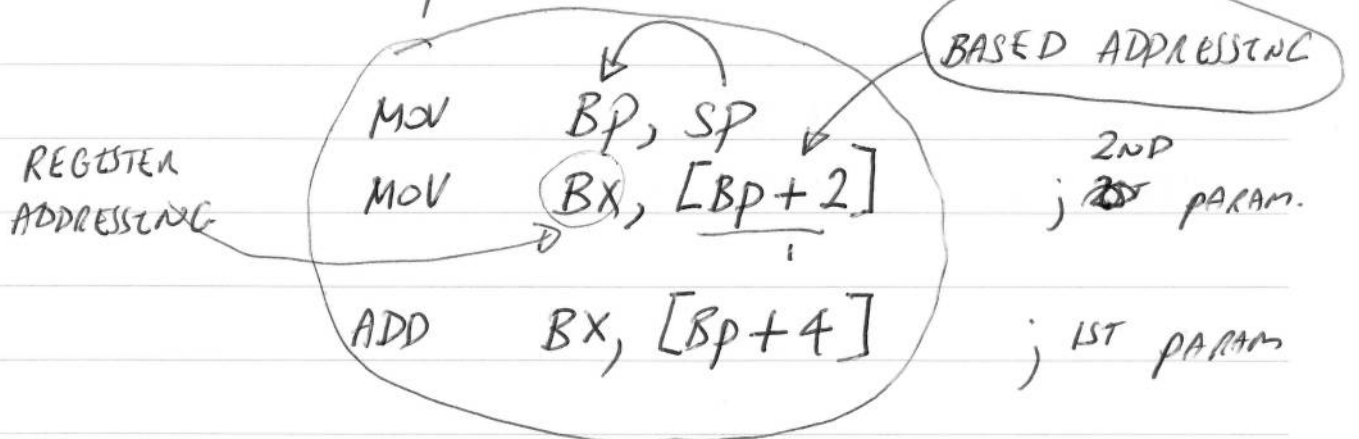
(6)

AFTER WE CALL THE PROCEDURE THE STACK LOOKS LIKE:

	0100:	0002	1ST PARAMETER
	00FE:	0007	2ND "
Sp: 00FC	00FC:	→ 000B	RETURN ADDRESS

Sp: 00FC

INSIDE PROCEDURE



; AT THIS POINT BX CONTAINS THE TOTAL WHEN YOU ADD 1ST AND 2ND PARAMETERS.