Experiment No:1

FAMILIARISATION OF X86 ASSEMBLY LANGUAGE PROGRAMMING

Sim:
To familiarise with assembler directives, addressing modes and memory modes.

Assembly language is a low-level programming language closely associated with a computer's architecture x86 Assembly is written specifically for Intel's 8086 and later processors, and it provides direct control over hardware using simple mnumonics and machine-level instructions.

She Intel x 86 architecture is a CISC (Complex Instruction Set Computer) architecture inbroduced with the 8086 processor. It uses a segmented memory model, supports 16-bit registers, and is backward compatable with larlier Intel processor. Key features include:

. 16-bit data and address bus (8086).

· 20 - bit physical address space allowing vaccess to IMB of memory.

· Use of memory regmentation for logical · Includes General Prospose Pregisters, segment adobussing. Registers, and Assemblers An assembler converts assembly language source code (.ASM) into machine code (.OBJ), .EXE) - Popular 286 assemblers include: · TASM (Surbo Assembler) - Borland's assembler, often used with DOSBOX ·MASM (Microsoft Macro Assembler) - Official Microsoft assembler-· EM V8086 - Educational assembler with an emulator for running 8086 code as Linuis. as dinuix. Assembler Workflow Source Code (-ASM) -> Assembler -> Object File (.OBJ) -Shinkes Registers in x86 Registers are small, fast storage locations in the

General Purpox Registers (16-bit): Description Register Accumulator AX BX Counter CX DX Data Each of these can be spilt into 8- leit halves · AX -> AH (high) and AL (low), etc Segment Registers Description lode segment Register CS Data Segment DS Stack Segment SS Extra Segment ES

Painter and Index Registers:
·SI (Source Index), DI (Destination Index).
Memory Sigmentation
286 architectur was segmented memory
A physical addres = segment x 16 + Offset
Same of a
· Coole Segment (ES): Contains the instructions to be executed.
- Det So + (De) - Store requiable and constants
Stack & (Cs): Manages Sundian calls and
leral variable.
· Data Segment (DS):- Stores reariables and renotants · Stack Segment (SS): Manages function calls and local variables · Ealie Segment (ES): Vsed in string and memory operations
Assembler Directives
A A H A STATE OF THE STATE OF T
Assembles clirectures are commands is the warmen
Assembles clirectives are commands to the assembles and do not generale machine code.
tommon cluctures
Directive Meaning
model Defines memory model (eg: small, tiny)
data starts the data segment
coole starts the code segment

.

.

end Reserves stack space end Ends the program

Addressing Modes.
Addressing modes define how operands are accressed.

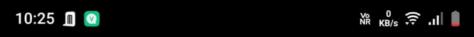
Mode	Example	Description
Immediate	mov al, 25	operand is not a constant
Register	mov ax, bx	Operand is a register
Direct	mov al, [1234h]	Operand is at memory address
Register Indirect	mov al, [bx]	Memory adolners in a suguiter Uses index suguiters
Indesced	mov al, [si]	Vses index jugisters
Based Indesed	mov al, [bx+si]	Combines base and index

Memory M	noolels in x8 roolels define	how code	and data are organized
Model	lode	Data	Notes
Diny			look and data in one so
small			lode and data in separate
Meduin	Many		Multiple code segments
Compact	1	Many	Multiple data segments
large	Many	Many	Multiple code and data segments

For Simple programs, model small is most commonly used.

RESULT:

Successfully familiavised with 286 programming.



#### ChatGPT



# Algorithm (8 Lines)

- Start the program and initialize data segment.
- 2. Display the message: "Enter two characters".
- Read the first character from the keyboard.
- 4. Store the first character in memory.
- Read the second character from the keyboard.
- 6. Store the second character in memory.
- Display the message: "You entered: " followed by the two characters.
- 8. Terminate the program.

Let me know if you also want a **flowchart** or **block diagram**.

日 A 日 Co 子

Ask anything







### ChatGPT



## Algorithm (10 Lines)

- Start the program and initialize the data segment.
- 2. Declare an 8-bit value 0x7A using DB.
- 3. Declare a 16-bit value 0x3456 using DW.
- 4. Declare a 32-bit value 0x11223344 using DD.
- 5. Load the 8-bit value into register AL.
- Increment AL and store it back in memory.
- 7. Load the 16-bit value into register AX.
- Increment AX and store it back in memory.
- Load the 32-bit value into register EAX, increment it, and store it back.
- Terminate the program using DOS interrupt INT 21h.

1 at ma lenaur if you need this as a flaurahaut

Ask anything





#### ChatGPT



Here is the 8-line algorithm to add two 8-bit numbers in Assembly (MASM):

# Algorithm to Add Two 8-bit Numbers (8 Lines)

- Start the program and initialize the data segment.
- Load the first 8-bit number into register AL.
- 3. Load the second 8-bit number into register BL.
- Add the contents of BL to AL (AL = AL + BL).
- 5. Check for carry if needed (optional step).
- 6. Store or display the result from AL.
- 7. Terminate the program.
- 8. End.



Ask anything



