UNIT VIII · LESSON 02

8086 Microprocessor

architecture (BIU / EU) · registers · segmentation · flags · min/max mode · addressing modes · instruction set · interrupts · assembler directives

20
questions in Unit VIII
8085 · 8086 · 8051 · ARM · RTOS

FEATURES 8086 AT A GLANCE

- Intel, **1978**, **16-bit** processor, HMOS, 40-pin, +5 V, 5/8/10 MHz
- **16-bit data bus**, **20-bit address bus → 1 MB** memory; 64 K I/O ports (16-bit port address)
- **Pipelining** via a **6-byte instruction queue** (prefetch); 8088 = same with 8-bit external bus, 4-byte queue
- Multiplexed AD_0–AD_{15}, A_{16}/S_3–A_{19}/S_6; **BHE'** selects the high byte (odd bank)
- Memory banks: **even (A_0 = 0, D_0–D_7)**, **odd (BHE' = 0, D_8–D_{15})**

14 × 16-BIT REGISTERS

Group	Registers	Special use
General	**AX** (AH, AL), **BX** (BH, BL), **CX** (CH, CL), **DX** (DH, DL)	AX accumulator (MUL/DIV, I/O); BX base; **CX count** (LOOP, REP, shifts CL); DX data (I/O port, MUL/DIV high word)
Pointer	SP, BP	stack top; base pointer (stack frames)
Index	**SI, DI**	source / destination index (string operations)

BIU + EU ARCHITECTURE

Unit	Contains	Job
BIU (bus interface unit)	segment registers CS, DS, SS, ES; IP; address adder (Σ); **6-byte queue**	fetch instructions, read/write memory & I/O, compute physical address
EU (execution unit)	ALU, general registers, flags, pointer & index registers, control	decode & execute

- BIU and EU work **in parallel** → fetch overlaps execution

14 × 16-BIT REGISTERS (CONTD.)

Group	Registers	Special use
Segment	CS, DS, SS, ES	code, data, stack, extra
Others	IP, FLAGS	

PHYSICAL ADDRESS SEGMENTATION

- **Physical address = segment × 16 (10H) + offset** (20 bits)
- Example: CS = 3000H, IP = 1234H → **31234H**; DS = 2000H, offset 0FFFFH → 2FFFFH
- Segment size **64 KB**; up to 16 non-overlapping segments in 1 MB
- Default pairs: **CS:IP**, **SS:SP / SS:BP**, **DS:BX, SI, DI**, **ES:DI** (string destination)
- Benefits: relocatability, 20-bit address with 16-bit registers, separation of code/data/stack

8086 ADDRESSING MODES

Mode	Example	Effective address
Immediate	`MOV AX, 1234H`	—
Register	`MOV AX, BX`	—
Direct	`MOV AX, [5000H]`	5000H
Register indirect	`MOV AX, [BX]`	BX (or SI, DI)
Based	`MOV AX, [BX + 08H]`	BX + disp (BP → SS)
Indexed	`MOV AX, [SI + 10H]`	SI + disp
Based indexed	`MOV AX, [BX + SI]`	BX + SI
Based indexed + displacement	`MOV AX, [BX + SI + 05H]`	BX + SI + disp

9 FLAGS FLAGS

Type	Flags
Status (6)	**CF**, **PF**, **AF**, **ZF**, **SF**, **OF** (signed overflow)
Control (3)	**TF** (trap, single-step), **IF** (interrupt enable), **DF** (direction for strings: 0 = increment, 1 = decrement)

- CLC/STC/CMC, CLD/STD, CLI/STI manipulate flags
- OF = 1 when 7FH + 1 (8-bit signed overflow)

MN/MX' PIN MIN / MAX MODE

Minimum mode (MN/MX' = 1)	Maximum mode (MN/MX' = 0)
single processor	multiprocessor (with 8087 coprocessor, 8089)
8086 generates control signals (ALE, DEN, DT/R', M/IO', WR', INTA', HOLD/HLDA)	**8288 bus controller** generates them from S_0–S_2
—	RQ'/GT' lines, LOCK', QS_0–QS_1 (queue status)

- 8284 clock generator, 8282/8283 latches, 8286/8287 transceivers

8086 ADDRESSING MODES (CONTD.)

Mode	Example	Effective address
String	`MOVSB`	DS:SI → ES:DI
I/O port	`IN AL, 80H` / `IN AL, DX`	direct (8-bit) / indirect (DX)

8086 ADDRESSING MODES (CONTD.)

Mode	Example	Effective address
Relative	`JMP short label`	IP + disp

256 TYPES INTERRUPTS

- **Interrupt vector table** at **00000H–003FFH**: 256 × 4 bytes (CS:IP)
- Vector address of type n = **n × 4** (INT 21H → 84H)

Type	Interrupt
0	divide error
1	single step (TF)
2	**NMI**
3	breakpoint (1-byte INT 3)
4	overflow (INTO)
5–31	reserved
32–255	user / software (DOS INT 21H)

- Priority: divide error/INT n/INTO (highest) → NMI → INTR → single step
- On interrupt: push FLAGS, clear IF & TF, push CS, push IP, load new CS:IP; **IRET** restores
- INTR (maskable, via 8259 PIC) needs IF = 1

GROUPS INSTRUCTION SET

Group	Examples
Data transfer	MOV, PUSH, POP, XCHG, XLAT, LEA, LDS, LES, IN, OUT, LAHF, SAHF, PUSHF, POPF
Arithmetic	ADD, ADC, SUB, SBB, INC, DEC, NEG, CMP, **MUL, IMUL, DIV, IDIV**, DAA, DAS, AAA, AAS, AAM, AAD, CBW, CWD
Logical / shift	AND, OR, XOR, NOT, TEST, SHL/SAL, SHR, SAR, ROL, ROR, RCL, RCR
String	MOVS, CMPS, SCAS, LODS, STOS with **REP, REPE/REPZ, REPNE/REPNZ**
Control transfer	JMP, Jcc, CALL, RET, **LOOP, LOOPE, LOOPNE, JCXZ**, INT, INTO, IRET
Processor control	HLT, WAIT, LOCK, ESC, NOP, STC, CLD, STI

- **MUL BL**: AX ← AL × BL; **MUL BX**: DX:AX ← AX × BX
- **DIV BL**: AL ← AX / BL (quotient), AH ← remainder; DIV BX: AX ← DX:AX / BX, DX ← remainder
- `LOOP label` = DEC CX; JNZ label (no flags changed)
- `LEA SI, ARRAY` loads the offset address
- `XLAT`: AL ← [BX + AL] (table look-up)

MASM ASSEMBLER DIRECTIVES

- **DB, DW, DD, DQ, DT** (define byte/word/doubleword ...), DUP
- **ASSUME** (segment registers), **SEGMENT / ENDS**, **PROC / ENDP**, END, ORG, EQU, OFFSET, PTR, LABEL, EVEN, NEAR/FAR

MASM ASSEMBLER DIRECTIVES (CONTD.)

```
DATA SEGMENT
  A  DW 1234H
  B  DW 0ABCDH
  S  DW ?
DATA ENDS
CODE SEGMENT
 ASSUME CS:CODE, DS:DATA
START: MOV AX, DATA
       MOV DS, AX      ; no immediate
       MOV AX, A
       ADD AX, B       ; AX = 0BE01H
       MOV S, AX
       MOV AH, 4CH
       INT 21H         ; return to DOS
CODE ENDS
END START
```

EVOLUTION 8086 FAMILY

CPU	Year	Bits	Address space
8086 / 8088	1978 / 79	16	1 MB (20-bit)
80286	1982	16	16 MB, protected mode
80386	1985	32	4 GB, paging, virtual 8086
80486	1989	32	on-chip cache + FPU
Pentium	1993	32 (64-bit bus)	superscalar

Q → A RAPID RECALL

8086 address bus	20 bits (1 MB)
Instruction queue size	6 bytes
Physical address of 1000H:2000H	12000H
Register used as loop counter	CX

Q → A RAPID RECALL (CONTD.)

Flag controlling string direction	DF
Unit containing the queue	BIU
Max-mode bus controller	8288
IVT location	00000H–003FFH

Q → A RAPID RECALL (CONTD.)

Vector address of INT 10H	00040H
Type number of NMI	2
Directive defining a word	DW
DIV BL quotient goes to	AL

TRAP EXAM TRAPS

- Segment registers cannot be loaded with immediate data (`MOV DS, 2000H` is illegal)
- MOV memory, memory is not allowed
- BP-based addressing uses **SS** by default, not DS
- OF (signed) and CF (unsigned) overflow are different
- 8088 is internally 16-bit but has an 8-bit external data bus