Org: 80x86 Microprocessor

Alex Chi

Update: March 24, 2020

Contents

1 Evolution of 80x86 family

- 8086, born in 1978
  - 16-bit microprocessor
  - 20-bit address data bus
  - first pipelined micro-processor
- 8088
  - 16-bit internal, 8-bit external data bus
  - fit in 8-bit world
- 80286, 80386, 80486
  - real/protected modes
  - virtual memory

2 Internal Structure of 8086

- bus interface unit: access memory and peripherals
- execution unit: executes instructions previously fetched
- refer to slides

2.1 BIU

- 16-bit segment registers CS, DS, ES, SS

- 16-bit instruction pointer IP
- 20-bit address adder: $CS \times 16 + IP$

2.2 EU

- 16-bit GP: AX, BX, CX, DX
  - and AH, AL; BH, BL; . . .
- 16-bit pointer registers: SP, BP
- 16-bit index registers: SI, DI
- flag register 9-bit (of 16-bit)
- ALU

2.3 Pipelining

- BIU pre-fetch instruction when there's more than 2 empty bytes
- sequential instruction execution
- branch penalty

2.4 8086/8088 Pins

- 8088 only has 8 A/D pins
- NMI - non-maskable interrupt
- INTR - interrupt
- CLK - clock
- BHE - bus high enable
- MN/MX - maximum, minimum
- RD - read
- WR - write (8086 doesn't use R/W)
- M/IO
- DT/R - transmit / receive
- DEN - data enable
- ALE - address latch enable
- HOLD / HLDA - hold and hold ack used for stealing cycle
- INTA - interrupt ack
- READY - I/O ready
- no BHE for 8088

2.5 Minimum Mode Configuration

- minimum mode: single CPU
- maximum mide: multiple co-CPU (8087, 8288)
- data bus: 8286 (data transceiver), bridge and provide power
- address bus: 8282 (latch)
- 8284 (clock)

3 Logical & Physical Address

- physical address
- logical address
- `CS:IP`
- translation
  - shift segment value left 4 bits, and add above value
  - logical 2500:0000 - 2500:FFFF
  - physical 25000 - 34FFF
  - wrap around if larger than FFFFF
  - physical → logical: not 1-1
  - segment overlapping: that's OK, might be desirable

3.1 Code Segment

- code address can only use `CS:IP`
- `CS:IP` cannot be changed with `mov`
- use `jump`, `call` procedure

3.2 Data Segment

- `DS:offset`
- offset can be register or immediate
- register is limited to `BX`, `SI`, `DI`
- change `DS` if data is beyond scope

3.3 Data Representation in Memory

- little / big endian
- Intel: little endian
- IBM: big endian

3.4 Stack Segment

- `SS:SP` or `SS:BP`
- most registers (except segment and SP) can be pushed into and popped from stack
- stack grows downward (upper address to lower address)
- `push ax`, `pop ax`

3.5 Extra Segment

- `ES:offset` or `ES:register(3)`
- essential for string operations

3.6 Memory map of IBM PC

- `00000-9FFFFF` RAM
  - 64K-256K
  - MS-DOS and user application
  - DOS does memory management
- `A0000~BFFFF` Video Display RAM
- `C0000~FFFFF` ROM
  - 64K BIOS
  - adapter cards

3.7 BIOS

- test all devices
- load DOS from disk
- hand over control to DOS
- BIOS is located at `FFFF0` (CS is set to `FFFF` on reset)

3.8 Flag Register (PSW)

- CF, PF, AF, ZF, SF, OF (conditional flags)
- carry, parity, auxiliary carry (3-4 carry), zero, sign, overflow
- DF, IF, TF (control flags)
- direction of string operations, interrupt (enable or disable maskable interrupt), trap (1-by-1 instruction debug)

3.9 8086 Addressing Modes

- 7 modes
- MOV dest, src
- register addressing mode `MOV BX, DX`, (except CS and IP)
- immediate addressing mode `MOV AX, 2550H`, `MOV CX, 625`, cannot be moved into segment registers
- direct addressing mode `MOV DL, [2400H]` (DS:2400H, 1 byte); `MOV [3518H], AL`; `MOV DX, WORD PTR [2400H]`
- register indirect addressing mode `MOV AL, [BX]`
- base relative addressing mode `MOV CX, [BX]+10` (DS:BX or SS:BP + C)
- indexed relative addressing mode `MOV DX, [SI]+5` (DS:DI or DS:SI + C)
- based indexed relative addressing mode `MOV CL, [BX][DI] + 8` (DS:BX or SS:BP + SI/DI + C)

3.10 Segment Overrides

- default
  - CS:IP
  - DS:[SI,DI,BX]
  - SS:[SP,BP]
- `MOV AX, CS:[BP]`