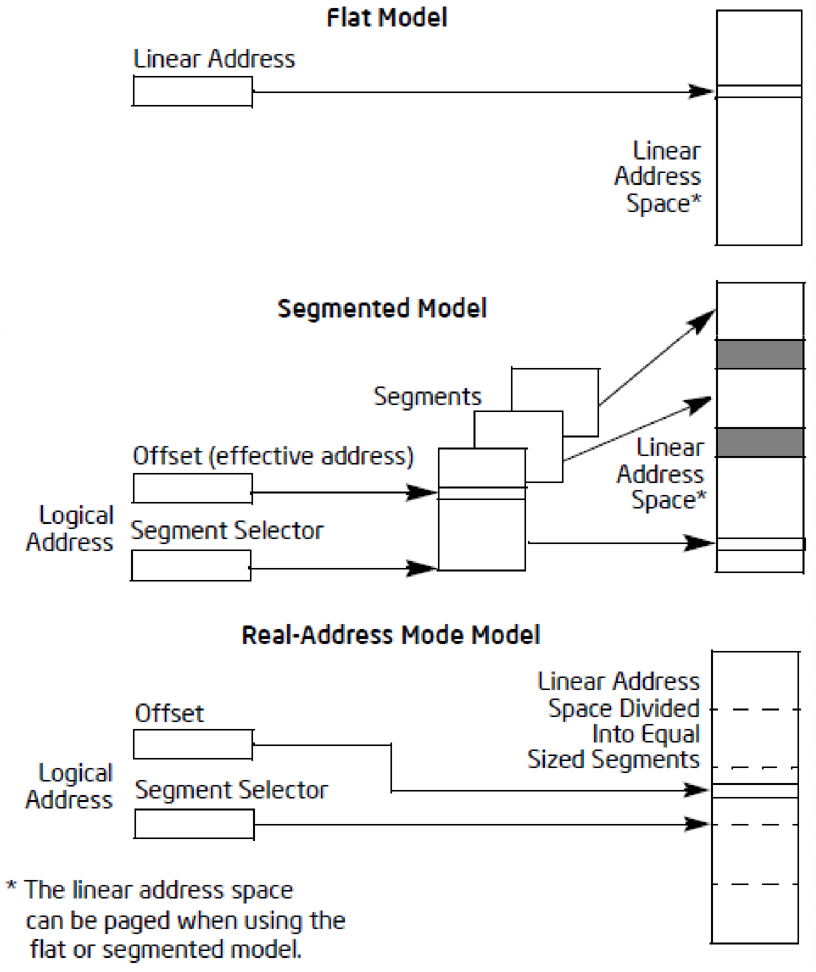
# EC325 Microprocessors IA32,IA32e Environment

Yasser F. O. Mohammad

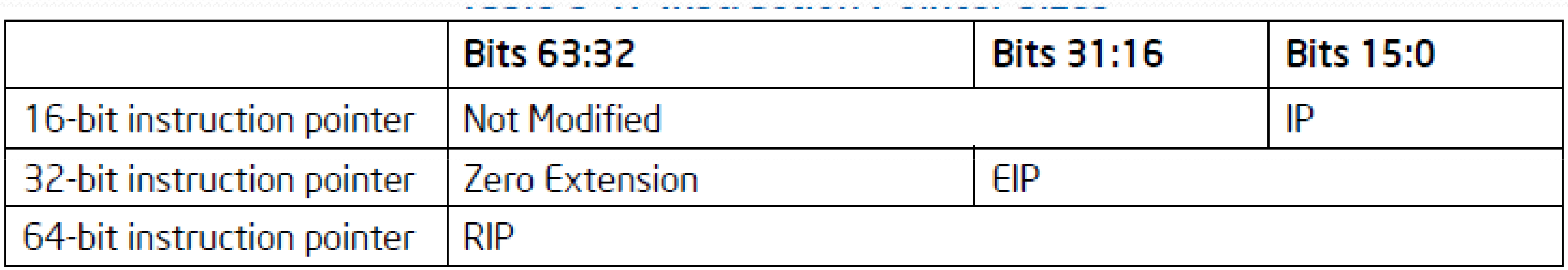
# REMINDER 1:IA32 Memory Models



# Operating and Memory Mode

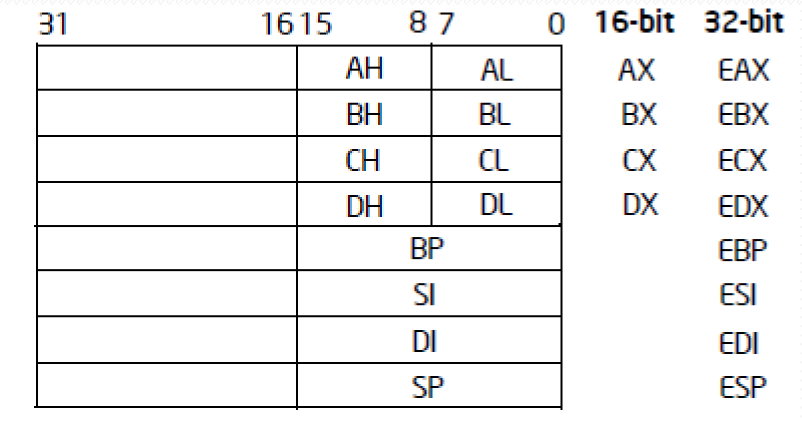
- Protected
  - All available (32 bits)
- Real-Address Mode
  - Only Flat with 16 Bit Registers
- SMM
  - Separate Read Address Space
- Compatibility Mode
  - All available (32 bits)
- 64-bit
  - Segmentation is usually Disabled

# IP in different modes

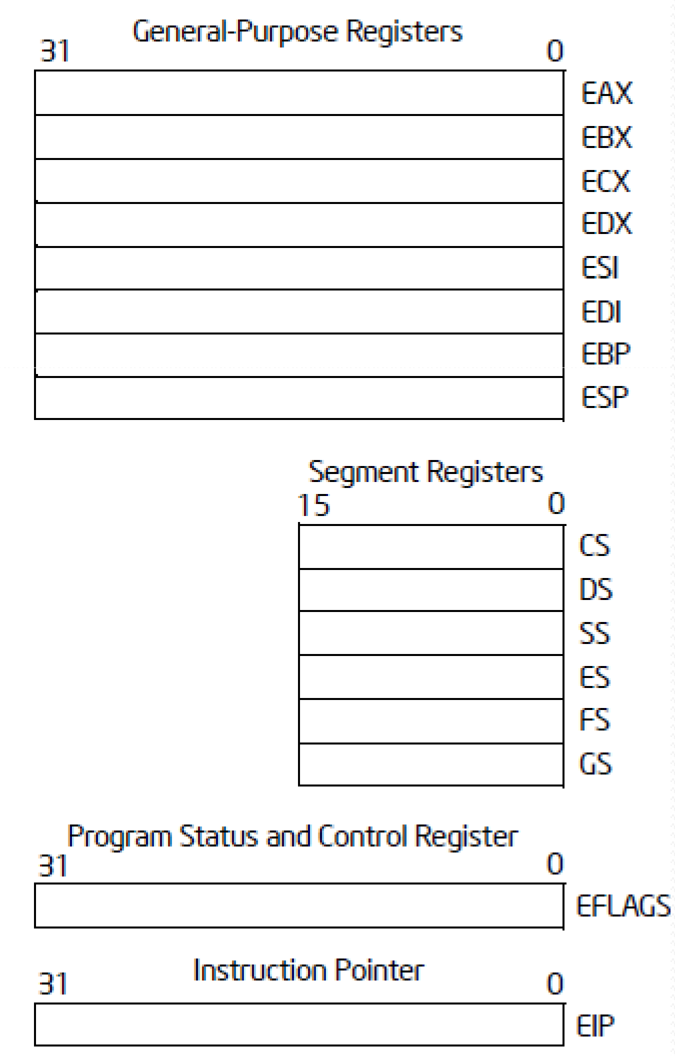


| | Bits 63:32 | Bits 31:16 | Bits 15:0 |
|---|---|---|---|
| 16-bit instruction pointer | Not Modified | | IP |
| 32-bit instruction pointer | Zero Extension | EIP | |
| 64-bit instruction pointer | RIP | | |

# GPRs (32 bits)



| 31 16 | 15 8 | 7 0 | 16-bit | 32-bit |
|---|---|---|---|---|
| | AH | AL | AX | EAX |
| | BH | BL | BX | EBX |
| | CH | CL | CX | ECX |
| | DH | DL | DX | EDX |
| | BP | | | EBP |
| | SI | | | ESI |
| | DI | | | EDI |
| | SP | | | ESP |



General-Purpose Registers

31 0

EAX<br>
EBX<br>
ECX<br>
EDX<br>
ESI<br>
EDI<br>
EBP<br>
ESP

Segment Registers

15 0

CS<br>
DS<br>
SS<br>
ES<br>
FS<br>
GS

Program Status and Control Register

31 0

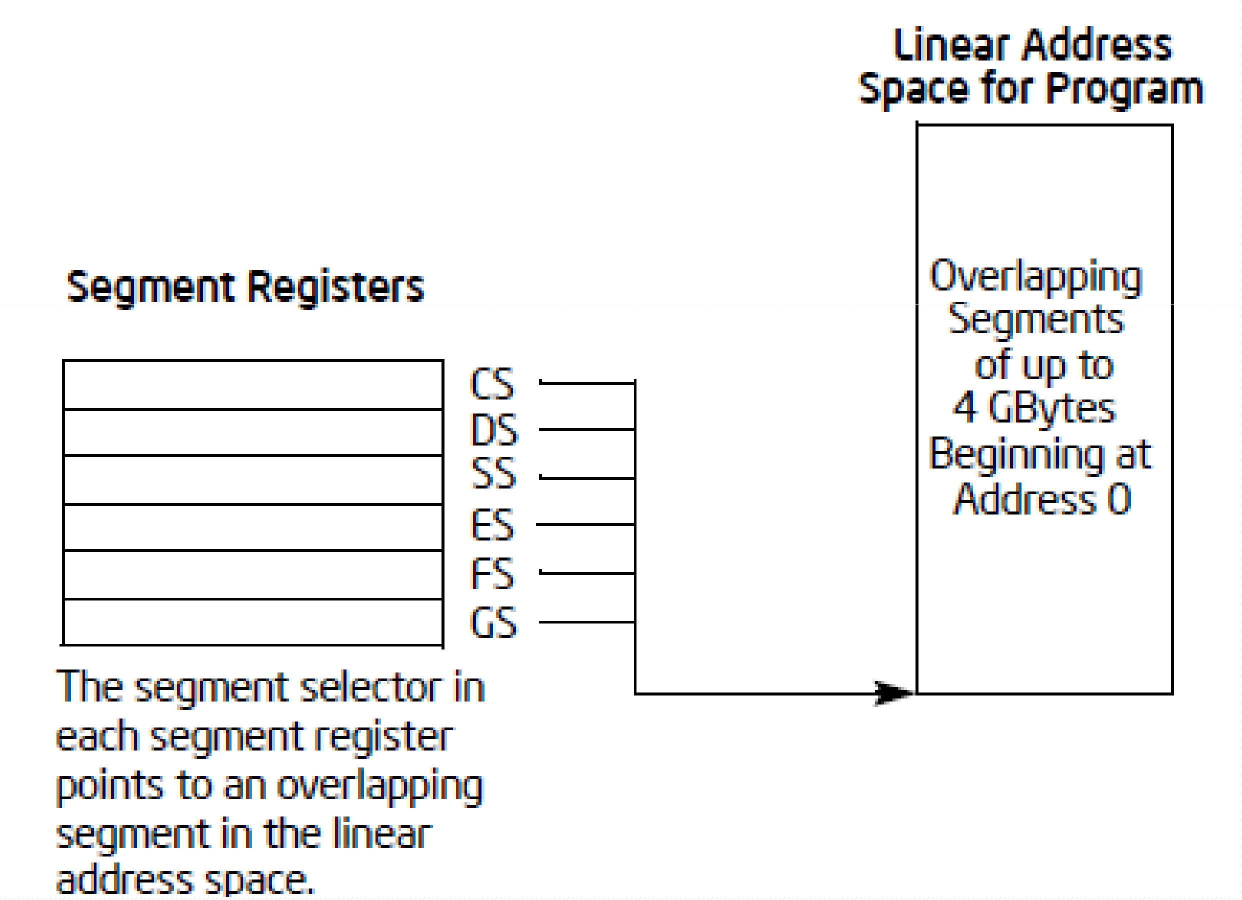
EFLAGS

Instruction Pointer

31 0

EIP

# Use of Segment Registers
# Flat Memory



Linear Address Space for Program

Segment Registers

CS

DS

SS

ES

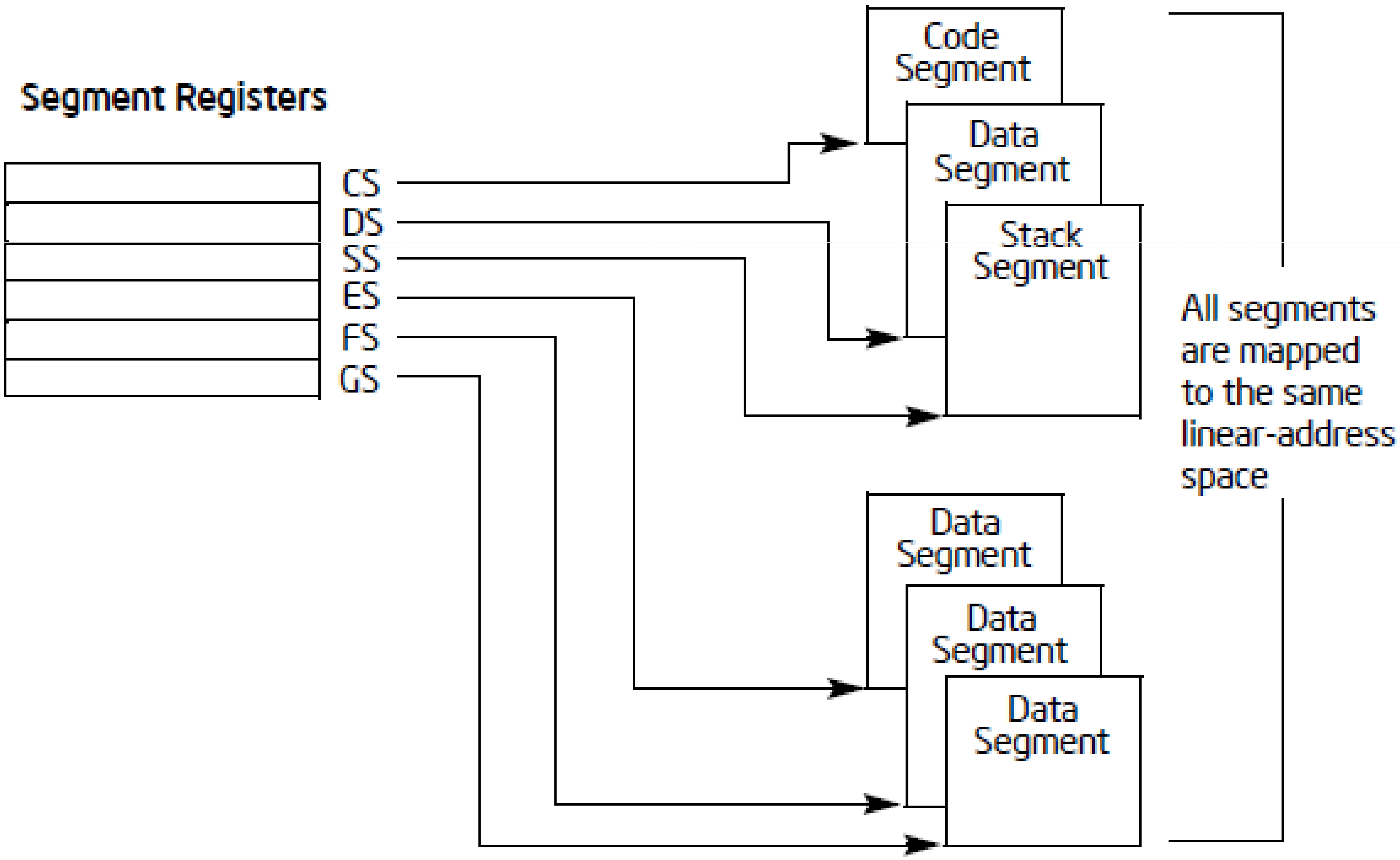
FS

GS

Overlapping Segments of up to 4 GBytes Beginning at Address 0

The segment selector in each segment register points to an overlapping segment in the linear address space.

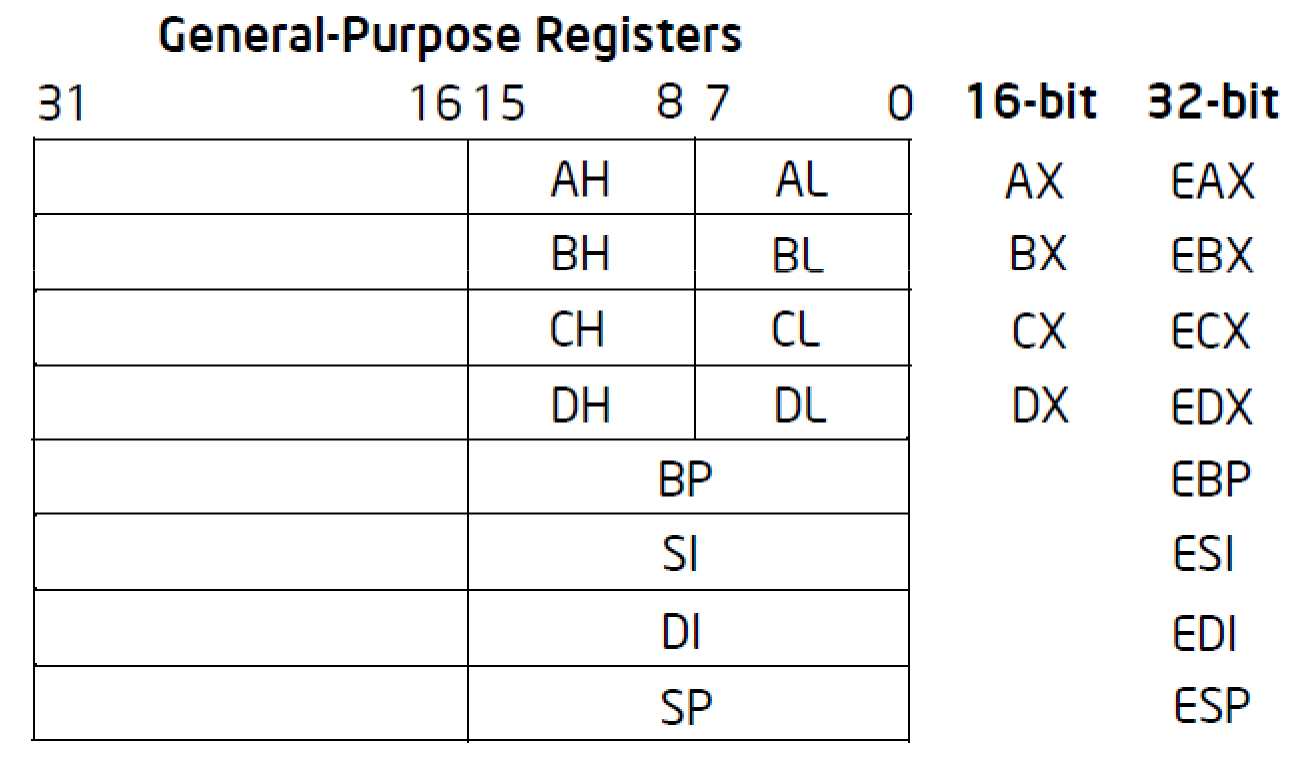
# Use of Segment Registers Segmented Mode



# Use of GPRs

- **EAX** — Accumulator for operands and results data
- **EBX** — Pointer to data in the DS segment
- **ECX** — Counter for string and loop operations
- **EDX** — I/O pointer
- **ESI** — Pointer to data in the segment pointed to by the DS register; source pointer for string operations
- **EDI** — Pointer to data (or destination) in the segment pointed to by the ES register; destination pointer for string operations
- **ESP** — Stack pointer (in the SS segment)

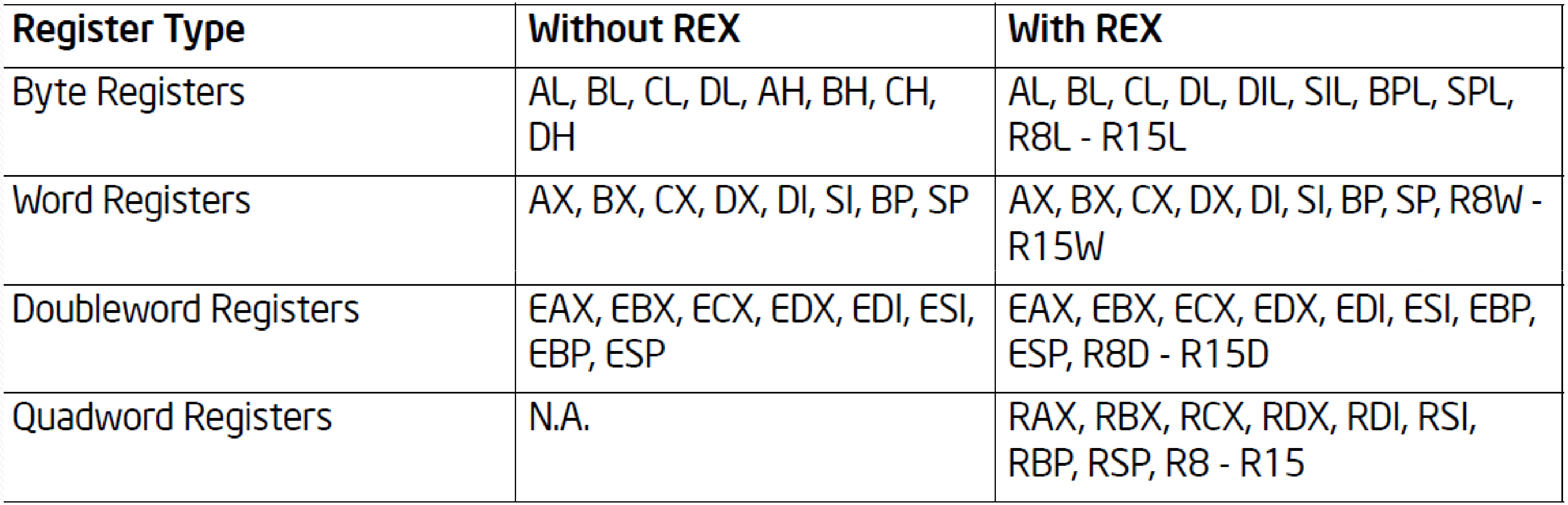
# GRP names



General-Purpose Registers

| 31 16 | 15 8 | 7 0 | 16-bit | 32-bit |
|---|---|---|---|---|
| | AH | AL | AX | EAX |
| | BH | BL | BX | EBX |
| | CH | CL | CX | ECX |
| | DH | DL | DX | EDX |
| | BP | | | EBP |
| | SI | | | ESI |
| | DI | | | EDI |
| | SP | | | ESP |

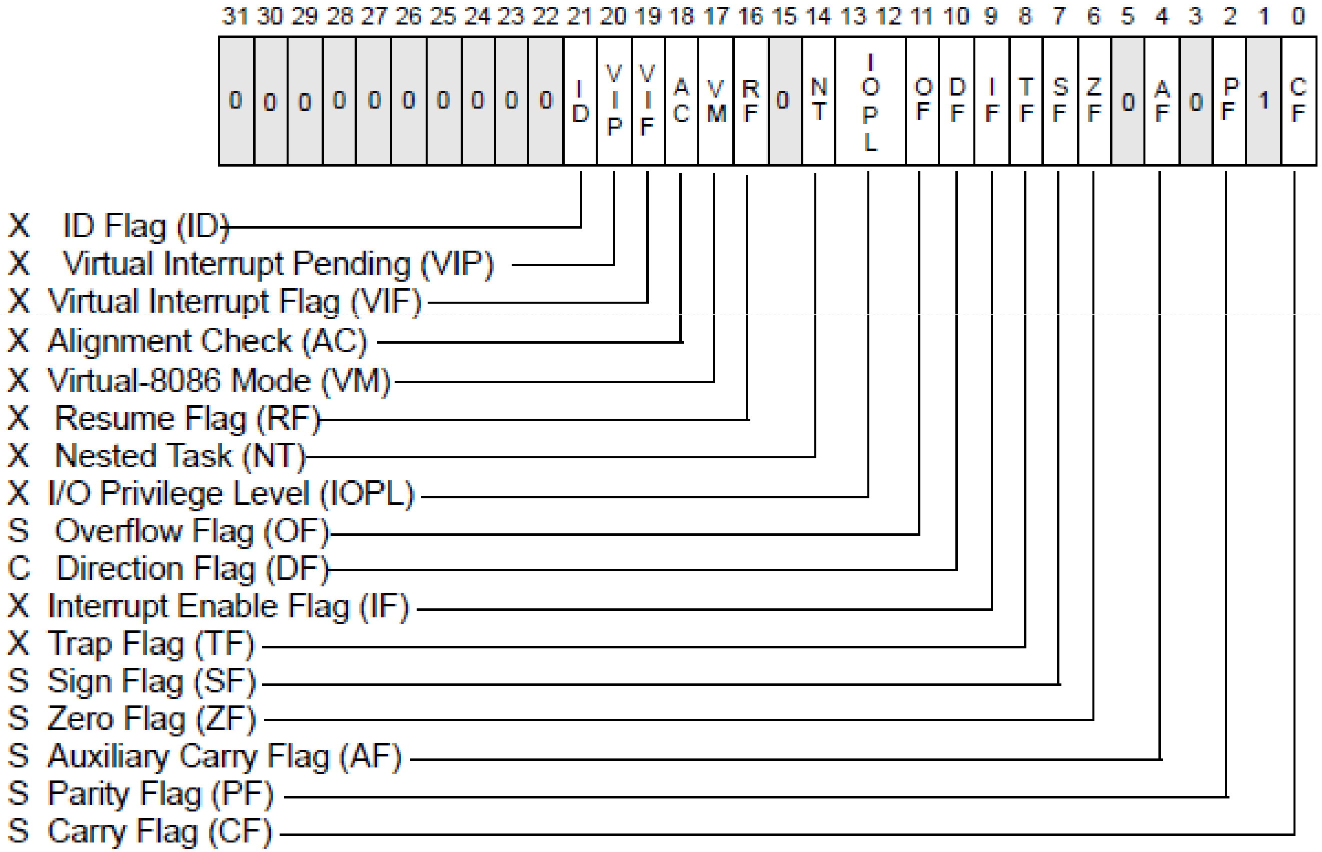
# 64-bit registers



| Register Type | Without REX | With REX |
|---|---|---|
| Byte Registers | AL, BL, CL, DL, AH, BH, CH, DH | AL, BL, CL, DL, DIL, SIL, BPL, SPL, R8L - R15L |
| Word Registers | AX, BX, CX, DX, DI, SI, BP, SP | AX, BX, CX, DX, DI, SI, BP, SP, R8W - R15W |
| Doubleword Registers | EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP | EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP, R8D - R15D |
| Quadword Registers | N.A. | RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, R8 - R15 |

- Same instruction cannot access a legacy high byte and a new byte register
- Operations in parts of registers keep the rest of them untouched

# EFLAGS



| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ID | VIP | VIF | AC | VM | RF | 0 | NT | IOPL |  | OF | DF | IF | TF | SF | ZF | 0 | AF | 0 | PF | 1 | CF |

X ID Flag (ID)
X Virtual Interrupt Pending (VIP)
X Virtual Interrupt Flag (VIF)
X Alignment Check (AC)
X Virtual-8086 Mode (VM)
X Resume Flag (RF)
X Nested Task (NT)
X I/O Privilege Level (IOPL)
S Overflow Flag (OF)
C Direction Flag (DF)
X Interrupt Enable Flag (IF)
X Trap Flag (TF)
S Sign Flag (SF)
S Zero Flag (ZF)
S Auxiliary Carry Flag (AF)
S Parity Flag (PF)
S Carry Flag (CF)

S Indicates a Status Flag
C Indicates a Control Flag
X Indicates a System Flag

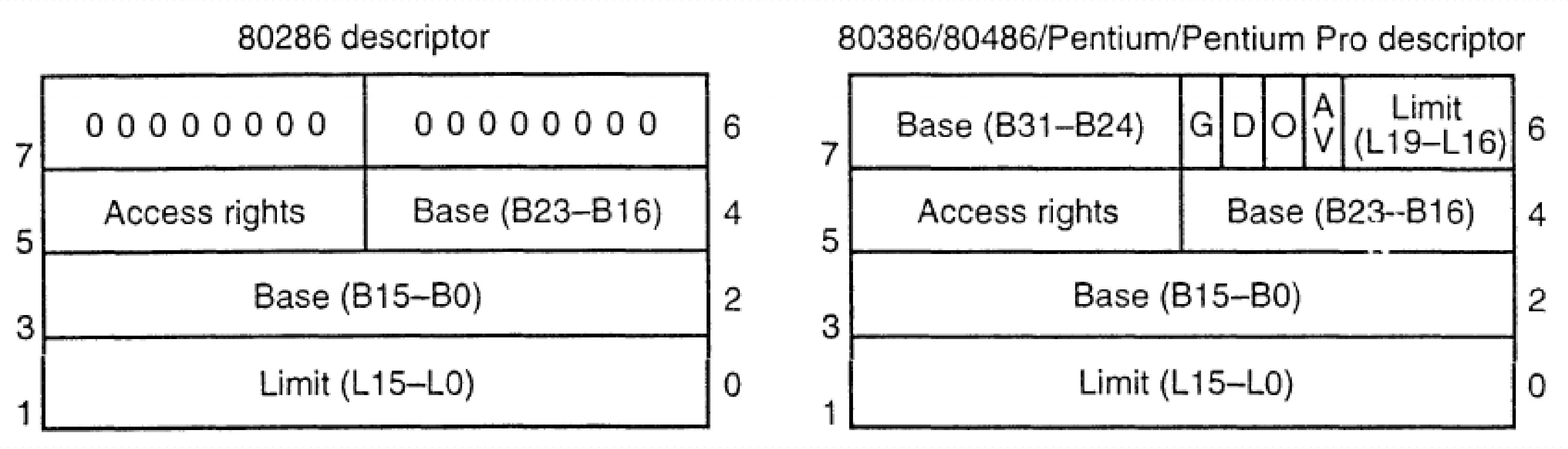
# Flag Meaning

- CF: Set if there is a carry
- PF: Set if least significant byte of result has even ones
- AF: Set if an arithmetic operation generates carry/borrow in bit 3 (BCD)
- ZF: Set if zero
- SF: Set if MSB is 1
- OF: Set on overflow (2's)
- DF: Direction in string operations
- TF: Single step during debugging
- IF: Interrupt enable
- IOPL: privilege level
- NT: Nested task
- RF: Used by debugger
- VM: Virtual 8086 mode
- AC: Alignment Checks
- ID: can I ask your name??

# Protected Mode Addressing

- Segment registers store selectors rather than base address (bases are not really bases!!!!)
- 8K global descriptors
- 8K local descriptors
- Descriptor = 8 bytes

# Descriptor Format



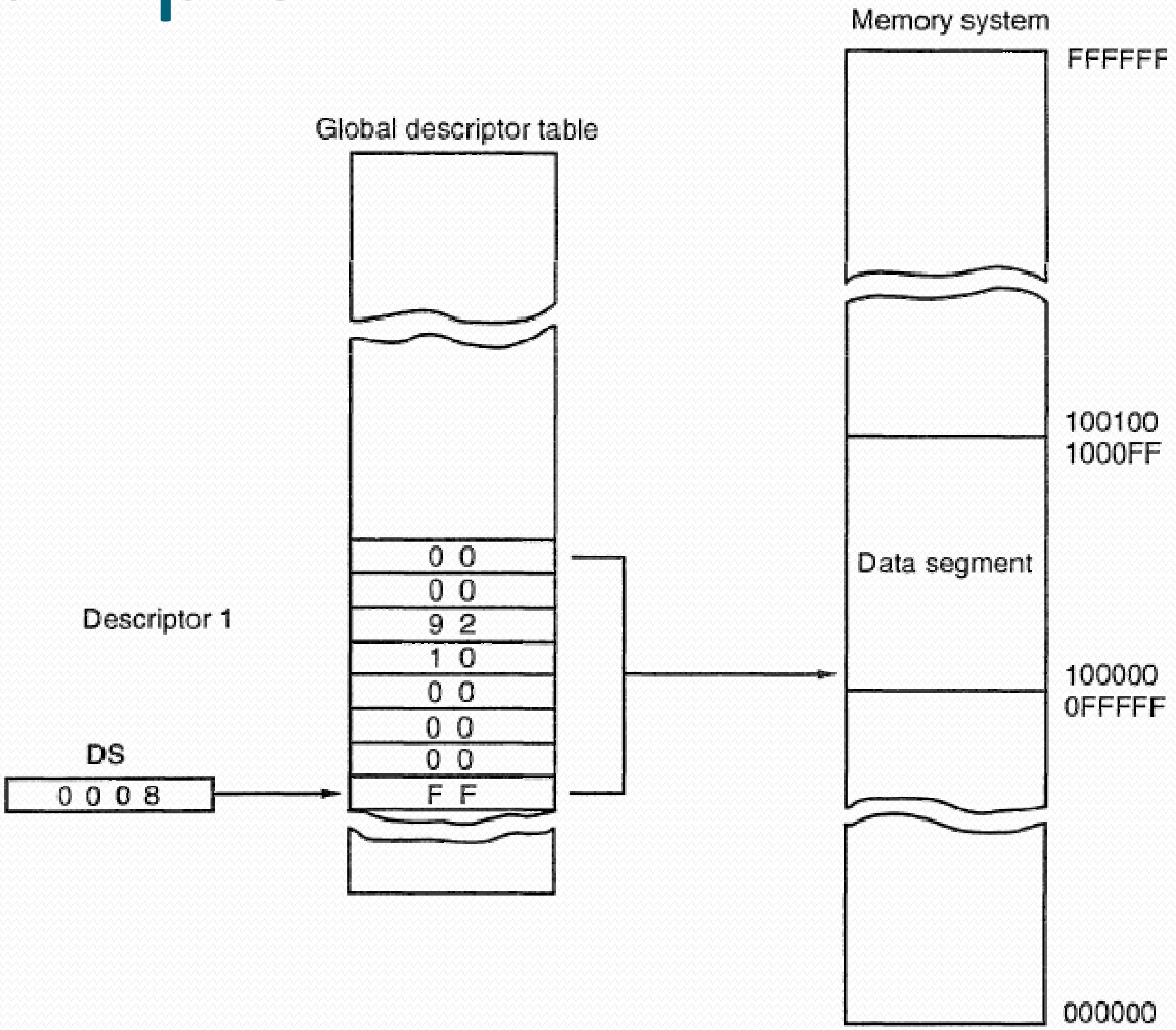
80286 descriptor

| | | | |
|---|---|---|---|
| 7 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 6 |
| 5 | Access rights | Base (B23–B16) | 4 |
| 3 | Base (B15–B0) | | 2 |
| 1 | Limit (L15–L0) | | 0 |

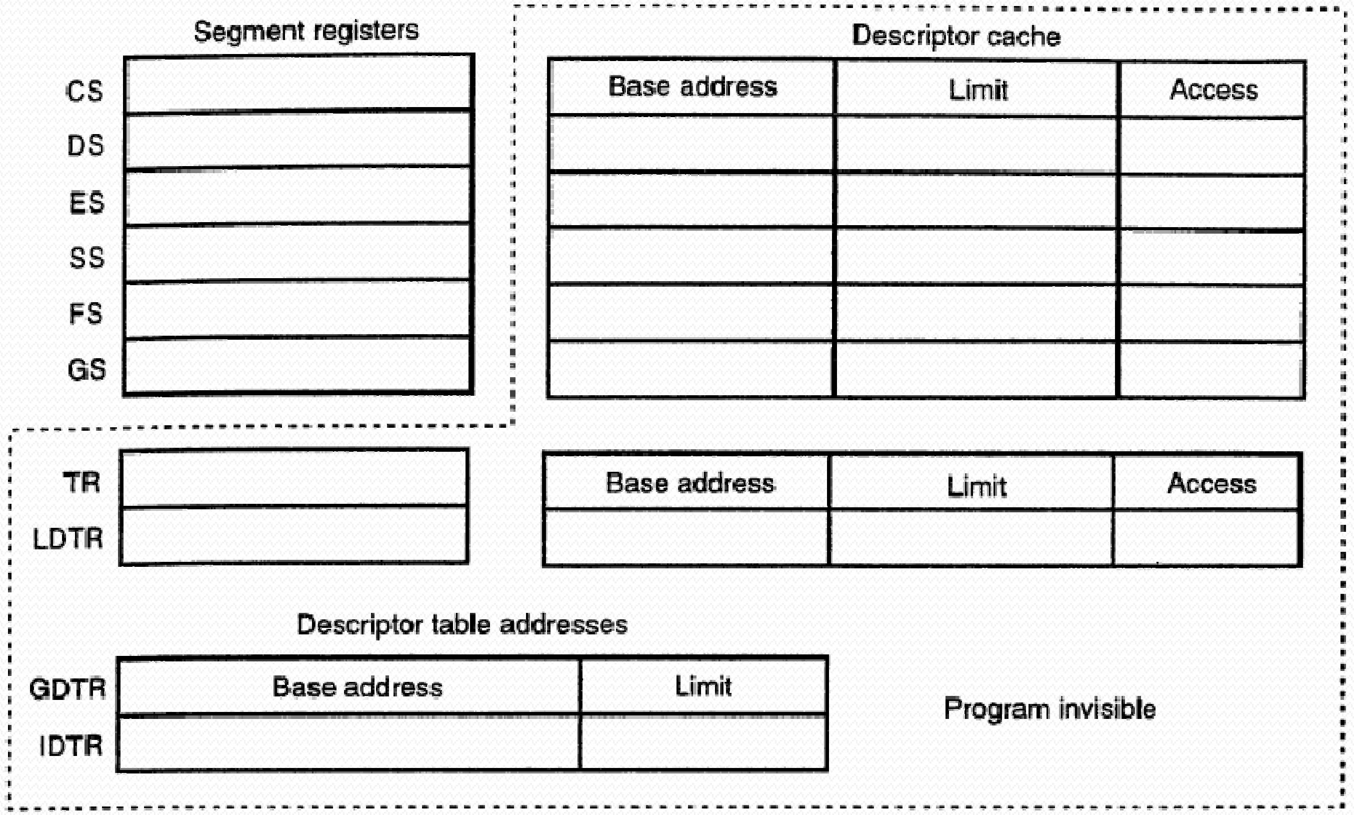
80386/80486/Pentium/Pentium Pro descriptor

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | Base (B31–B24) | G | D | O | A<br>V | Limit<br>(L19–L16) | 6 |
| 5 | Access rights | Base (B23--B16) | | | | | 4 |
| 3 | Base (B15–B0) | | | | | | 2 |
| 1 | Limit (L15–L0) | | | | | | 0 |

# Example


Memory system
FFFFFF
Global descriptor table
100100
1000FF
0 0
0 0
9 2
1 0
0 0
0 0
0 0
F F
Descriptor 1
Data segment
100000
0FFFFF
DS
0 0 0 8
000000

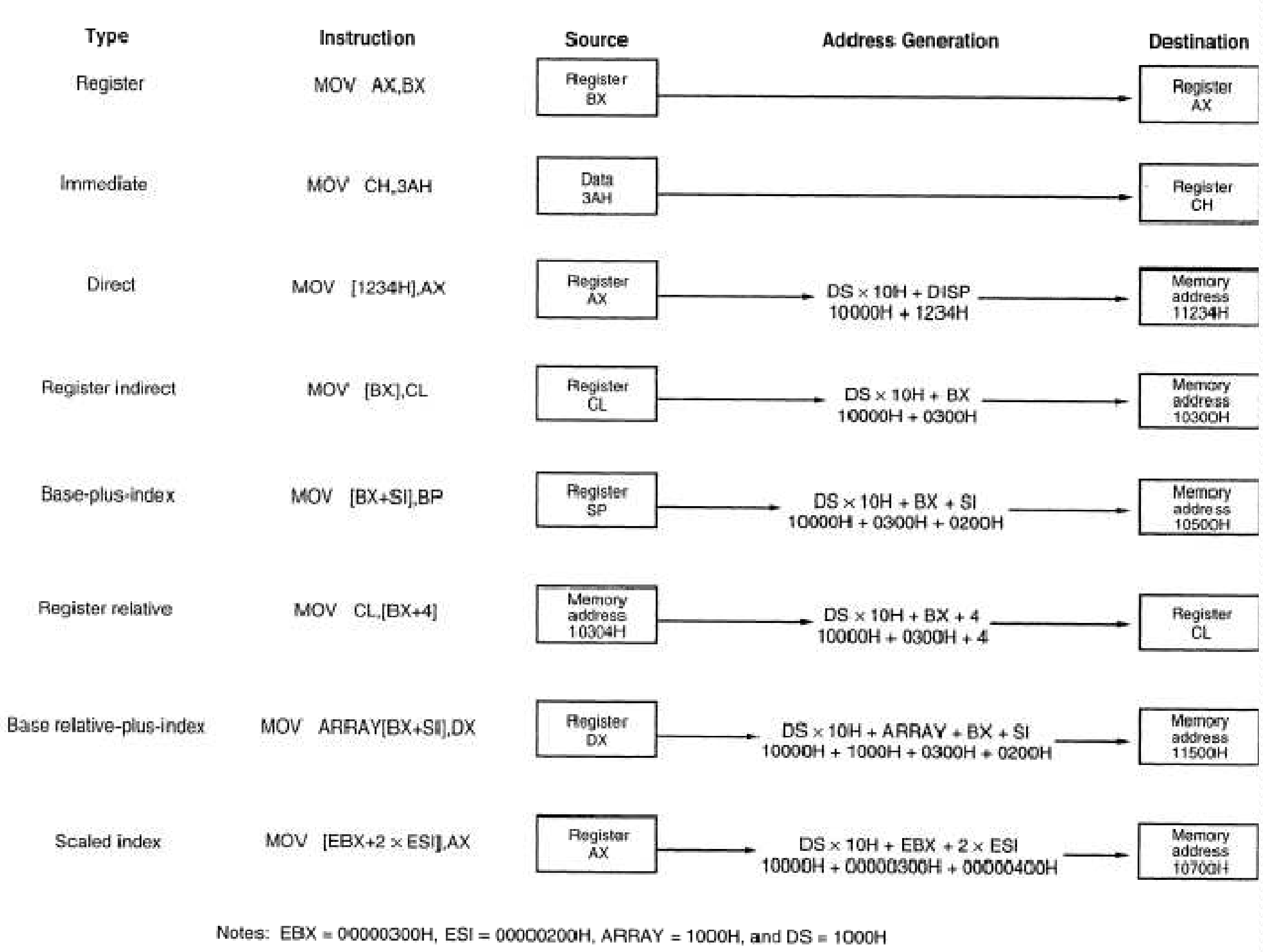
# Program Invisible Registers



Notes:

1. The 80286 does not contain FS and GS nor the program-invisible portions of these registers.
2. The 80286 contains a base address that is 24-bits and a limit that is 16-bits.
3. The 80386/80486/Pentium/Pentium Pro contain a base address that is 32-bits and a limit that is 20-bits.
4. The access rights are 8-bits in the 80286 and 12-bits in the 80386/80486/Pentium.

# Addressing Modes



| Type | Instruction | Source | Address Generation | Destination |
|---|---|---|---|---|
| Register | MOV AX,BX | Register BX | | Register AX |
| Immediate | MOV CH,3AH | Data 3AH | | Register CH |
| Direct | MOV [1234H],AX | Register AX | DS × 10H + DISP<br>10000H + 1234H | Memory address 11234H |
| Register indirect | MOV [BX],CL | Register CL | DS × 10H + BX<br>10000H + 0300H | Memory address 10300H |
| Base-plus-index | MOV [BX+SI],BP | Register SP | DS × 10H + BX + SI<br>10000H + 0300H + 0200H | Memory address 10500H |
| Register relative | MOV CL,[BX+4] | Memory address 10304H | DS × 10H + BX + 4<br>10000H + 0300H + 4 | Register CL |
| Base relative-plus-index | MOV ARRAY[BX+SI],DX | Register DX | DS × 10H + ARRAY + BX + SI<br>10000H + 1000H + 0300H + 0200H | Memory address 11500H |
| Scaled index | MOV [EBX+2 × ESI],AX | Register AX | DS × 10H + EBX + 2 × ESI<br>10000H + 00000300H + 00000400H | Memory address 10700H |

Notes: EBX = 00000300H, ESI = 00000200H, ARRAY = 1000H, and DS = 1000H