IA-32 Addressing Modes

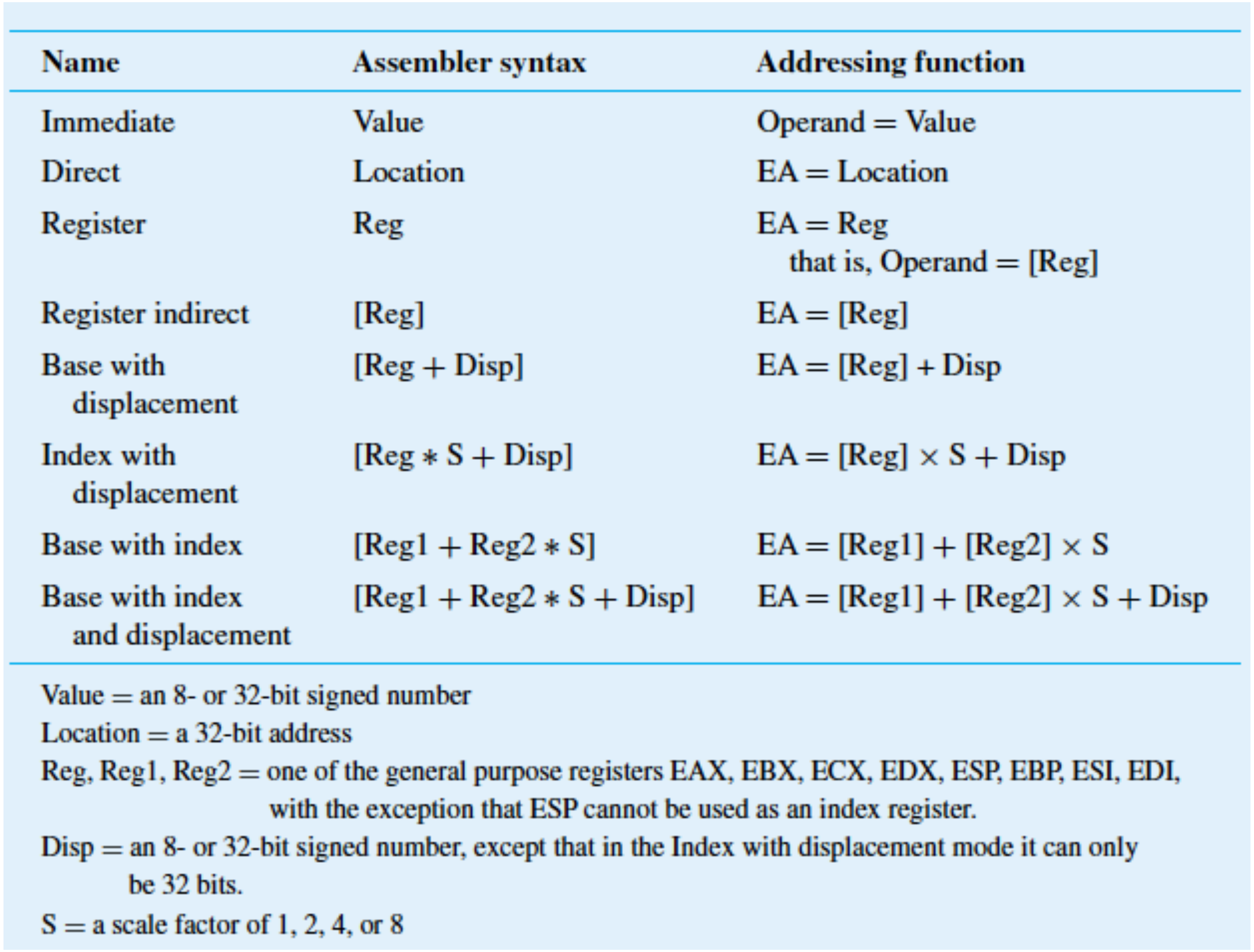
- IA-32 addressing modes are CISC like
  - Large number of flexible addressing modes
    - Register Mode – operands are in registers
    - Immediate Mode – instruction contains operand
    - Direct mode – instruction contains operand address
    - Register indirect – operand address in register
    - Base with displacement – [reg]+displacement = op adrs
    - Index with displacement – [reg]*scale + disp. = op adrs
    - Base with index – op adrs = [breg] + [ireg]*scale
    - Base with index & displacement
      - op adrs = [base_reg] + [index_reg]*scale + displacement

JOHNS HOPKINS UNIVERSITY | Engineering for Professionals

IA-32 Addressing Modes

- IA-32 addressing modes are CISC like
  - Large number of flexible addressing modes
    - Register Mode – operands are in registers
    - Immediate Mode – instruction contains operand
    - Direct mode – instruction contains operand address
    - Register indirect – operand address in register
    - Base with displacement – [reg]+displacement = op adrs
    - Index with displacement – [reg]*scale + disp. = op adrs
    - Base with index – op adrs = [breg] + [ireg]*scale
    - Base with index & displacement
      - op adrs = [base_reg] + [index_reg]*scale + displacement

IA-32 Addressing Modes



Name	Assembler syntax	Addressing function
Immediate	Value	Operand = Value
Direct	Location	EA = Location
Register	Reg	EA = Reg that is, Operand = [Reg]
Register indirect	[Reg]	EA = [Reg]
Base with displacement	[Reg + Disp]	EA = [Reg] + Disp
Index with displacement	[Reg $*$ S + Disp]	EA = [Reg] $\times$ S + Disp
Base with index	[Reg1 + Reg2 $*$ S]	EA = [Reg1] + [Reg2] $\times$ S
Base with index and displacement	[Reg1 + Reg2 $*$ S + Disp]	EA = [Reg1] + [Reg2] $\times$ S + Disp

Value = an 8- or 32-bit signed number

Location = a 32-bit address

Reg, Reg1, Reg2 = one of the general purpose registers EAX, EBX, ECX, EDX, ESP, EBP, ESI, EDI, with the exception that ESP cannot be used as an index register.

Disp = an 8- or 32-bit signed number, except that in the Index with displacement mode it can only be 32 bits.

S = a scale factor of 1, 2, 4, or 8

IA-32 Addressing Modes

- Instructions can have 0, 1 or 2 operands
  - Two-operand syntax: OP destination,source
  - Examples based on MOV instruction:
    - MOV EBP,EAX copies EAX reg into EBP reg
    - MOV EAX,25 copies 32-bit constant into EAX
    - MOV AX,320 copies 16-bit constant into AX
    - MOV AL,125 copies 8-bit constant into AL
    - MOV EAX,LOC1 copies 32 bits at LOC1 into EAX
    - MOV EBX, OFFSET LOC1
      - Puts address LOC1 into EBX
    - MOV EAX,[EBX]
      - EAX = 32-bit contents of location whose address is in EBX

IA-32 Addressing Modes

- Base with displacement examples
  - Assume that EBP contains 2000
  - MOV EAX,[EBP+60]
    - Copies contents of doubleword (32 bits) at 2060 into EAX
  - MOV AL,[EBP+60]
    - Copies contents of byte (8 bits) at 2060 into AL
  - MOV [EBP+67],AH
    - Copies contents of AH into byte at address 2067
  - MOV [EPB+100],28 size of constant is unclear
    - MOV BYTE PTR [EBP+67],28 for 8-bit
    - MOV WORD PTR [EBP+67],28 for 16-bit
    - MOV DWORD PTR [EBP+67],28 for 32-bit

IA-32 Addressing Modes

- Base with index & displacement example
  - Assume that [EBP] = 2000 & [ESI] = 0
  - MOV EAX,[EBP + ESI*4 + 100]
    - Copies contents of doubleword (32 bits) at 2100 into EAX
  - To copy contents of doublewords at 2100, 2104, 2108, etc. place in loop and increment ESI by 1 for each iteration
    - E.g., to step through array of 32-bit elements
    - Scale factor of 2 for 16-bit elements
    - Scale factor of 1 for 8-bit elements or characters