

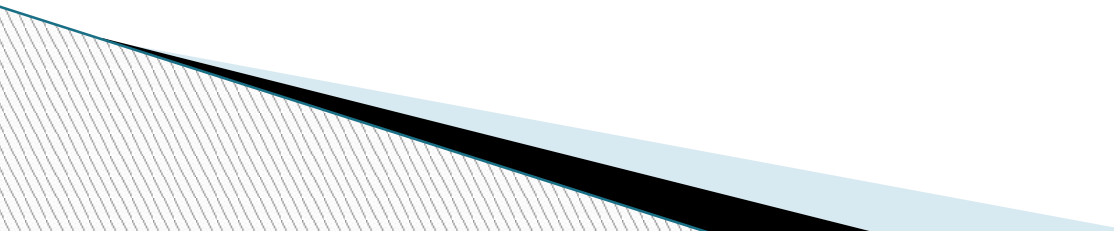
8086 Data Transfer Instruction



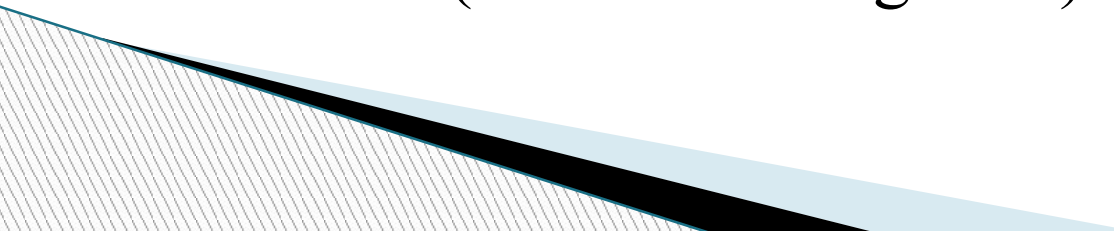
Introduction

- ▶ Data transfer instructions are those which are used to movement of the data from one location to another location such as memory to microprocessor, microprocessor to port or memory or vice versa. Data transfer instructions include such as move, Exchange etc.

Types of Instruction

- 1 Data-Transfer Instructions
 - 2 Arithmetic Instructions
 - 3 Logic Instructions
 - 4 Shift Instructions
 - 5 Rotate Instructions
- 

Data-Transfer Instructions

- ▶ The data-transfer functions provide the ability to move data either between its internal registers or between an internal register and a storage location in memory
 - ▶ The data-transfer functions include
 - MOV (Move byte or word)
 - XCHG (Exchange byte or word)
 - XLAT (Translate byte)
 - LEA (Load effective address)
 - LDS (Load data segment)
 - LES (Load extra segment)
- 

Data-Transfer Instructions (cont.)

▶ The MOVE Instruction

The move (MOV) instruction is used to transfer a byte or a word of data from a source operand to a destination operand

Mnemonic	Meaning	Format	Operation	Flags affected
MOV	Move	MOV D,S	(S) → (D)	None

e.g. MOV DX, CS

 MOV [SUM], AX

Data-Transfer Instructions (cont.)

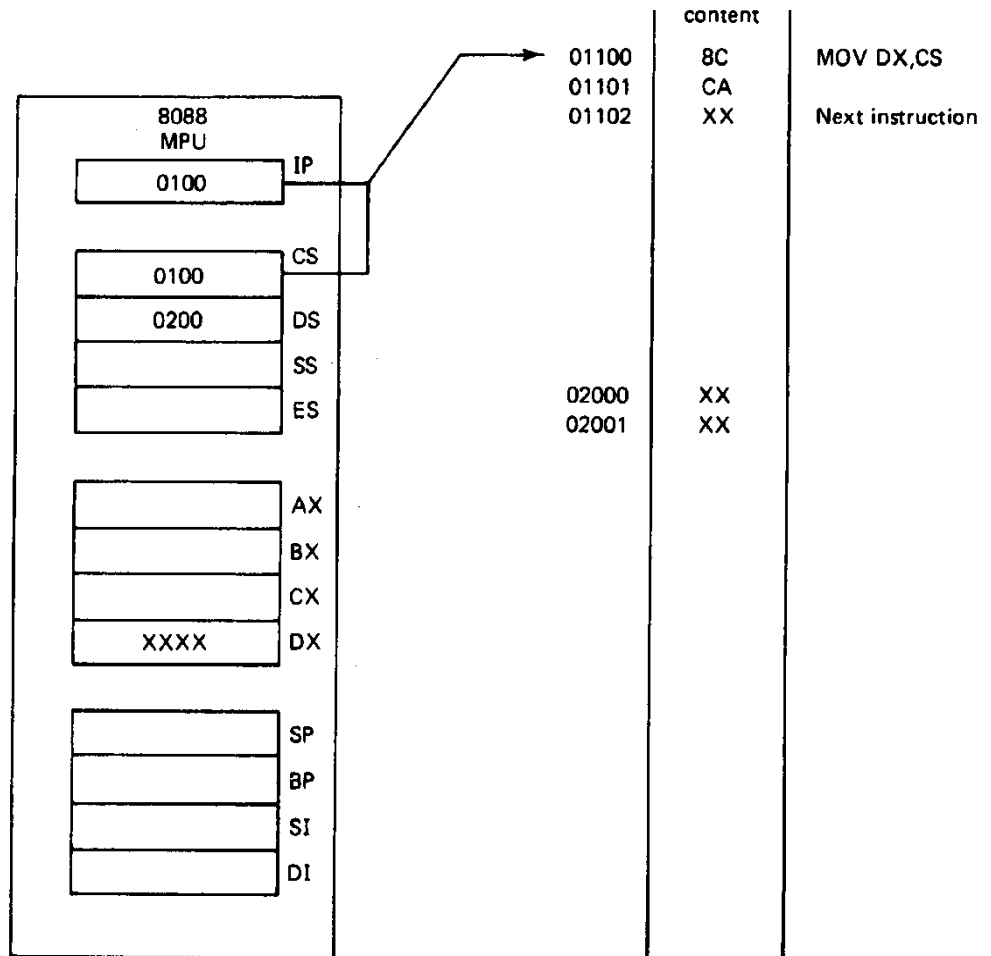
- ▶ The MOVE Instruction

Note that the MOV instruction cannot transfer data directly between external memory

Destination	Source
Memory	Accumulator
Accumulator	Memory
Register	Register
Register	Memory
Memory	Register
Register	Immediate
Memory	Immediate
Seg-reg	Reg16
Seg-reg	Mem16
Reg16	Seg-reg
Memory	Seg-reg

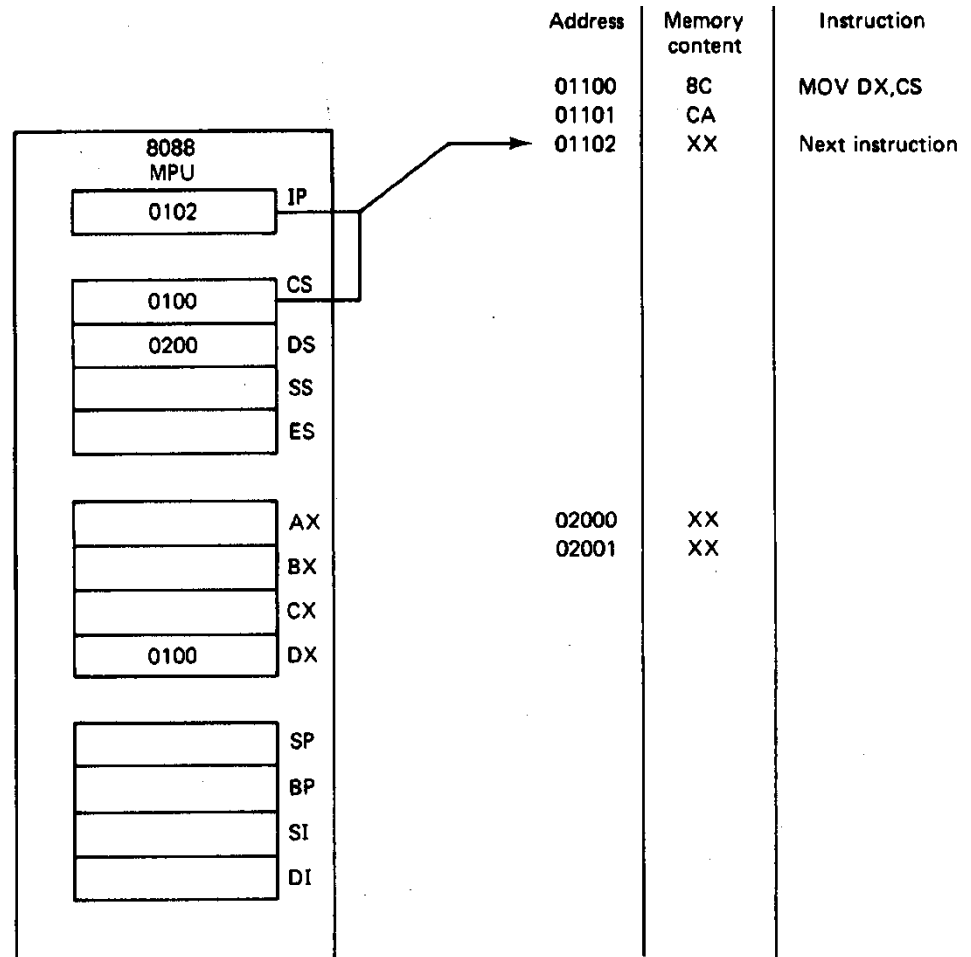
Data-Transfer Instructions (cont.)

- ▶ The MOVE Instruction
MOV DX, CS



Data-Transfer Instructions (cont.)

- ▶ The MOVE Instruction
 - MOV DX, CS



Data-Transfer Instructions (cont.)

▶ *EXAMPLE:*

What is the effect of executing the instruction?

MOV CX, [SOURCE_MEM],

where SOURCE_MEM equal to 2016 is a memory location offset relative to the current data segment starting at 1A00016

▶ *Solution:*

((DS)0+2016) (CL)

((DS)0+2016+116) (CH)

Therefore CL is loaded with the contents held at memory address

$1A00016 + 2016 = 1A02016$

and CH is loaded with the contents of memory address

$1A00016 + 2016 + 116 = 1A02116 \rightarrow$

Data-Transfer Instructions (cont.)

- ▶ The XCHG Instruction

- The exchange (XCHG) instruction can be used to swap data between two general-purpose registers or between a general purpose register and a storage location in memory

Mnemonic	Meaning	Format	Operation	Flags affected
XCHG	Exchange	XCHG D,S	(D) ↔ (S)	None

e.g. XCHG AX, DX

Destination	Source
Accumulator	Reg16
Memory	Register
Register	Register
Register	Memory

Data-Transfer Instructions (cont.)

▶ *EXAMPLE:*

What is the result of executing the following instruction?

XCHG [SUM], BX

where SUM = 123416, (DS)=120016

▶ *Solution:*

((DS)0+SUM) (BX)

PA = 1200016 + 123416 = 1323416

Execution of the instruction performs the following 16-bit swap:

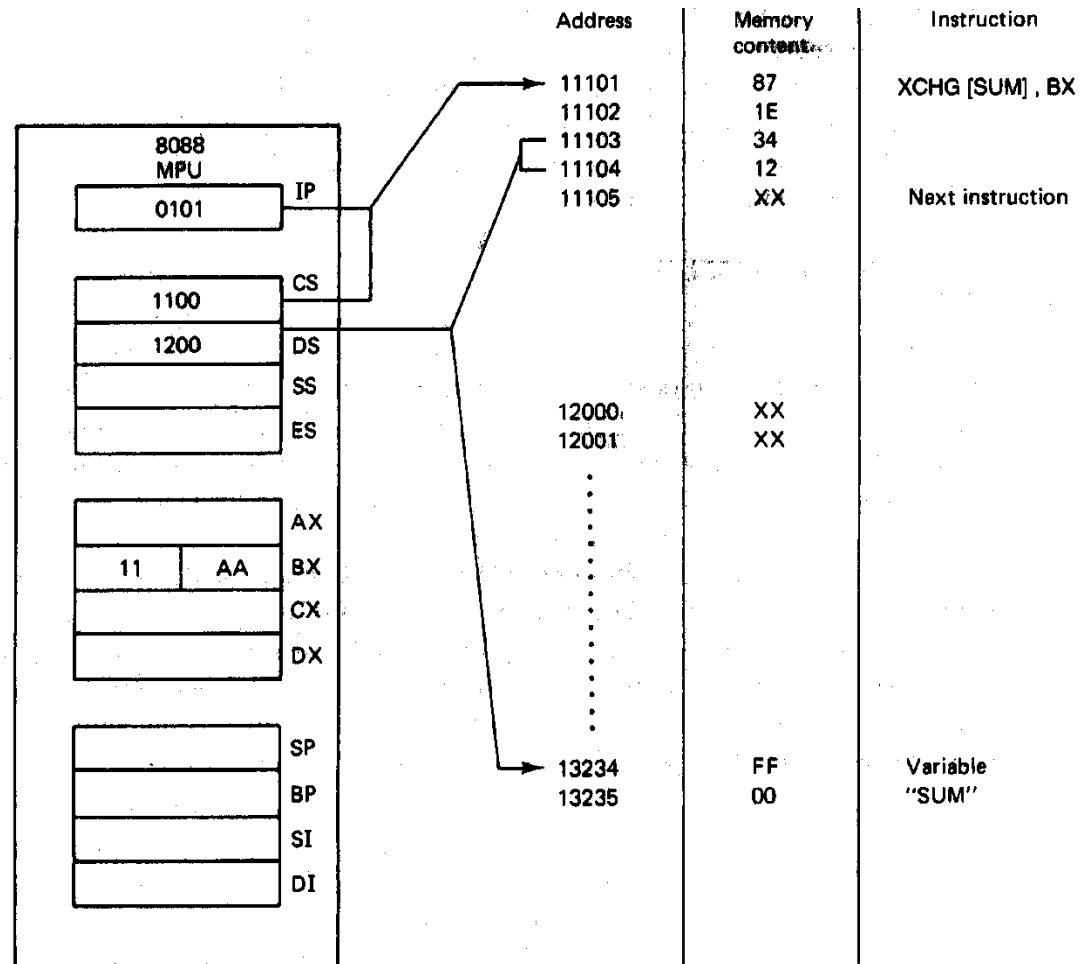
(1323416) (BL)

(1323516) (BH)

So we get (BX) = 00FF16, (SUM) = 11AA16

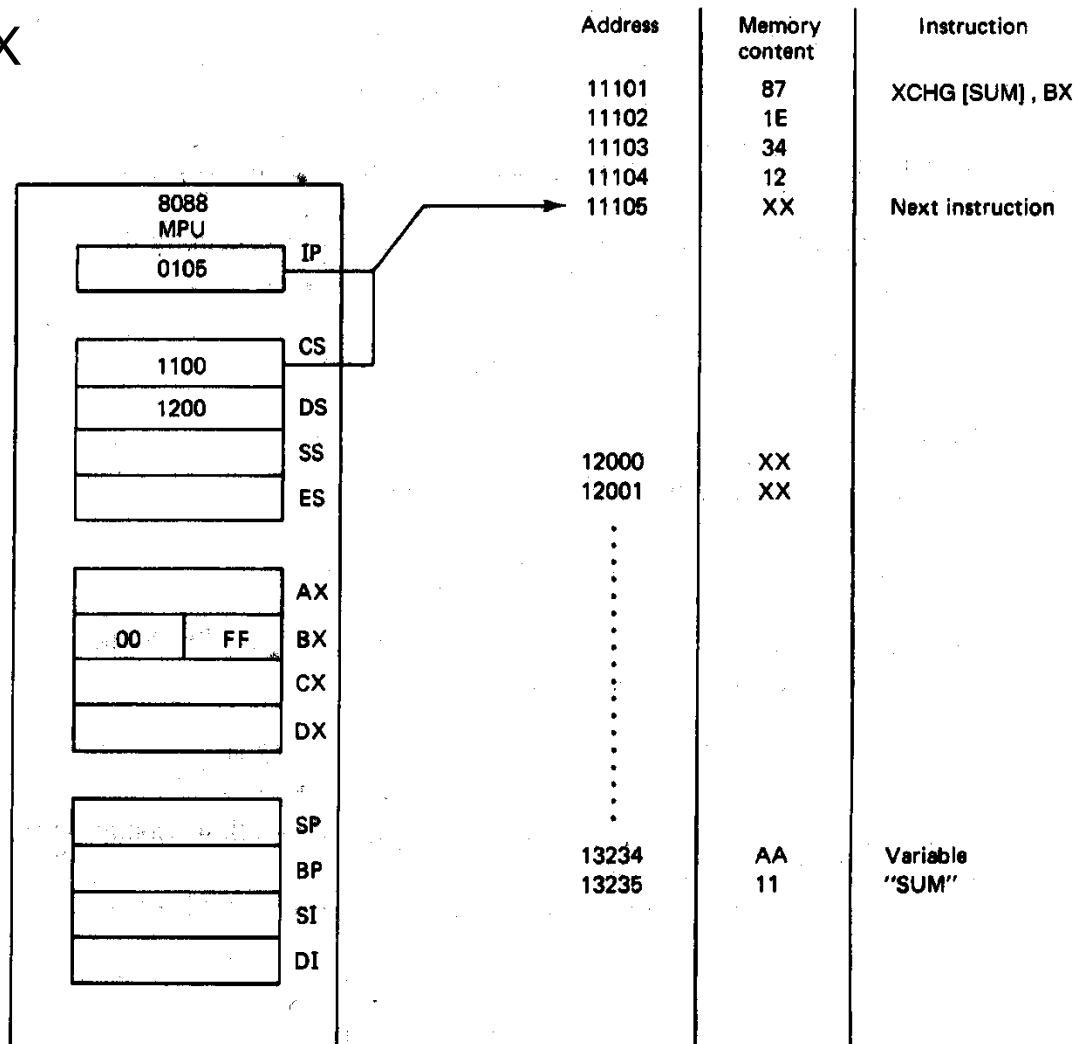
Data-Transfer Instructions (cont.)

- ▶ The XCHG Instruction
 - XCHG [SUM], BX



Data-Transfer Instructions (cont.)

- ▶ The XCHG Instruction
 - XCHG [SUM], BX



Data-Transfer Instructions (cont.)

▶ The XLAT Instruction

- The translate (XLAT) instruction is used to simplify implementation of the lookup-table operation.
 - Execution of the XLAT replaces the contents of AL by the contents of the accessed lookup-table location

Mnemonic	Meaning	Format	Operation	Flags affected
XLAT	Translate	XLAT	$((AL)+(BX)+(DS)0) \rightarrow (AL)$	None

e.g. $PA = (DS)0 + (BX) + (AL)$
 $= 0300016 + 010016 + 0D16 = 0310D16$
(0310D16) **(AL)**

Data-Transfer Instructions (cont.)

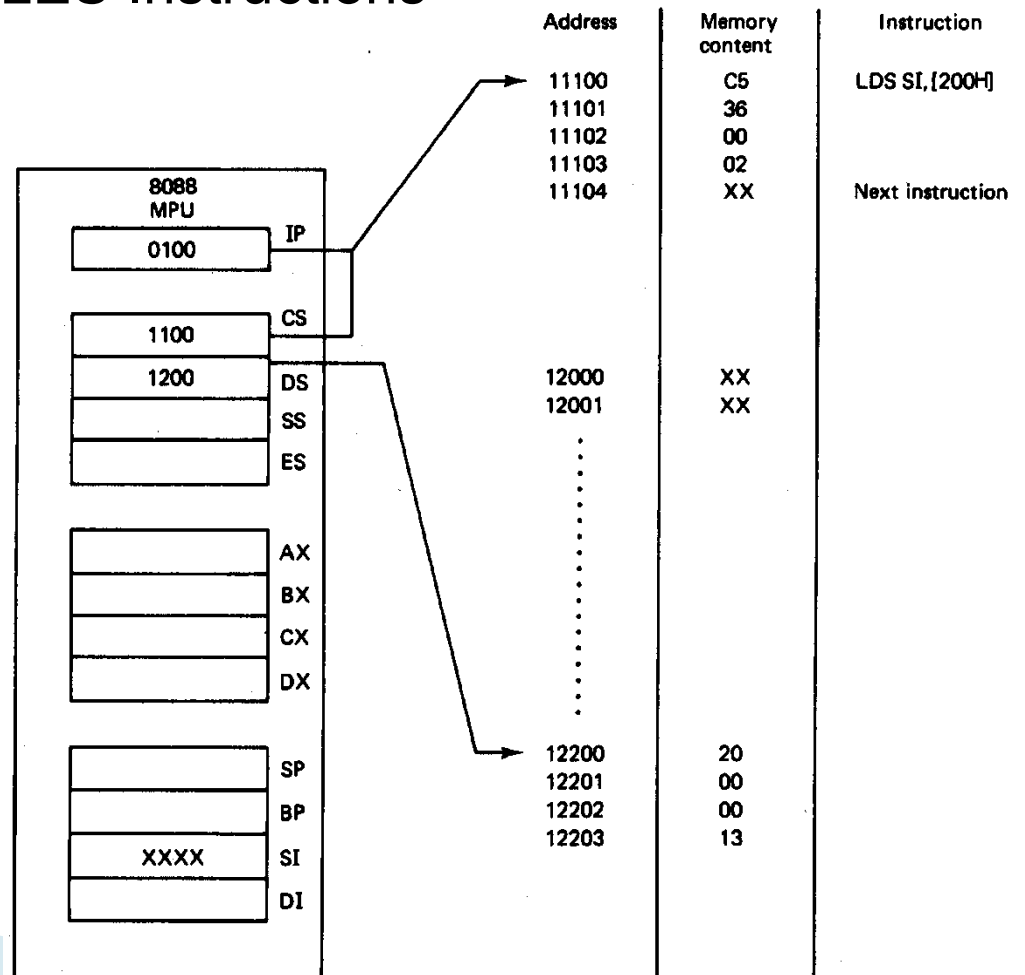
- ▶ The LEA, LDS, and LES Instructions
 - The LEA, LDS, LES instructions provide the ability to manipulate memory addresses by loading either a 16-bit offset address into a general-purpose register or a register together with a segment address into either DS or ES

Mnemonic	Meaning	Format	Operation	Flags affected
LEA	Load effective address	LEA Reg16,EA	EA → (Reg16)	None
LDS	Load register and DS	LDS Reg16,EA	EA → (Reg16) EA+2 → (DS)	None
LES	Load register and ES	LES Reg16,EA	EA → (Reg16) EA+2 → (ES)	None

▶ e.g. LEA SI, [DI+BX+5H]

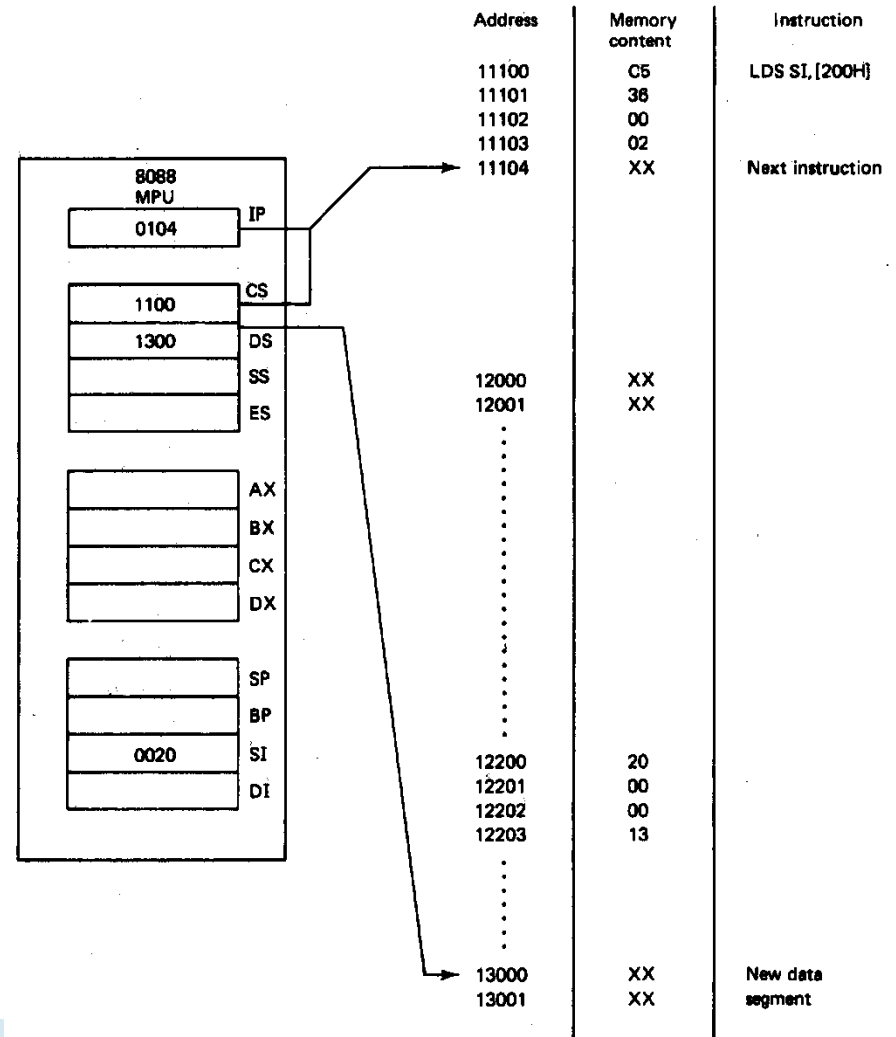
Data-Transfer Instructions (cont.)

- ▶ The LEA, LDS, and LES Instructions
 - LDS SI, [200H]



Data-Transfer Instructions (cont.)

- ▶ The LEA, LDS, and LES Instructions
 - LDS SI, [200H]

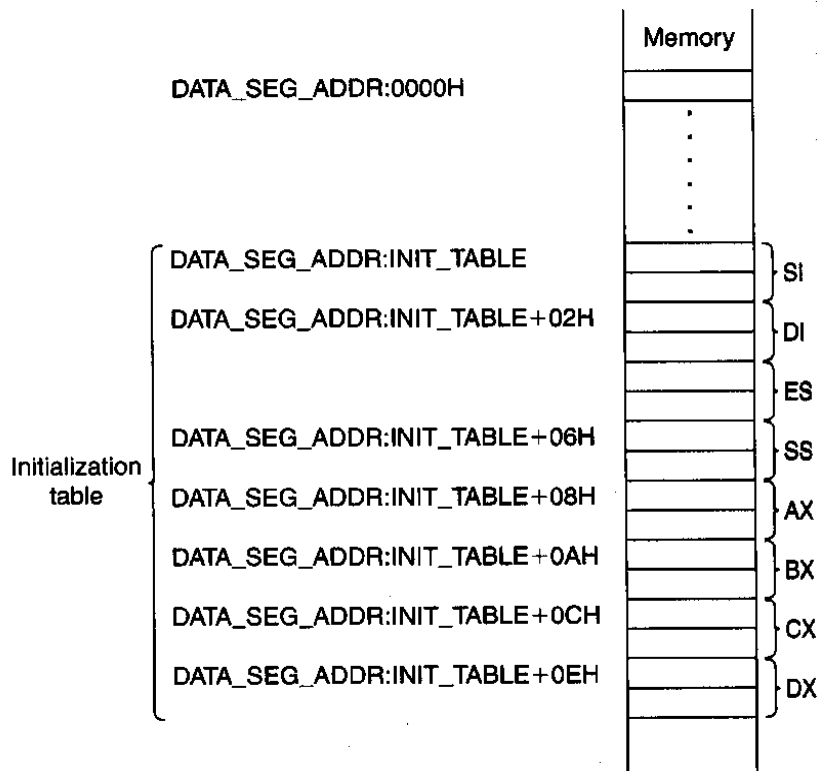


Data-Transfer Instructions (cont.)

▶ *EXAMPLE:*

- Initializing the internal registers of the 8088 from a table in memory

▶ *Solution:*



```
MOV AX, DATA_SEG_ADDR
MOV DS,AX
MOV SI,[INIT_TABLE]
LES DI,[INIT_TABLE+02H]
MOV AX,[INIT_TABLE+06H]
MOV SS,AX
MOV AX,[INIT_TABLE+08H]
MOV BX,[INIT_TABLE+0AH]
MOV CX,[INIT_TABLE+0CH]
MOV DX,[INIT_TABLE+0EH]
```

Scope of research

- ▶ we can developed new data transfer instruction which will be more suitable then current instruction. or instruction which is more easily understand by user and the microprocessor. and which has capability of fast execution speed with large data and less time.
- 