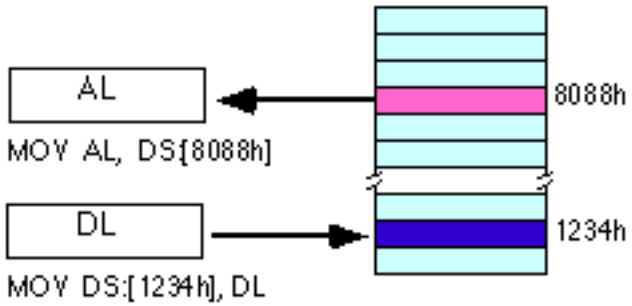
TUT

Addressing Modes, PUSH & POP instructions

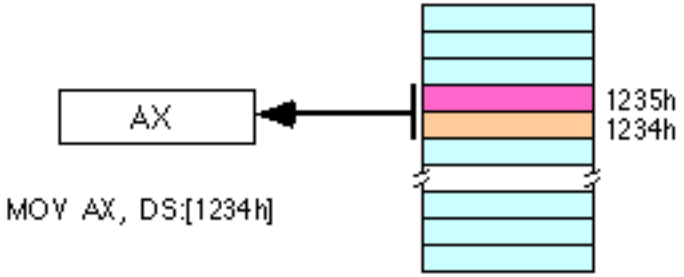
Which Memory location will be addressed with following instructions?

- MOV AL,DS:[8088h]
- loads the al register with a copy of the byte at memory location 8088h.
- MOV DS:[1234H],DL
- stores the value in the dl register to memory location 1234h:

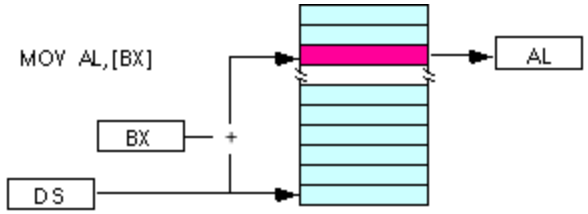


Which Memory location will be addressed with following instructions?

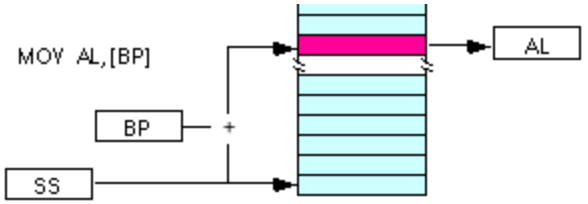
- MOV AX,DS:[1234h]



- MOV AL,[BX]

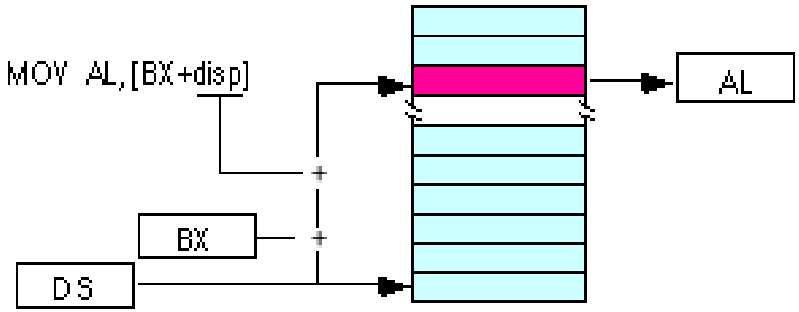


- MOV AL,[BP]

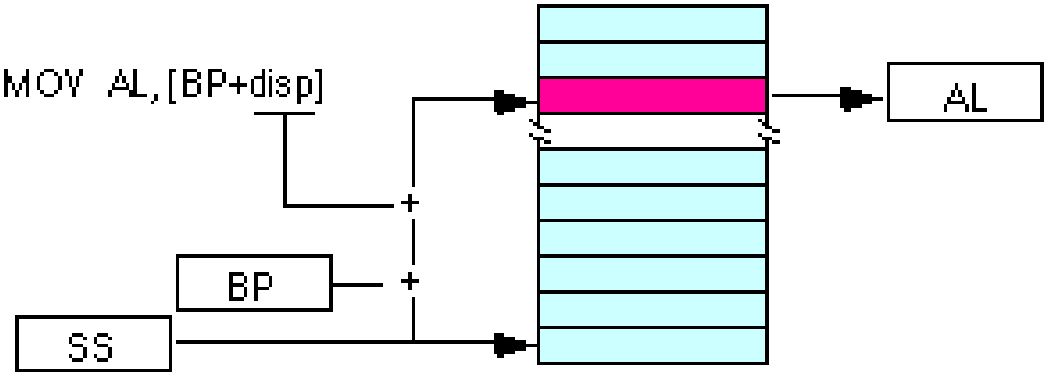


Which Memory location will be addressed with following instructions?

- MOV AL,[BX+DISP]



- MOV AL,[BP+DISP]



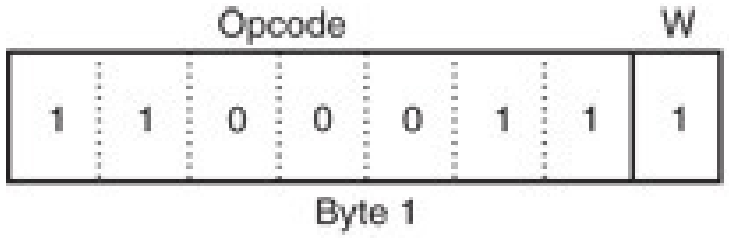
Suppose BP contains 1000h, BX contains 2000h, SI contains 120h, and DI contains 5. What does following instructions loads

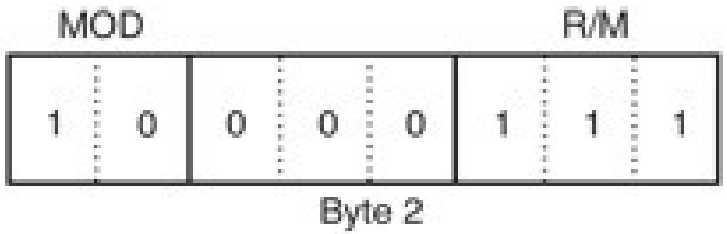
- MOV AL,10H[BX+SI]
- loads al from address DS:2130
- MOV CH,125H[BP+DI]
- loads ch from location SS:112A;
- MOV BX,CS:2[BX][DI]
- loads bx from location CS:2007.

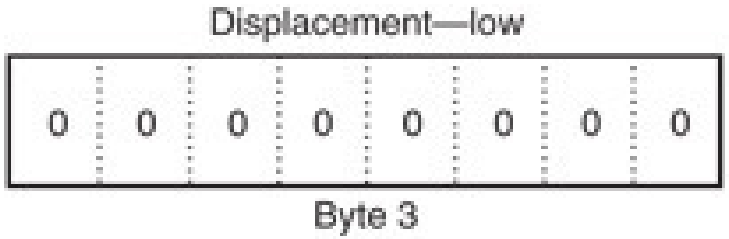
Which Memory location will be addressed with following instructions?

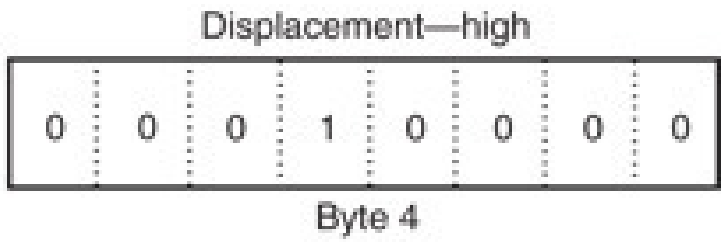
- MOV WORD PTR [BX+1000H] ,1234H
- moves a 1234H into a word-sized memory location addressed by sum of 1000H, BX, and DS x 10H
- What is length of this instruction ?
- 6-byte instruction
  - 2 bytes for the opcode; 2 bytes are the data of 1234H; 2 bytes are the displacement of 1000H
- Find instruction code (HA)

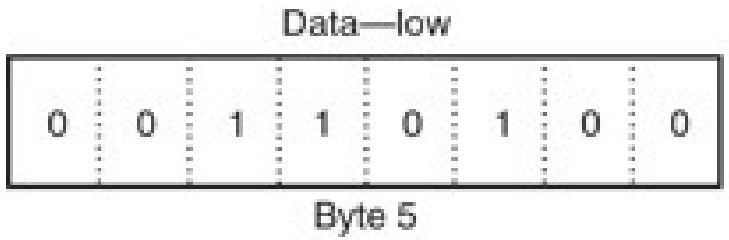
Instruction code for : MOV WORD PTR [BX+1000H] ,1234H

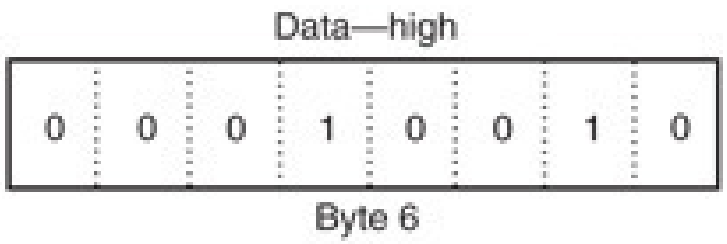




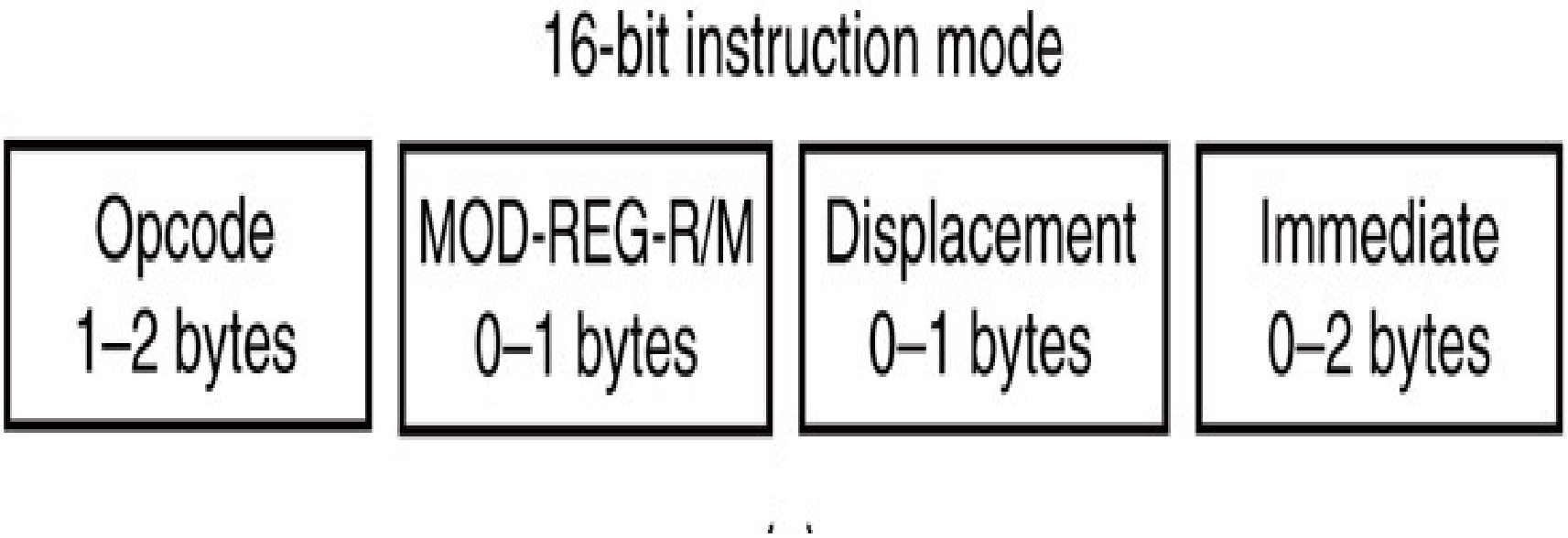








16-bit INSTRUCTION FORMAT



Segment MOV Instructions

- If contents of a segment register are moved by MOV, PUSH, or POP instructions, a special bits (REG field) select the segment register.
- The opcode for this type of MOV instruction is different for the prior MOV instructions
- An immediate segment register MOV is not available in the instruction set
- To load a segment register with immediate data, first load another register with the data and move it to a segment register.

A MOV BX,CS instruction converted to binary machine language.

Opcode = MOV
MOD = R/M is a register
REG = CS
R/M = BX

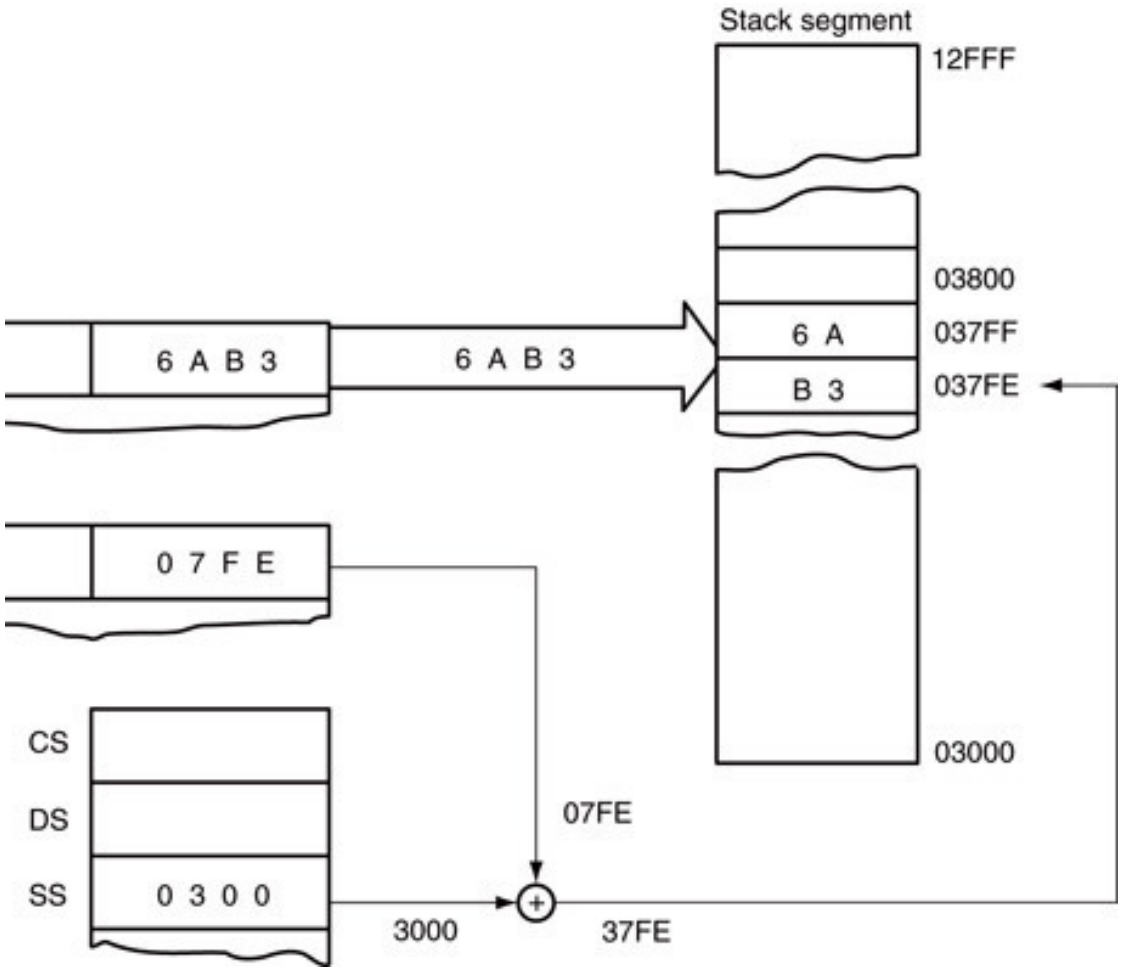
- Segment registers can be moved between any 16-bit register or 16-bit memory location.

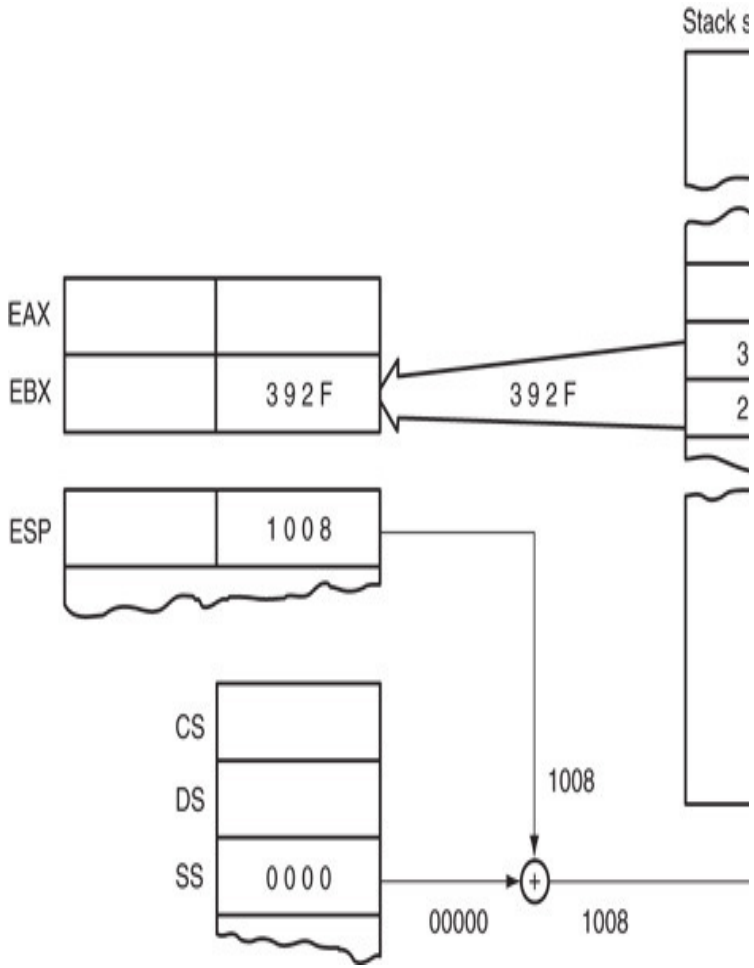
PUSH/POP

- Important instructions that *store* and *retrieve* data from the LIFO (last-in, first-out) stack memory.
- Six forms of the PUSH and POP instructions:
  - register, memory, immediate
  - segment register, flags, all registers

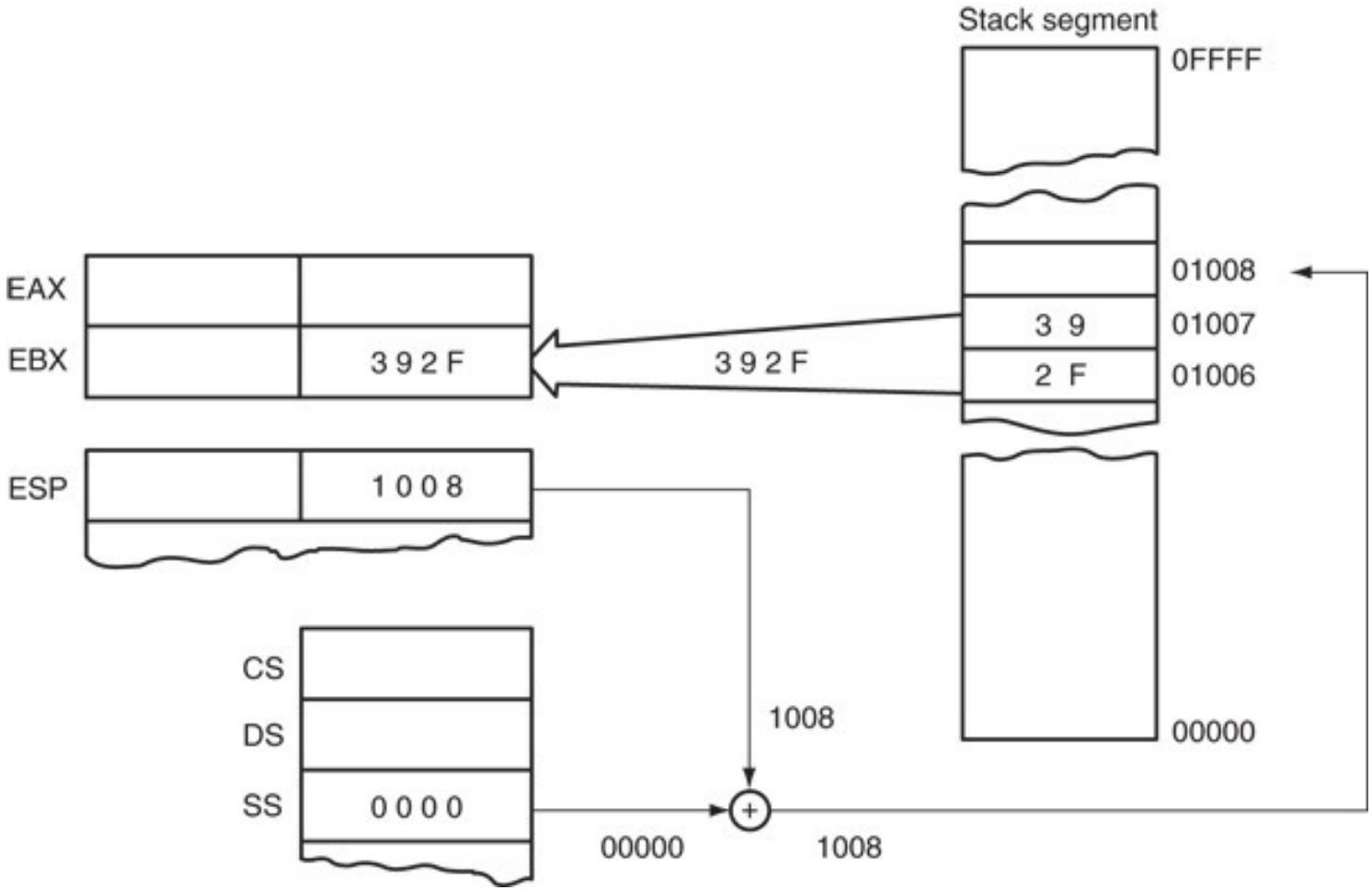
- Register addressing allows contents of any 16-bit register to transfer to & from the stack.
- Memory-addressing PUSH and POP instructions store contents of a 16- bit memory location on the stack or stack data into a memory location.
- Immediate addressing allows immediate data to be pushed onto the stack, ***but not popped off the stack.***
- Segment register addressing allows contents of any segment register to be pushed onto the stack or removed from the stack.
- The flags may be pushed or popped from that stack.
  - contents of all registers may be pushed or popped

The effect of the PUSH AX instruction : Which stack memory locations will be accessed?
Assume : SP= 07FE, SS=0300





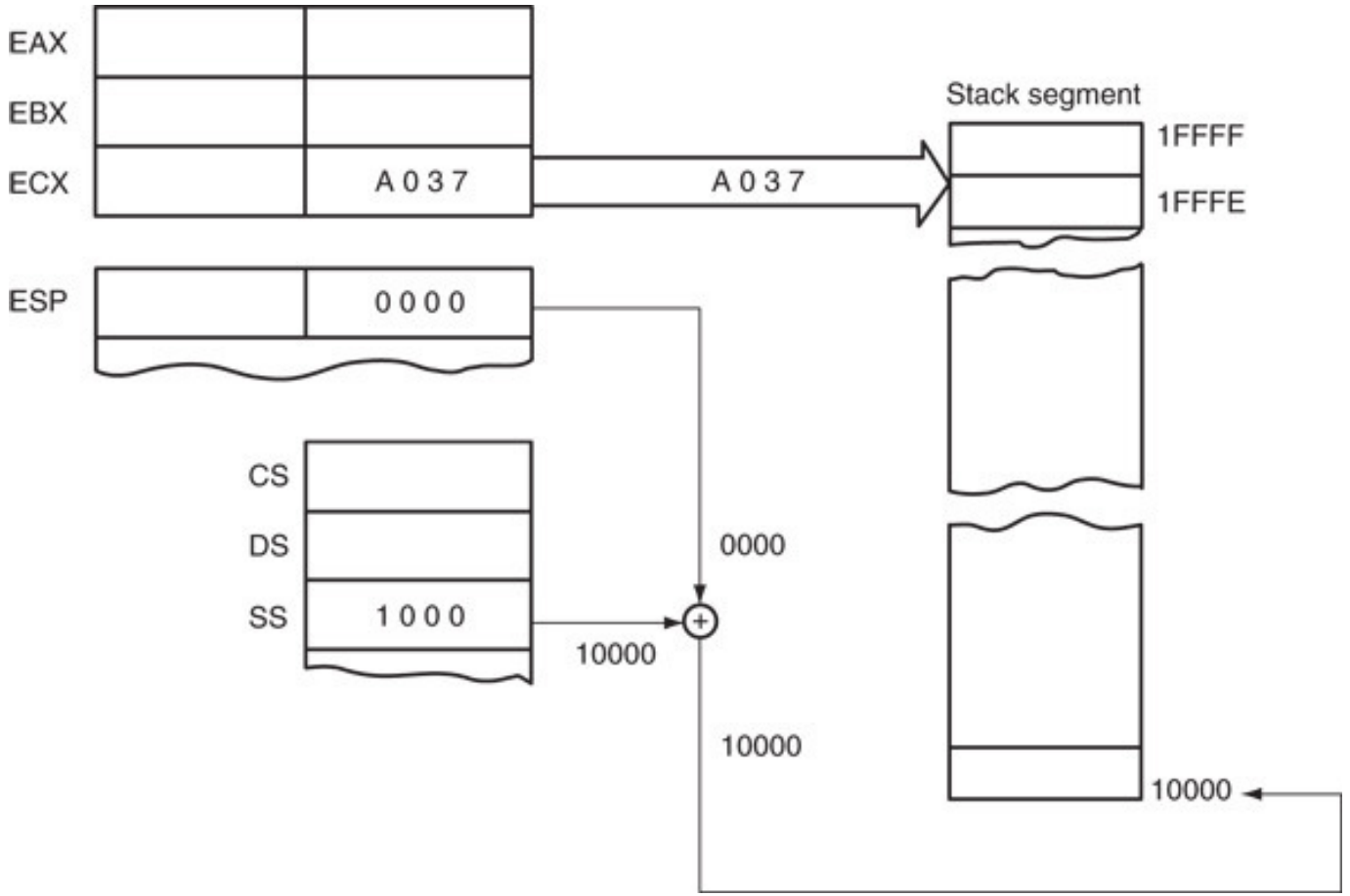
The POP BX instruction, showing how data are removed from the stack.
What is the effect of the execution of this instruction if SS=0000H and SP=1008 assumed.



Initializing the Stack

- When the stack area is initialized, load both the stack segment (SS) register and the stack pointer (SP) register.
- All segments are cyclic in nature

The PUSH CX instruction, showing the cyclical nature of the stack segment. This instruction is shown just before execution, to illustrate that the stack bottom is contiguous to the top.



All segments are cyclic in nature the top location of a segment is contiguous with the bottom location of the segment