39 c 39 f 19 s
5 13 19 33
ω C

51

12 0 20 11 9
C H A P 4

Homework 3 Solution Key
Note: It could not be a 32-bit more instruction.

Because the problem occurred only a 16-bit.

Displacement (4,004) — and not a 32-bit one.

CEX, SS: [BP + 4,004]

MOV EAX, word [BP + 16,004]

MOV EAX, word [BP + 16,004]

85 9E 00 4C
For ALL cases:

1. **Stack Segment**: 16-bit

2. **Assuming**:

   - The operand being a 32-bit register.
   - Subtracts 2 from SP and Moves The Counter.

   **ESI** and **Indexes SP by 4**

   **Move The Counter off the (Top of) Stack**

   **Move** AX on Top of the Stack

   **Subtracts 2 from SP and Moves The Counter**

   **Setup**:

   \[
   6766
   \]

   \[
   8B
   \]

   **MOV ESI [EAX]**

   **MOV ESI [EAX]**
Assuming an 16-bit stack segment.

By example:

After: $SS:SP$

SS = 0200
SP = 0100
X = 1400

Subtracts 4 from SP and moves the D-word

80000000 M onto the stack.

(4) Subtracts 4 from SP and moves the D-word.

(5) Move the counter of the (top of) stack to SP.

(6) DS and initializes SP by 2.

(7) DS and initializes SP by 2.

(8) Subtracts 4 from SP and moves the counter.

That is:

PPPPP
02100
00
00
19

Recall
DS would be used instead.

Register. If we use MOV AX, DS:[EBP],
would normally use SS as the segment.
Should be used. For example, MOV AL, [EBP]
to specify (overrice) the segment register that
any instruction that addresses memory in other
A byte that may be added in front of almost
the processor EAX registers.

ASI, preserving on the Pentium, uses in
Then, 8 is added/subtracted to/from
position given by RSI is copied to RAX
Thus is, the 64-bit data stored at memory
\[
\text{RSI} \rightarrow \text{RSI} + 8
\]
\[
\text{RSI} \rightarrow \text{RAX} \rightarrow \text{LDOSO}
\]
of memory.

The assembler to allocate one byte

a specific address: e.g. 08

Tells

assembly operation in respect to

The assembler should perform the

as to control the way in which

The assembler, which is used to change

A special instruction/keyword defined by

? (You got the idea)

Same for DX/ES

\[
\begin{align*}
\text{push} & \text{ edx} \\
\text{pop} & \text{ ecx} \\
\text{push} & \text{ edx} \\
\text{push} & \text{ ecx}
\end{align*}
\]
\[ A = 0 \quad 0 = 0 \]
\[ C = 0 \quad 1 = 1 \]
\[ Z = 0 \quad S = 1 \]

\[ \text{DAA, JAS} \]

\[ \text{S.33} \]

\[ \text{S.19} \]

\[ \text{S.38} \]
C11
MVI DI, 0F87H
MVI AL, 66H
MVI CX, 300H

SECTION S-6

S.49
S.55