

# Lab 4

**Sara Frazer**

11/15/23

CDA 3203 Computer Logic Design

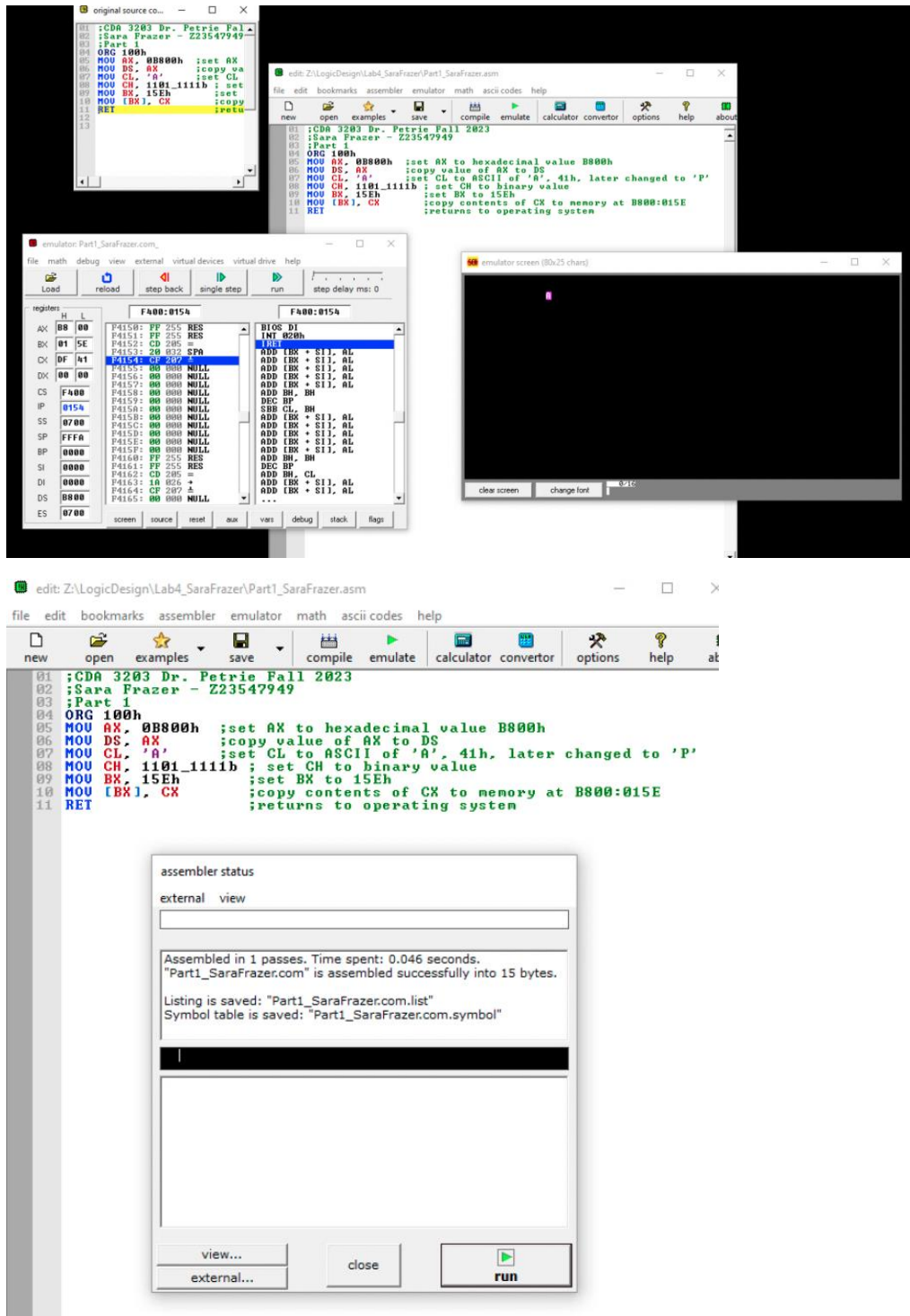
Fall 2023

Dr. Maria Petrie

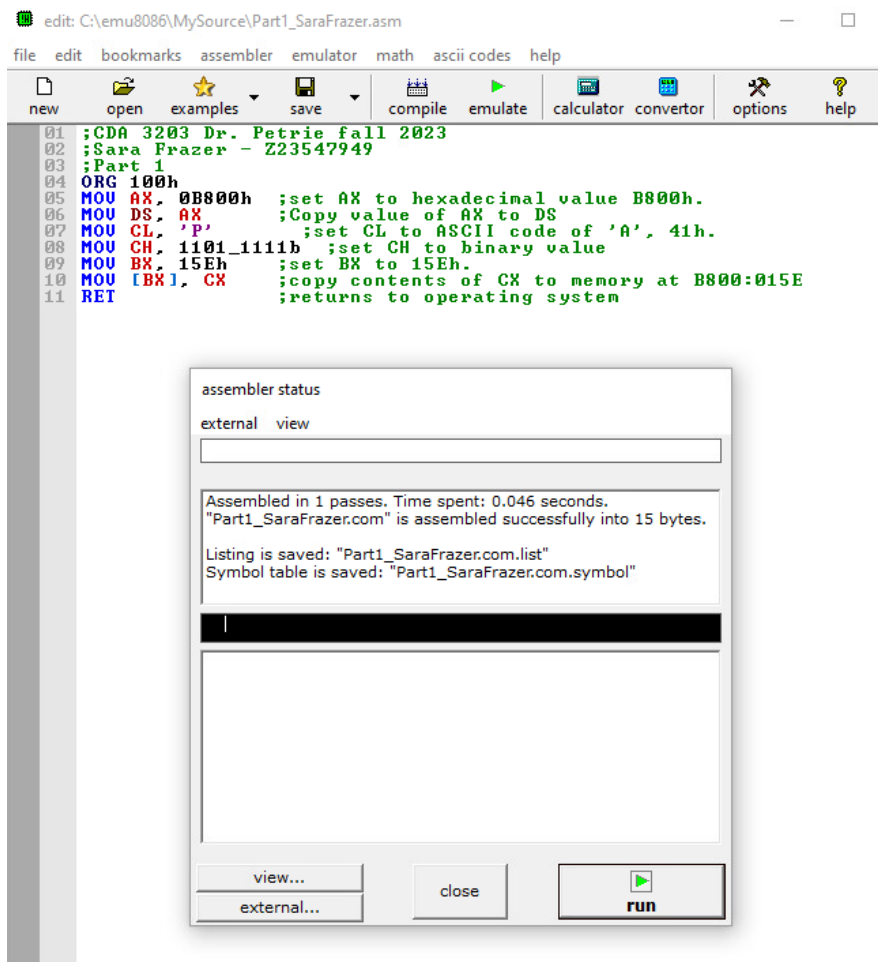
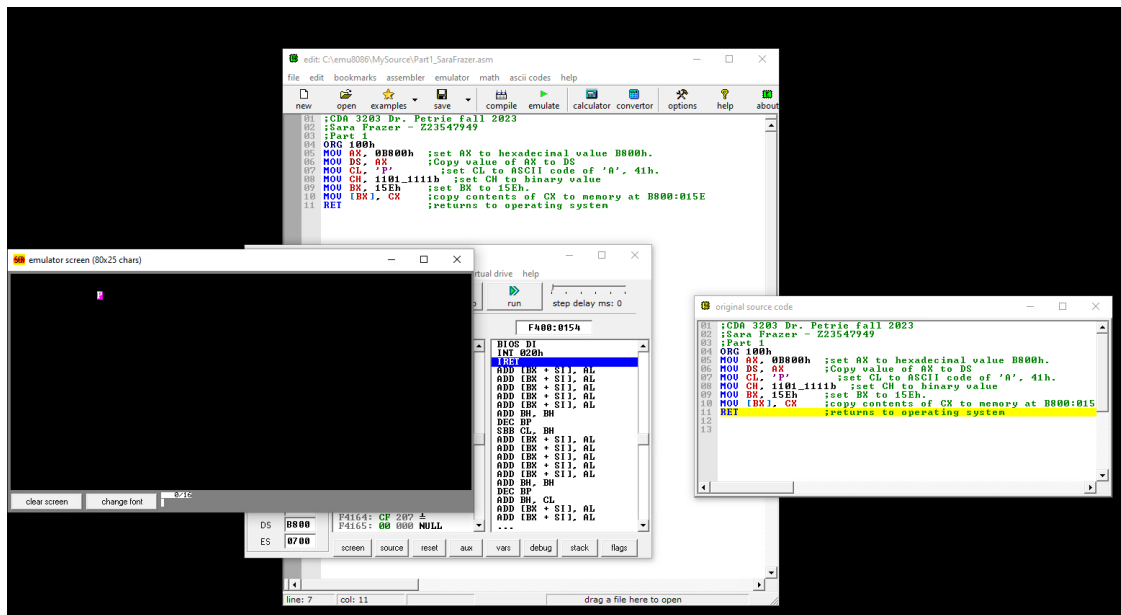
Florida Atlantic University

## Part 1

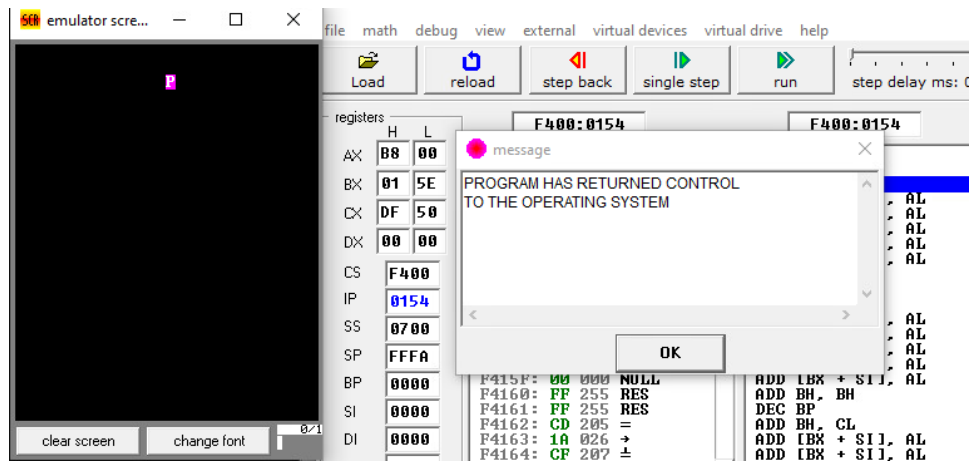
### Snip of editor window that displays 'A' on emulator screen with successful compilation window



## Snip of emulator window containing machine code to display 'P'



## Snip window displaying message



## Flags Window



## Hand work

Write the HEX Machine Code for each Assembly Instruction

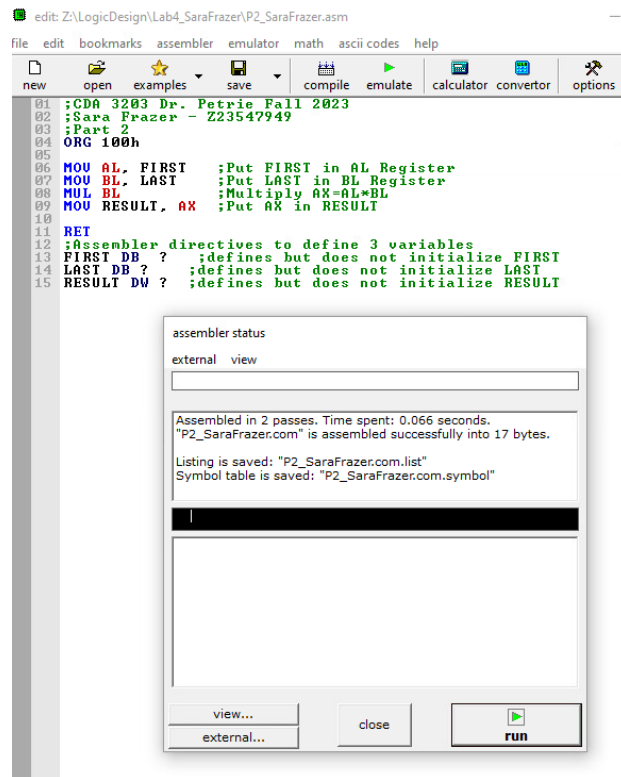
- |                       |        |
|-----------------------|--------|
| 1. MOV AX, 0B800h     | B800B8 |
| 2. MOV DS, AX         | 8ED8   |
| 3. MOV CL, 'A'        | B141   |
| 4. MOV CH, 1101_1111b | B5DF   |
| 5. MOV BX, 15Eh       | BB5E01 |
| 6. MOV [BX], CX       | 890F   |
| 7. RET                | C3     |

8. The CS:IP address of the first instruction is 0700 0100
9. The CS:IP address of the last instruction is 0700 010E
10. What is the **Effective Address** of the first instruction?  
Show work of how you calculated the physical address.

$$\begin{array}{r} + 0700 \\ 0000 \\ \hline 0700 \end{array}$$

## Part 2

### Successful Compilation



### Hand Work

- Compare the instructions in the Editor window with the assembled code in the Emulator window.  
What did the Emulator substitute for the following in the assembled instructions:  
FIRST was changed to 0010Dh  
LAST was changed to 0010Eh  
RESULT was changed to 0010Fh
- In Symbol Table snippet find the Offset of  
RESULT = 0010F R3
- DS:Offset is the address of RESULT. Calculate the 20-bit  
**Effective Address** of RESULT in Hex? 07109  
Show work of how you calculated the physical address:  
$$\begin{array}{r} 0700 \\ + 0109 \\ \hline 07109 \end{array}$$
- Look at RESULT and AX in the different windows snipped show value AFTER you Single Stepped :  
In Emulator window, AX = R30F01  
In Variables window, RESULT = 1127  
In RAM window, 2 bytes of RESULT = 67 04
- SPOT Course Evaluation – Complete Survey before SPOT closes (see Canvas Dashboard) and submit proof you completed the evaluation. Do not submit your answers, just proof it was completed. Submit snip.

**-10** If used EE-203 computers and didn't erase any saved .asm and .com files for both Part 1 and 2!

## Before Running Code

emulator: P2\_SaraFrazer.com

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	11
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0100

07100:	A2	162	6
07101:	0D	013	CRET
07102:	01	001	0
07103:	88	136	e
07104:	1E	030	▲
07105:	0E	014	▯
07106:	01	001	0
07107:	F6	246	÷
07108:	E3	227	▯
07109:	A3	163	ú
0710A:	0F	015	*
0710B:	01	001	0
0710C:	C3	195	†
0710D:	17	023	‡
0710E:	31	049	1
0710F:	00	000	NULL
07110:	00	000	NULL
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

0700:0100

```
MOV [0010Dh], AL
MOV [0010Eh], BL
MUL BL
MOV [0010Fh], AX
RET
POP SS
XOR [BX + SI], AX
ADD [BX + SI + 09090h], 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags

## After Running Code

emulator: P2\_SaraFrazer.com

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	04	67
BX	00	31
CX	00	11
DX	00	00
CS	0700	
IP	010C	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:010C

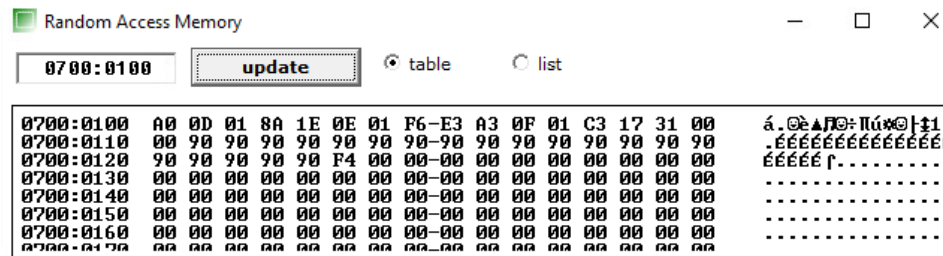
07100:	A0	160	á
07101:	0D	013	CRET
07102:	01	001	0
07103:	8A	138	è
07104:	1E	030	▲
07105:	0E	014	▯
07106:	01	001	0
07107:	F6	246	÷
07108:	E3	227	▯
07109:	A3	163	ú
0710A:	0F	015	*
0710B:	01	001	0
0710C:	C3	195	†
0710D:	17	023	‡
0710E:	31	049	1
0710F:	67	103	g
07110:	04	004	♦
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

0700:010C

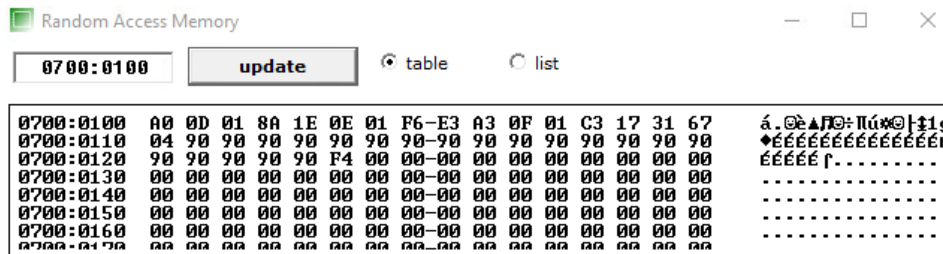
```
MOV AL, [0010Dh]
MOV BL, [0010Eh]
MUL BL
MOV [0010Fh], AX
RET
POP SS
XOR [BX] + 04h, SP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags

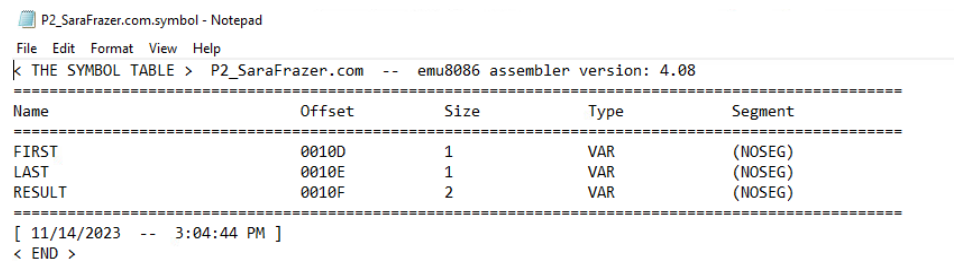
### RAM Window Before



### RAM Window After



## Symbol Table



### Variables After Run

