Laboratory Manual

for

Computer Organization and Assembly Language

Course Instructors

Lab Instructor(s)

Section

Semester

Department of Computer Science



COAL Lab 5Manual

Objectives:

- Zero/Sign Extension of Integers, MOVZX, MOVSX
- Status Flags
- Data related Operators & Directives
- Problems & Assignments

5.1 Zero Extension of Integers

Pseudo-op	Explanation	Syntax	Example	
MOVZX	Move with zero-extend	MOVZXreg32, reg16/mem16 MOVZXreg32, reg8/mem8 MOVZXreg16, reg8/mem8	MOVZXeax,bx MOVZXeax,bl MOVZXax,bl	

5.2 Sign Extension of Integers

Pseudo-op	Explanation	Syntax	Example	
MOVSX	Move with sign-extend	Same as above	Same as above	

5.3 Flags Register:

Flags Registerdeterminesthecurrent stateoftheprocessor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program. Generally you cannot access these registers directly. Two types of flags register are available.

- 1. Status Flags: CF, PF, AF, ZF, SF, OF
- 2. Control Flags: TF, IF, DF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				О	D	Ι	T	S	Z		A		P		C

The individual flags are explained in the following table.

CF	Carry Flag	Thisflagis set to 1 when there is an <i>unsignedoverflow</i> .
PF	Parity Flag	Thisflagis set to 1 when there is even number of one bits in result.
AF	Auxiliary	During addition or subtraction if a carry goes from one nibble to the next this
	Carry Flag	flag is set.
ZF	Zero Flag	It is set to 1 when last mathematical or logical instruction has <i>produced a zero in</i>
		its destination.
SF	Sign Flag	It is set to 1 when <i>result isnegative</i> .
TF	Trap Flag	Usedfor on-chip debugging.
IF	Interrupt	When this flag is set to 1 CPU reacts to interrupts from external devices.
	Flag	
DF	Direction	Thisflagisusedbysomeinstructionstoprocessdata chains, when thisflagis set to0-
	Flag	theprocessingisdoneforward, when this flagis set to1theprocessing isdone
		backward.



OF	Overflow	It isset to 1 when there is a signed overflow.
	Flag	

5.4 Data Related Operators & Directives

- OFFSET Operator, discussed in previous lab.
- PTR Operator, discussed in previous lab.
- TYPE Operator

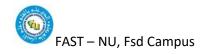
Pseudo-op	Explanation	Syntax	Example
TYPE	Returns size in bytes of a single elements of a variable	MOVreg, TYPE mem	.data Var1 BYTE ? Var2 DWORD ? .code MOV AX, TYPE VAR2 ; AX = 0004

• **LENGTHOF** Operator

Pseudo-op	Explanation	Syntax	Example
LENGTHOF	Counts the number of elements in an array	MOVreg, LENGTHOF mem	.data Var1 BYTE 10, 20, 30 Var2 DWORD 5 DUP (3 DUP (?)) .code MOV AX, LENGTHOF VAR2 ; AX = 000F

• SIZEOF Operator

Pseudo-op	Explanation	Syntax	Example
SIZEOF	Returns LEGNTHOF*TYPE	MOVreg, SIZEOF mem	.data Var1 DWORD 10H, 20H, 30H Var2 DWORD 5 DUP (3 DUP (?)) .code MOV AX, SIZEOF VAR1 ; AX = 000C



Problem(s) / Assignment(s)

Discussion & Practice

Estimated completion time: 1 hr, 30 mins

Problem 5.1: *MonitoringStatus Flag*

Estimated completion time:15 mins

Sr.	Instructions	Flags						
No.		CF	OF	SF	ZF			
1.	MOVAX,620H							
2.	SUBAH, OF6H							
3.	MOVAX,720H							
4.	SUBAX, OE 6H							
5.	MOVZXEBX,AX							
6.	MOVBL,-127							
7.	MOVSXEAX,BL							
8.	DEC BL							
9.	MOVAH, 620H							
10.	SUBAH, OF6H							

Problem 5.2: Array Manipulation

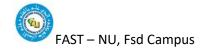
Estimated completion time:20 mins

Let us have an array,

A_array WORD10FFH, 6323H, 0ABCDH, 828H

Compute the sum for higher and lower bytes of each value, Store result of each addition in AX, BX, CX and DX registers and display them on console. Observe CF, OF, ZF and SF during each addition. You may Ignore the carry in each addition.

Hint: AX = 10 + FF



Problem 5.3: Array Manipulation

Estimated completion time:20 mins

Repeat Problem 5.2 with alteration of

- 1. Putting the result in a 5 elements DWORD size array namely B_array.
- 2. Keeping Status Flag intact at the end of the program and store it as 5th element of B_array.

Also display

- 1. Type of B_array in AH.
- 2. No. of characters of B_array in BH.
- 3. Size of B_array in BL.

You are done with your exercise(s), make your submission ©

Submission Guidelines:

- Attach your .asm files of all programs and examples by your slate account
- Once you have attached all the files, click on the 'Submit' button.