Microprocessor Based System Design Laboratory (Gy)

Department of Computer Science and Technology, IIEST Shibpur

Date: August 16, 2024

Experiment No. 4: 8085 Assembly Language Programming – Writing Subroutines

Objective: To write 8085 Assembly Language Programs involving Subroutines and test them at the 8085 SDK (System Development Kit).

Preamble: By this time you are familiar with writing, executing, and testing 8085 assembly language programs involving **conditions and iterations**

Following are a few examples where simple **C functions** are "*implemented*" by 8085 programs. Please consult the **2-page Instruction Set** and **16-page Instruction Set Reference Encyclopedia** for validating the program and comment(s) against each instruction of the program.

Example 1. **C function** to compute ***f(x,y) = 2(x+y)***

```
unsigned int f(unsigned int x, unsigned int y) {
        return (2*(x+y));
}

int main() {
        unsigned int m, n;
        m = f(2,3);
        n = f(3, 7);
}
```

Label	**Assembly Instruction**	**Memory Address (Hex)**	**Machine Language (Hex)**	**Comment**
Alternative 1: Let parameters x and y be passed through registers **B** and **C**, respectively. Return Value is received through register **D**				
Data Part {		2000		Location for *m*
		2001		Location for *n*
⋮				
main:	MVI B,02H	2100	06	For calling f(2,3), 2 should be passed through register B
		2101	02	
	MVI C, 03H	2102	0E	For calling f(2,3), 3 should be passed through register C
		2103	03	
	CALL f	**2104**	**CD**	**The function f is loaded at 2117H location**
		2105	17	
		2106	21	
	MOV A,D	2107	7A	Function f is expected to return its value through D, copy it to A
	STA 2000H	2108	32	Store the return value in variable m (i.e., at 2000H)
		2109	00	
		210A	20	
	MVI B,03H	210B	06	For calling f(3,7), 3 should be passed through register B
		210C	03	
	MVI C, 07H	210D	0E	For calling f(3,7), 7 should be passed through register C
		210E	07	

	CALL f	**210F**	**CD**	**The function f is loaded at 2117H location.**
		2110	17	
		2111	21	
	MOV A,D	2112	7A	Function f is expected to return its value through D, copy it to A
	STA 2001H	2113	32	Store the return value in variable n (i.e., at 2001H)
		2114	01	
		2115	20	
	HLT **or** RST 5	2116	76 **or** EF	**HLT** halts 8085. You have to "**RESET**" the **kit** to make the kit working again. Check the contents of the memory location 2000H (storing variable ***m***) and location 2001H (storing variable ***n***) (by "**EXMEM**" key press). If you use RST 5 (in place of HLT) the Interrupt Service Routine (**ISR**) in your kit for the software interrupt instruction RST 5, saves the contents of all the registers of the 8085 microprocessor at fixed memory locations **(see chapter 6, page 43, SDK User Manual)**. Subsequently, you can check the contents of the register **A, B, C, D** (by "**EXREG**" key press) in addition to memory locations 2000H and 2001H (by "**EXMEM**" key press). Check contents of **PC** too.
f:	**MOV A, B**	**2117**	**78**	**Function f starts here. The 1st parameter in B is copied to A**
	ADD C	2118	81	The 2nd parameter in **C** is added to the 1st parameter (now in **A)**
	ORA A	2119	B7	Used to clear the CY flag since the next instruction uses it. Note that A remains unchanged.
	RAL	211A	17	The sum (in **A**) is left shifted by 1 bit to achieve multiplication by 2. This instruction shifts CY flag in Lsb of A. Please note that CY flag has been made 0 by the previous instruction (ORA A)
	MOV D, A	211B	57	Result is kept in D before returning
	RET	211C	C9	Return to the caller

- **Please follow the tabular format of the above examples while writing your programs for the problems given below.**
- **You may, however, defer writing comments until you have a correct running program.**

In today's laboratory class **each of you** have to

1. **try the example program given above**
2. write 8085 programs (including ***main()*** function) as specified below **in the format shown in the examples**. For each program, you **(i)** write the assembly code first, **(ii)** and then translate it to machine language using the Instruction Set Table provided to you, **(iii)** and finally you load, execute, and test the program.
3. **Get the program, that you have written in your notebook, signed by your teacher. Take a photo of your program and submit it at your Google Classroom.** The problems statements are available as classwork in the Classroom.

Programs

[**Write main() function too for each of the following programs.** Consult the **2-page Instruction Set** and **16-page Instruction Set Reference Encyclopedia** for choosing appropriate Instructions while writing programs]

1. **Try the example programs given above**
2. Repeat the program shown in Example 1 (above) by changing parameter and return value passing through memory locations. Let memory locations 2000H, 2001H be used to pass x, y, respectively (in place of registers B, C) and the return value be passed through memory location 2002H (in place of register D in the example).
3. Implement the C function ***int isupper(int c)*** that return 1 if ***c*** is an uppercase letter (that is, ’A’ <= ***c*** <= ’Z’), returns 0 otherwise.
   Let the memory location 2000H be used to pass the parameter ***c*** and 2001H be used for the return value.
4. Implement the C function ***int islower(int c)*** that return 1 if ***c*** is a lowercase letter (that is, ’a’ <= ***c*** <= ’z’), returns 0 otherwise.
   Let the memory location 2002H be used to pass the parameter ***c*** and 2003H be used for the return value.
5. Implement the C function ***int isalpha(int c)*** that return 1 if ***c*** is an English letter (lowercase or uppercase), returns 0 otherwise. Use ***isupper()*** and ***islower()*** functions to implement this function.
   Let the memory location 2004H be used to pass the parameter ***c*** and 2005H be used for the return value.
6. Implement the C function ***unsigned int largest (unsigned int data[], unsigned int size)*** that returns the largest of the ***size*** number of elements of the array ***data[]***.
   Let the memory locations 2006H-2007H (2 Bytes) be used for the parameter ***data[]***, 2008H for the parameter ***size,*** and 2009H be used for the return value.
7. Implement the C function ***void swap (unsigned int *n1, unsigned int *n2)*** that swaps the integer data pointed to by ***n1*** and ***n2***.
   Let the memory locations 200AH-200BH (2 Bytes) be used for the parameter ***n1***, and 200CH-200DH (2 Bytes) for the parameter ***n2.***