Experiment 3 – Procedure usage and stack operations

2020-2021 Spring

Preliminaries:

1. Students who will attend to this experiment are assumed to know:
   - Usage of registers on 8086
   - Usage of memory operations on 8086
   - Usage of variables on 8086
   - Usage of addressing modes on 8086
   - Usage of procedures within assembly on 8086
   - Usage of stack operations on 8086
   - Usage of emu8086
2. Study related topics from course slides and the textbook
3. Run example codes from slides and textbook
4. Study instruction set for 8086
5. Always comment your code!!

Work:

1. Write an assembly code that uses a procedure/subroutine to find all-zero nibble's offset in AX register. Print offset of that nibble to the screen. Call the procedure for at least 5 different examples in your main code.

   Example: AX = 10F7h, second nibble is all-zero, offset is 2(counting from highest nibble).

   Hint: You can use "int 21h" to print characters to screen.

2. Write an assembly code that uses a procedure/subroutine to exchange the high nibble of AH with the low nibble of AL when the middle nibbles of AX are zero, using only shift, rotate and exchange(XCHG) operations, no MOV except setting AX for the first time, no AND-OR-XOR operations either.
   Call the procedure for at least 5 different examples in your main code.

   Example: AX = B007h, after calling the procedure, AX = 700Bh.

   Bonus: Write a procedure when the middle parts are nonzero with and-or operations permitted. Example: AX = B217h, after calling the procedure, AX = 721Bh.

3. (Optional) Write an 8086 program that searches the BIOS ROM for its creation date and displays that date on the monitor. If a date cannot be found display the message "date not found".

   Hint: Typically the BIOS ROM date is stored in the form xx/xx/xx beginning at system address F000:FFF5. Each character is in ASCII form and the entire string is terminated with the null character (00). Add a '$' character to the end of the string and make it ready for DOS function 09, INT 21h.

   Hint: Running your program under emu8086 may result some unexpected output, for testing on a real or real-enough environment, either run your program on a 32-bit Windows(which has necessary compatibility to run DOS programs written in assembly) or DOSBox which emulates a real 8086 compatible computer system accurately.

   Don't report anything in preliminary work for optional questions. You will be responsible for all questions during experiment.

Notes :

- **You should prepare a preliminary work report with the answers of the questions on the "Work" part.**
- **All answers should be in English, it may be better to put your assembly codes in a Text box for better readability, code parts has to use a Type Writer font like `Courier New`.**
- **Each answer code file should be uploaded to the system seperately. You should also upload a proper report containing all answers and results together with your comments.**