1. Explain two ways of generating a clock for a CPU, as discussed in the class and explain which is preferable and state reasons. A clock signal for a CPU can be generated by a quartz crystal or an RLC circuit. A Quartz Crystal is precise and can let the CPU reach speeds above 10 MHz. Quartz crystals are smaller, have lower temperatures and low cost, they are the preferred clock method for a CPU. The RLC has a resistor, inductor, and a capacitor. In an RLC, the resistance is low.
2. What are the key features of a synchronous memory write cycle.

CPU puts address and command on the synchronous bus and at the next clock state the Memory will look at the address and command, then in an operation with no wait states, will then immediately signal the data and ready signal. It will then see the ready signal and the CPU will remove the address, memory and write signals. In an operation with one wait state it wont immediately signal the ready and data signal, and will instead wait one clock cycle.

1. Declare the following:
   * 1. An uninitialized data declaration each for a 16-bit signed and unsigned integer

val1 SBYTE ?

val2 BYTE ?

* + 1. An uninitialized data declaration each for an 64-bit signed and unsigned integer

val3 QWORD ?

* + 1. An initialized data declaration for a 16-bit unsigned integer with the value B14Ch

val4 WORD B14Ch

* + 1. A null-terminated string variable with the value “Computer Architecture”.

str BYTE “Computer Architecture”, 0

* + 1. A symbolic constant named “*Area of a circle*” using the equal-sign directive and assign it an arithmetic expression that calculates the circumference in terms of Pi and radius, R of the circle.

AREAOFACIRCLE = Pi \* (R \* R)

1. Show the order of individual bytes in memory *(lowest to highest****)*** for the following variables *(use little endian order****):***

                     Fig WORD A1C1h  
                     Apple  DWORD A2B7E45Bh

|  |  |
| --- | --- |
| 0000 | C1h |
| 0001 | A1h |
| 0002 | 5Bh |
| 0003 | E4h |
| 0004 | B7h |
| 0005 | A2h |

1. Use assembler directives to declare an signed DWORD array of five elements and initialize it with the following values: 1, 22, -115, 124, -331. Show how to calculate the number of elements in this array and assign that value to a symbolic constant named “*NumberOfElements*”

list SDWORD 1, 22, -115, 124, -331

NUMBEROFELEMENTS = ($ - list) / 4

1. Using the AddTwo.asm program from the textbook as a reference, write a program *Add2Sub.asm* that adds two signed word sized integers, and then subtracts one word sized integer. Hand write the code. You do not need to assemble/execute.
2. ; Add2Sub.asm Samuel Adams
3. .386
4. .model flat,stdcall
5. .stack 4096
6. ExitProcess PROTO,dwExitCode:DWORD
7. .data
8. val1 SDWORD 4000h
9. val2 SDWORD 3000h
10. val3 SDWORD 1000h
11. finalValue SDWORD ?
12. .code
13. main PROC
14. mov eax,val1 ; start with 4000h
15. add eax,val2 ; add 3000h
16. sub eax,val3 ; subtract 1000h
17. mov finalValue,eax ; store the result (6000h)
19. INVOKE ExitProcess, 0 ; exits
20. main ENDP
21. END main