Fast**National University of Computer & Emerging Sciences, Karachi  
Fall2024 CSDepartment  
Assignment No. 01  
5th September 2024**

|  |  |
| --- | --- |
| **Course Code: EE2003** | **Course Name: Computer Organization & Assembly Language** |
| **Instructor Name / Names:** Shoaib Rauf, Kashan, Aashir Mahboob, Atiya, Muhammad Kariz,Muhammad Usman, Nauraiz Subhan | |

**Instructions:**

* Attempt all the questions.
* Don’t share your work, if your submission is matched to any member of your class, both will be marked 0 straight without asking who shared or who magically copied.
* You have to submit in Hard copy in the class timing on XXXXX i.e. 16th Sept,2024.
* No late submissions will be accepted.

**Max Points**: 50

**Question 1. [6] Points**

1. Compare High level languages with Assembly language, elaborating on there similarities and relationship.
2. Briefly explain the contrast between Assemblers and Compilers.
3. By referring to your explanation in part a and b, explain why High level languages are regarded as more portable as compared to Assembly languages. Justify your answer, with at least two examples.

**Questions 2 [4] Points**

Following is an Arduino compatible C program to blink an LED. Explain, by referring to the levels, the concept of virtual Machines. Also ***identify*** the VM(s) as VM(1) and VM(0), where compilation and translation occur respectively.

|  |  |
| --- | --- |
| **Level 3** | **Level 2** |
| define F\_CPU 16000000  define BLINK\_DELAY\_MS 1000  include <avr/io.h>  include <util/delay.h>  int main (void)  {  // Arduino digital pin 13 (pin 5 of PORTB) for output  DDRB |= 0B100000; // PORTB5    while(1) {  // turn LED on  PORTB |= 0B100000; // PORTB5  \_delay\_ms(BLINK\_DELAY\_MS);    // turn LED off  PORTB &= ~ 0B100000; // PORTB5  \_delay\_ms(BLINK\_DELAY\_MS);  }  } | main:  sbi 0x04, 5 ; PORTB5 output  loop: ; main loop begin  sbi 0x05, 5 ; PORTB5 high  call delay\_1000ms ; delay 1s  cbi 0x05, 5 ; 5 PORTB5 low  call delay\_1000ms ; delay 1s  rjmp loop ; main loop  delay\_1000ms: ; subroutine for 1s delay  ; initialize counters  ldi r18, 0xFF ; 255  ldi r24, 0xD3 ; 211  ldi r25, 0x30 ; 48  inner\_loop:  subi r18, 0x01 ; 1  sbci r24, 0x00 ; 0  sbci r25, 0x00 ; 0  brne inner\_loop  ret |

|  |  |
| --- | --- |
| **Level 1** | **Level 0** |
| **AVR enhanced Reduced Instruction Set Computer architecture** | IMG_256 |

**Question 3 [4+2+2+2] Points**

|  |  |
| --- | --- |
| **Mnemonic** | **Description** |
| SUB | Subtract one value from another |
| MOV | Move (assign) one value to another |

***MOV EAX, F\_F\_F\_F\_h*** *;Enter the last 4 digits of your roll number in the ;empty spaces e.g roll = 16K2404 > F2F4F0F4h*

***ADD EAX, 10000100h*** *;Add 10000100h to the contents of EAX*

1. Fill the table with the updated Flag values after the execution of the aforementioned Assembly language instructions.

|  |  |
| --- | --- |
| **FLAGS** | **VALUE** |
| Carry Flag (CF) |  |
| Overflow Flag (OF) |  |
| Zero Flag (ZF) |  |
| Sign Flag (SF) |  |
| Parity Flag (PF) |  |
| Auxiliary Carry Flag (AF) |  |

1. Explain why 8086 processors operating in Real Address Modes, can only access  **1MB** of RAM at a time.
2. Using the following Logical Address find the physical address of this memory location.

**12AB:025F**

1. Although 8086 processor has a 20 bit address bus, the segment address and offset are not in 20 bits representation. **Explain why?**

**Question 4 [5] Points**

Write a program that defines symbolic constants for all seven days of the week. Create an array variable that uses the symbols as initializers.

**Question 5 [5] Points**

Using the DUP directive, allocate space for 5 doublewords and 2 bytes in a data segment. Then fill the next 15 spaces with the character &, the 7 spaces that follow with the character % and the space after that with the character capital M.

**Question 6 [5+5] Points**

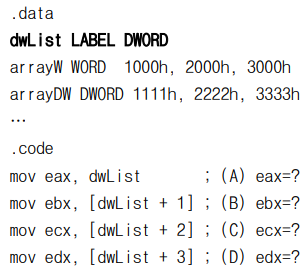
What are the values of AL register and overflow (OF), Sign Flag (SF)and carry flags (CF) after the execution of code below? Justify your answer. (Show all working of it)

i)

ii)

**Question 7 [5+5] Points**

Let’s suppose ‘dwList LABEL DWORD’ be added in the following code. What are the values of registers in (AD) in the code below? If there is an error, write ERROR and justify your answer. Please write in hexadecimal form



|  |  |  |
| --- | --- | --- |
| BOD | Benefit of Doubt | Max 1p/question |
| ECF | Error Carry Forward | Max 1p/question |

|  |  |  |  |
| --- | --- | --- | --- |
| Q1. a) | Similarities:   * Used to write programs that execute instructions on a computer * Converted to machine code. * Control of execution flow through loops, conditionals, and function calls.   Relationship:   * Translation process, one to many/one to one a brief explanation.   Also allow, anything similar to the following,   * High level languages can express complex operations with fewer lines of code using rich abstractions and libraries. * Assembly language requires more detailed instructions due to its low level nature. | Max 2P |  |
| b) | Any suitable comparison between the two in terms of the conversion process, e.g.   * Assembler performs translation from a mid/low level language with a one to one correspondence with machine instructions. * Compiler performs more complex tasks, involving multiple stages * lexical analysis, syntax analysis, semantic analysis, optimization, and code generation. | Max 2p |  |
| c) | Any suitable explanation with any two of the following keywords,  Abstraction from Hardware  Standardization Virtual Machine JVM/PVM  Assembly language is processor specific. | Max 2p |  |
|  |  |  |  |
| Q2. | VM0: The actual hardware and machine language of the Arduino micro-controller.  VM1: The assembly language or ISA level used for programming the micro-controller, which is generated by compiling the high-level C code.  Compilation occurs at VM1 (assembly language level).  Translation to machine code occurs at VM0 (hardware level).  Allow any related suitable explanation. | Max 4p |  |
|  |  |  |  |
| Q3  a) | Carry Flag (CF): Indicates if there was a carry out of the most significant bit during addition;  Overflow Flag (OF): Indicates if there was a signed overflow, meaning the result exceeded the represent able range;  Zero Flag (ZF): Indicates if the result of the operation is zero;  Sign Flag (SF): Indicates if the result of the operation is negative (i.e., the most significant bit is set);  Parity Flag (PF): Indicates if the number of set bits (1s) in the result is even;  Auxiliary Carry Flag (AF): Indicates if there was a carry from bit 3 to bit 4 in the lower nibble (used in BCD operations); | 4p Max | 2p Max as BOD/ECF |
| b) | Reference to size of Address bus i.e. 20-bit  2^26 = 1MB conversion | 2p Max |  |
| c) | = (Segment × 10h) + Offset  = 0x12AB0 + 0x025F = 0x12D0F | 2p Max | EOF |
| d) | A suitable reference to 16 bit Instruction Pointer register and segment registers. | 2p max | BOD |
|  |  |  |  |
| Q4 |  |  |  |
| Q5 |  |  |  |
|  |  |  |  |
|  |  |  |  |