**University of Central Punjab**

**Faculty of Information Technology**

**Spring 2022**



1. **Course Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Code** | CSCS3543 | | | |
| **Course Title** | Computer Organization and Assembly Language (Theory) | | | |
| **Credit Hours** | 4(3+1) | | | |
| **Assessment Instruments with Weights** (quizzes, midterms, final, assignments, presentation, lab work, etc.) | Assignments (20%)  Quizzes (15%)  Class Participation (5%)  Mid Term (20%)  Final Term (40%) | | | |
| **Course Instructors** |  | | | |
| **Course Coordinator** | Dr. Adnan Ghafoor | | | |
| **Office Hours** | TBA | | | |
| **Plagiarism Policy** | **All the parties involved in first cheating case in Midterm or Final term will be awarded ‘F’ for the course.** | | | |
| **Current Catalog Description** | The objective of this course is to understand the organization of microprocessor for instance Intel x86 in terms of assembly language instruction. This course primarily focuses on the detailed analysis and working of microprocessor. | | | |
| **Textbook (Notes)** | Assembly language, Design and Interfacing 4th edition, Mazidi & Mazidi | | | |
| **Reference Books** | * Assembly Language Intel based Computers, 4th Edition, by KIP, R, IRVINE * Computer Architecture and organization 8th Edition, by William Stallings. * Art of Assembly, | | | |
| **Teaching-Learning Methodology** | * More emphasis will be on coding assignments and practical lab sessions. Regular and online quizzes will be held to help students improving their understanding. Extra tutorial session will be conducted to increase problem solving kills. | | | |
| **Course Goals** | Upon successful completion of this course, students will be able:   * To develop the understanding, working and organization of microprocessor. * Develop a habit to write both memory and time optimized code using assembly language. | | | |
| **Topics Covered in the Course, with Number of Lectures on Each Topic** | Attached | | | |
| **Lab Projects** | Yes | | | |
| **Theory & Lab Exam** |  | | | |
| **Class Time Spent on** (in credit hours) | **Theory** | **Problem Analysis** | **Solution Design** | **Labs** |
| 1 | 1 | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Lecture#** | **Contents** | **Book Reference (Sections)** | **Assessment** |
| 1 | 1 | Introduction to the Course | Irvine: 2.1.1 |  |
| Introduction to 8086 processor | Art of Assembly: 3.1 |  |
| Address and Data buses and their size | Brey: 2.1 |  |
| Control bus (Read Write and Memory/IO signals) | **Mazidi: 0.2** |  |
| Processor-Memory-I/O Model |  |  |
| 2 |  | Irvine: 2.2.2 |  |
| Basic Computer Architecture | Art of Assembly: 4.1 |  |
| General purpose registers | Brey: 2.1 |  |
| Segment registers | **Mazidi: 1.2** |  |
| Index registers |  |  |
| Purpose of these registers |  |  |
|  |  |  |
| 2 | 3 | Programing Model of X86 | Irvine: 14.1.5 |  |
| Assembler Directives (.model, .stack, .data, .code, .exit), | **Mazidi: 1.4** |  |
| Basic program structure, | **Mazidi: 2.1** |  |
| Data movement among registers using (MOV instruction), | Irvine: 14.2.2 |  |
| Writing Hello World Program |  |  |
| Process Termination (Usage of Int 0x21, Service 0x4c) |  |  |
| 4 | Register addressing mode | Irvine: 12.3.6 |  |
| Immediate addressing mode | Art of Assembly: 4.6 |  |
| Direct addressing mode | Brey: 3.1 |  |
| Indirect addressing mode | Art of Assembly: 4.7 |  |
| Logical to Physical Address Conversion | **Mazidi: 1.6** |  |
| Examples with MOV instruction |  |  |
|  |  |  |
|  |  |  |
| 3 | 5 | ALU instructions | Art of Assembly: 6.5 | A1 |
| Flag register | **Mazidi: 3.1** | Q1 |
| Programming practice questions |  |  |
| 6 | Data types | Art of Assembly: 5.3 |  |
| Variable declaration and initialization | **Mazidi: 2.5** |  |
| ALU operations on variables |  |  |
| 4 | 7 | Arrays & Loops | Art of Assembly: 5.6.1 |  |
| Base plus index addressing mode | Brey: 6.1 |  |
| Base relative plus index addressing mode |  |  |
| 8 | Jump & Branch insturctions | Art of Assembly: 10.1 |  |
|  | Irvine: 6.9.4 |  |
| 5 | 9 | Bit Manipulation | Art of Assembly: 6.6 |  |
| Shift: Arithmetic and Logical Shift | **Mazidi: 3.5** |  |
| Rotations |  |  |
| 10 | Bit Masking | Irvine: |  |
| Selective Set | **Mazidi: 3.6** |  |
| Selective Clear | Art of Assembly: 9.6 |  |
| Selective Complement | Brey: |  |
| 6 | 11 | Stack |  |  |
| Introduction to Stack |  |  |
| Saving and restoring variable in stack |  |  |
| 12 | Subroutines |  | A2 |
| CALL and RET statements |  | Q2 |
| Simple subroutines | Art of Assembly: 8.7 |  |
| Parameter Passing through Stack | Brey: 6.3 |  |
| Return value |  |  |
| 7 | 13 | Subroutines |  |  |
| Parameter passing through stack |  |  |
|  |  |  |
| 14 | Multiplication & Division | Art of Assembly: 6.5.6 |  |
| MUL and DIV Instructions | Art of Assembly: 6.5.7 |  |
| Programming exercise | **Mazidi: 3.2** |  |
|  | **Mazidi: 6.9** |  |
| 8 | 15 | Adjustment |  |  |
| Instructions: AAA, AAD, AAM |  |  |
| Programming exercise |  |  |
|  |  |  |
| 16 | MID TERM |  |  |
| 9 | 17 | MID TERM PAPER REVIEW |  |  |
| 18 | Video Memory | Art of Assembly: 23.1 |  |
| Accessing video memory directly |  |  |
| Familiarize with ASCII and attribute |  |  |
| 10 | 19 | Display Memory |  |  |
| Screen Calculation and Printing |  |  |
| Programming exercise |  |  |
| 20 | String Instructions |  |  |
| Introduce basic string instructions | **Mazidi: 6.2** |  |
| LES and LDS instruction | Art of Assmbly: 6.3.3 |  |
| Programming exercise |  |  |
| 11 | 21 | Software Interrupts | Irvine: 14.1.4 | Q3 |
| Basic introduction of interrupts |  |  |
| BIOS and DOS interrupts for printing on screen and reading keys | **Mazidi: 4.1** |  |
| 22 | Software Interrupts | **Mazidi: 14.1** |  |
| Explain the concept of IVT (IVT) |  |  |
| Writing ISRs |  |  |
| Writing ISR for Interrupt 0x65 |  |  |
| 12 | 23 | Exceptions (Handling) | Art of Assembly: 17.3 |  |
| Divide by Zero |  |  |
| Overflow |  |  |
| 24 | Hardware Interrupts | Art of Assembly: 17.4 | A3 |
| Programmable Interrupt Controller | **Mazidi: 14.3** |  |
| IRQs |  |  |
| Writing ISR for Timer Interrupt |  |  |
| 13 | 25 | Interacting with I/O | Irvine: 2.4.4 | Q4 |
| IN & Out Instructions | Brey: 11.1 |  |
| Application: Reading Time from RTC | Art of Assembly: 17.4.7 |  |
| Programming exercise |  |  |
| 26 | X64 programming | Notes |  |
| Introduction to x64 programming |  |  |
| 64-bit registers |  |  |
| Parameter passing to 64-bit functions |  |  |
| 14 | 27 | x64 programming |  | A4 |
| Writing libraries in assembly language |  |  |
| Calling library function in high level language program |  |  |
| 28 | x64 programming |  |  |
| Programming exercise |  |  |
| 15 | 29 | Exploiting Hardware features: Multi-Media eXtension (MMX) | Brey: 14.5 | Q5 |
| MMX registers |  |  |
| Exploiting data-level parallelism |  |  |
| 30 | MMX |  |  |
| MMX exercise |  |  |
| 16 | 31 | Revision |  |  |
| 32 | Revision |  |  |