KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
*COMPUTER ENGINEERING DEPARTMENT*

COE 205 Computer Organization & Assembly Language

**Term 091 Lecture Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lec#** | **Date** | **Topics** | **Ref.** |
| 1 | S 3/10 | Syllabus. Introduction. Assembly and machine language. | Chapter1 &Int. Computer Org. & Assembly Lang. Prog.(Online) |
| 2 | M 5/10 | Compiler and assembler. Instruction and machine language, Instruction fields: opcode, operands, Assembly vs. Machine code, Translating languages. | Chapter1 &Int. Computer Org. & Assembly Lang. Prog.(Online) |
| 3 | W 7/10 | Advantages of High-Level languages, Why learn Assembly language, Assembly vs. High-level languages. Assembly language programming tools: Assembler, Linker, Debugger, Editor. Programmer’s view of a computer system. Instruction set architecture. I-8086 instruction set architecture. | Chapter1 &Int. Computer Org. & Assembly Lang. Prog.(Online) |
| 4 | S 10/10 | Basic Computer Organization, Processor. Processor, Clock, Memory, Address Space,Address, Data, and Control Bus, Memory Read and Write Cycles, Memory Devices: RAM. Static vs. Dynamic RAM. | Chapter1 & 2 & Int. Computer Org. & Assembly Lang. Prog.(Online) |
| 5 | M 12/10 | Processor-Memory Performance Gap, Memory Hierarchy: Registers, RAM, Cache, Hard disk. Magnetic Disk Storage access time. Data Representation: Binary Numbers, Hexadecimal Numbers, Base Conversions. | Chapter1 &Data Representation(Online) |
| 6 | W 14/10 | Integer Storage Sizes. Binary and, Hexadecimal Addition, Signed Integers. Two’s complement representation, Sign extension, 16’s complement. | Chapter1 &Data Representation (Online) |
| 7 | S 17/10 | Binary and hexadecimal subtraction, Ranges of signed integers, carry and overflow. Character representation: ASCII code. | Chapter1 &Data Representation (Online) |
| 8 | M 19/10 | ASCII code, parity bit. History of Intel Processors. Intel Core MicroArchitecture, CISC vs. RISC processors. IA-32 Registers. | Chapter 2 & Int. Computer Org. and Assembly Lang. Prog. (Online) |
| 9 | W 21/10 | IA-32 Registers, Status Flags. **(Quiz#1)** | Chapter 2 & Int. Computer Org. and Assembly Lang. Prog. (Online) |
| 10 | S 24/10 | Status Flags. Fetch-Execute cycle. Pipelined Execution. | Chapter 2 & Int. Computer Org. and Assembly Lang. Prog. (Online) |
| 11 | M 26/10 | Pipelined Execution, Superscalar Architecture, IA-32 Memory Management, Modes of Operation, Real Address Mode: Logical to Linear Address Translation. | Chapter 2 & Int. Computer Org. and Assembly Lang. Prog. (Online) |
| 12 | W 28/10 | Flat Memory Model. Protected Mode Architecture and address translation, Segment Descriptor Tables, Paging. | Chapter 2 & 3Assembly Language Syntax and Program Structure  (Online) |
| 13 | S 31/10 | Basic Elements of Assembly Language, Flat Memory Program Template, Assembling, Linking, and Debugging Programs. **Variable Declaration**: DB, BYTE, SBYTE, DW, WORD, SWORD, DD, DWORD, SDWORD, DQ, QWORD, DT, TBYTE. | Chapter 3 &Variable, Constant, and Array Declaration (Online) |
| 14 | M 2/11 | **Variable Declaration**: DB, BYTE, SBYTE, DW, WORD, SWORD, DD, DWORD, SDWORD, DQ, QWORD, DT, TBYTE. Defining strings. DUP Operator. Byte Ordering. | Chapter 3 &Variable, Constant, and Array Declaration (Online) |
| 15 | W 4/11 | Defining Symbolic Constants: =, EQU & TEXTEQU Directives. Offset Operator, Align Directive. Type, Lengthof, Sizeof, PTR operators. | Chapter 3 &Variable, Constant, and Array Declaration (Online) |
| 16 | S 7/11 | Label directive. Operand Types. **Basic Pentium Instructions**: MOV, MOVZX, MOVSX, XCHG. Direct Memory Operands, Direct-Offset Operands. | Chapter 4 &(Assembly)Basic Pentium Instructions (Online) |
| 17 | M 9/11 | Direct-Offset Operands.ADD and SUB instructions and their effect on flags. **(Quiz#2)** | Chapter 4 &(Assembly)Basic Pentium Instructions (Online) |
|  | T 10/11 | **Major Exam I** |  |
| 18 | W 11/11 | INC, DEC, NEG, ADC, SBB. Addressing Modes: Immediate, Register, Direct, Register Indirect. | Chapter 4 &(Assembly)Basic Pentium Instructions (Online) |
|  | W 11/11 | **Last Day for Dropping with W** |  |
| 19 | S 14/11 | Addressing modes: Based, Indexed, Based- Indexed. 8086 Addressing modes. Two dimensional array processing. | Chapter 4(Assembly)&Addressing Modes(Online) |
|  | S 14/11(Makeup) | Default segments and Segment Override.LEA instruction. JMP and Loop instruction. | Chapter 4(Assembly)&Addressing Modes(Online) |
| 20 | M 16/11 | No Class. |  |
| 21 | W 18/11 | Nested Loops. Example Programs: Copying a string, Summing an Array. PC relative addressing. Link Library Overview, Calling a Library Procedure, Linking to a Library, Input/Output procedures using book’s library (Irvine32.lib). | Chapter 4 & 5(Assembly)&Addressing Modes(Online) |
|  | 21/11-2/12 | **Eid Adha Holiday** |  |
| 22 | S 5/12 | Input/Output procedures using book’s library (Irvine32.lib), **Introduction to Stack. Stack instructions**: Push and Pop. Uses of the Runtime Stack: Temporary Storage of Registers. | Chapter 5(Assembly) &Stack & Procedures(Online) |
| 23 | M 7/12 | Nested Loop, Saving Return Address. Stack Instructions: pushad, pusha, popad, popa, pushfd, pushf, popfd, popf. **Defining and Using Procedures**, Call & RET Instructions. Parameter Passing, Passing Parameters in Registers. Preserving Registers, USES Operator. Parameter Passing Through Stack. | Chapter 5(Assembly) &Stack & Procedures (Online) |
| 24 | W 9/12 | Parameter Passing Through Stack, Call & Return Instructions, Freeing Passed Parameters from Stack, Local Variables, Stack Frame. | Chapter 5(Assembly) &Stack & Procedures (Online) |
| 25 | S 12/12 | Call & RET Instructions. Program Design using Procedures, Integer Summation Program, Example of a Recursive Procedure: Factorial. **(Quiz#3)** | Chapter 5(Assembly) &Stack & Procedures (Online) |
| 26 | M 14/12 | **Logical Instructions**: AND, OR, XOR**,** Test, NOT. Applications of logic instructions. | Chapter 6(Assembly)& Flow Control Instructions(Online) |
| 27 | W 16/12 | **Flow Control Instructions**: CMP Instruction. Conditional Jump instructions: Signed and Unsigned. Conditional Jump instructions: Single-flag jump instructions. Jumps Based on Equality, Jumps Based on Unsigned Comparison, Jumps Based on Signed Comparisons | Chapter 6(Assembly)& Flow Control Instructions(Online) |
| 28 | S 19/12 | BT Instruction, Conditional Loop Instructions: Loope/Loopz, Loopne/Loopnz.High-Level Decision Control Structures: IF-Then-Else. | Chapter 6(Assembly)& Flow Control Instructions(Online) |
|  | S 19/12 (Makeup) | Compound Expression with AND. Compound Expression with OR, WHILE Loops, Indirect Jump. | Chapter 6(Assembly)& Flow Control Instructions(Online) |
| 29 | M 21/12 | No Class. |  |
| 30 | W 23/12 | Indirect Jump, Switch Statement. Bubble Sort Algorithm. | Chapter 6(Assembly)& Flow Control Instructions(Online) |
|  | W 23/12 | **Last Day for Dropping all Courses with W** |  |
| 31 | S 26/12 | **Shift Instructions**: SHL, SAL, SHR, SAR. Applications of using shift instructions in performing multiplication & division. | Chapter7(Assembly)& Bit Manipulation Instructions(Online) |
| 32 | M 28/12 | **Rotate Instructions**: ROL, ROR, RCL, RCR. SHLD instruction. SHRD instructions. | Chapter7(Assembly)& Bit Manipulation Instructions(Online) |
| 33 | W 30/12 | Shift & Rotate Applications. **(Quiz#4)** | Chapter7(Assembly)& Bit Manipulation Instructions(Online) |
| 34 | S 2/1 | Shift & Rotate Applications. **Multiplication Instructions**: MUL, IMUL. | Chapter7(Assembly)& Advanced Arithmetic(Online) |
| 35 | M 4/1 | Pentium IMUL instructions. **Division Instructions**: DIV, IDIV. Sign Extension Instructions: CBW, CWD, CDQ. | Chapter7(Assembly)& Advanced Arithmetic(Online) |
|  | T 5/1 | Translating Arithmetic Expressions, Decimal string to number conversions. Number to decimal string conversion. **(Quiz#5)** | Chapter7(Assembly)& Advanced Arithmetic(Online) |
| 36 | W 6/1 | No Class.  |  |
|  | Th. 7/1 | **Major Exam II** |  |
| 37 | S 9/1 | **Introduction to Interrupts**. Interrupts vs. procedures.Types of Interrupts: Hardware. Software, Exceptions. Maskable and non-maskable Interrupts, Interrupt Flag. | Interrupts (Online) |
| 38 | M 11/1 | Interrupt Processing, IDT, Trap Flag, Dedicated Interrupts. | Interrupts (Online) |
| 39 | W 13/1 | **CPU Design**: Register Transfer. **Data-Path Design**. Connecting Registers using Muxs. | CPU Design (Online) |
| 40 | S 16/1 | **Data-Path Design**. Connecting Registers using a tri-state bus. Register Transfer Timing. | CPU Design (Online) |
| 41 | M 18/1 | **Single-Bus CPU Data path design**: Fetch Control Sequence. Synchronous vs. Asynchronous Memory Transfer, Execution control sequence for ADD with register indirect addressing mode. Execution control sequence for unconditional JMP. | CPU Design (Online) |
| 42 | W 20/1 | **Single-Bus CPU**: Execution control sequence for Conditional JMP, INC [R1], CMP, Loop instruction. Performance Considerations. **Two-Bus CPU**: Fetch Control Sequence, Execution Control Sequence for ADD with register indirect addressing mode, unconditional and conditional Jump. | CPU Design (Online) |
|  | W 20/1 | **Dropping all Courses with WP/WF** |  |
| 43 | S 23/1 | **Three-Bus CPU.** Fetch Control Sequence, Execution Control Sequence for ADD. unconditional and conditional Jump Instructions. **Control Unit Design: Hardwired Control Unit,** General Hardwired Control Unit Organization, Generation of Control Signals. | CPU Design (Online) |
| 44 | M 25/1 | Deriving Rout & Rin Signals. **CPU-Memory Interface Circuit.** | CPU Design (Online) |
| 45 | W 27/1 | **Microprogrammed Control Unit Design*:*** Control Word, Control Store, Micro Program Counter, Micro Instruction Register, Microinstruction, Microroutine. General Organization of Micro-programmed Control Unit, Branching Address, Sequencer. | CPU Design (Online) |