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

**COE 205 Computer Organization & Assembly Language**

**Proposal For**
**Lecture Breakdown**
**for the COE 205 Course**

| Week | Lecture | Topics | Ref. | Lab | |
|---|---|---|---|---|---|
| | | | | **Experiment** | **Content** |
| | | | | | |
| 1 | 1 | Syllabus. Introduction. Computer System, CPU, Memory, CPU-Memory Interface, Data Bus, Address Bus, Control Bus. Machine type. | Chapter1 (Organization) & Chapter 1 (Assembly) | **Assembly Language Programming**<br>• High-Level programming versus Low-Level programming,<br>• Machine Code<br>• Fetch-Execute cycle<br>• Description of the adopted tool | |
| | 2 | Memory Hierarchy: Registers, Cache, RAM, Hard disk, Tape.<br>Machine Language, Assembly Language.<br>Instruction Formats, Opcode, Operands. | | | |
| | 3 | Stored Program Concept, Fetch-Execute Cycle, Instruction Pointer. Instruction Register.<br>Data Typing in High Level and Assembly Language, Why assembly language programming, Assembler, Linker, Debugger.. | | | |
| | | | | | |
| 2 | 4 | Programmer's view of the computer, Instruction Set Architecture (ISA). i8086 processor characteristics. Control Unit& Datapath.<br>Interfacing the CPU to memory & I/O. Types of Buses. One and Two-bus Architectures. | Chapter1 (Organization) & Chapter 1 (Assembly) Appendix A & E (Assembly) Appendix A & E (Assembly) | **Introducing the MASM environment**<br>• Assembling Linking and running a program<br>• Using Code View<br>• 8086 Registers | |
| | 5 | Number representation: Binary, Hex., base conversion, Signed number representation: Sign-magnitude,1's complement, 2's complement.<br>Ranges. Overflow detection for unsigned and signed numbers. Finding 16's and 15's complement. Character Representation. ASCII Code. Even and Odd Parity. | | | |
| | 6 | Assembly language syntax & Program Structure. Overview of 8086 instructions. **(Quiz#1)** | | | |

| | | | | |
|---|---|---|---|---|
| 3 | 7 | Variable Declaration: DB, DW, DD. Offset and PTR operators. Constant declaration using EQU. | Chapter3 (Assembly) Chapter3 (Assembly) Section 12.5 & (Assembly) | **Program Structure and Data Representation (1/2)**<br>• Data representation,<br>• Variable & constant declaration<br>• ASCII code,<br>• ADD & SUB instructions |
| | 8 | Array declaration and referencing. DUP Operator. Input/Output using INT 21H. Reading a character with and without echo, displaying a character, displaying a string. | | |
| | 9 | Reading a string. Examples on the use of input/output using INT21H. **(Quiz#2)** | | |
| | | | | |
| | 10 | Memory Segmentation, Logical & Physical Addresses. | Chapters 2, 3 & 5 (Assembly) | **Data Representation (2/2)**<br>• Variable & constant declaration<br>• ASCII code<br>• Array declaration: the DUP operator |
| | 11 | Addressing Modes: Immediate, Register, Direct. Register-Indirect, Based, Indexed, Based-Indexed. | | |
| | 12 | IA32 Registers. Pentium Addressing modes. **(Quiz#3)** | | |
| | | | | |
| 5 | 14 | Status & Flags Register. Basic Assembly Instructions: MOV instruction. | Chapter 3 & 5 & 6 (Assembly) | **Input and Output and Basic Instructions**<br>• Interrupts (vs. Methods)<br>• INT 21 H, functions: 01, 02, 06 , 08, 09, 0AH<br>• Basic Assembly Language Instructions: ADD, ADC, SUB, SBB, INC, DEC, LOOP, Procedures |
| | 15 | Basic Assembly Instructions: XCHG, LEA, MOVZX, MOVSX, XLATB, ADD, SUB, INC, DEC. NEG, CMP. | | |
| | 16 | Basic Assembly Instructions: ADC, SBB. Introduction to flow control instructions. | | |
| | | | | |
| 6 | 17 | Multiplication Instructions: MUL, IMUL. Pentium IMUL instructions. Division Instructions: DIV, IDIV. | Chapters 3 & 6 & 8 (Assembly) | **Segmentation and Addressing Modes**<br>• Segmentation<br>• Addressing Modes<br>• Indexing and<br>• Data Manipulation: Array Manipulation |
| | 18 | Division Instructions: DIV, IDIV. Sign Extension Instructions: CBW, CWD, CDQ, CWDE. | | |
| | 19 | Applications of multiplication and division instructions.<br>Logical Instructions: NOT, AND, OR, XOR, Test. | | |

| | | | | |
|---|---|---|---|---|
| 7 | 20 | Logical Instructions: NOT, AND, OR, XOR, Test. Applications of logic instructions. Shift Instructions: SHL, SAL, SHR, SAR | Chapters 3,4, 7& 8 (Assembly) | **Branching Instructions**<br>• Unconditional jumps<br>• CMP Instruction<br>• Conditional Jumps: Flag based, Sign and Unsigned comparisons |
| | 21 | SHLD, SHRD. Rotate Instructions: ROL, ROR, RCL, RCR. Applications of shift & rotate instructions. **(Quiz#4)** | | |
| | 22 | Applications of using shift instructions in performing multiplication & division. Flow Control Instructions: Unconditional JMP. Types of jump target: Short, Near, Far. | | |
| | | | | |
| 8 | 23 | Conditional Jump instructions: Signed and Unsigned. Single-flag jump instructions. Loop Instructions: Loop, Loope/Loopz, Loopne/Loopnz. High-Level Decision Control Structures: IF-Then-Else, For Loop, While Loop, Repeat Until, Case Statement. | Chapter 4, 7,10 (Assembly) | **Arithmetic Instructions**<br>• ADD, ADC, SUBB, SBB<br>• MUL, IMUL<br>• DIV, IDIV |
| | 24 | Indirect Jump example. The Stack: PUSH and POP instructions. | | |
| | 25 | Stack instructions: PUSHA, POPA, PUSHAD, POPAD, PUSHF, POPF, PUSHFD, POPFD. Introduction to Procedures: CALL and RET. Procedure Definition. | | |
| | | | | |
| 9 | 26 | Passing Parameters to Procedures. RET n. Linking procedures defined externally using EXTRN and PUBLIC. Making a library of Procedures using LIB. Introduction to Macros. Macro Definition & Expansion. Pseudo parameters in macros. | Chapter 9 & 10 (Assembly) | **Logical Instructions**<br>• NOT<br>• AND, OR, XOR<br>• TEST<br>• Binary number manipulation |
| | 27 | Macros versus procedures. Macro Library. Examples of Macros. | | |
| | 28 | Macro Library. Examples of Macros. **(Quiz#6)** | | |

| | | | | |
|---|---|---|---|---|
| 10 | 28 | Repetitive macros: REP, IRP. Useful examples of Macros. Macro Conditionals: IF, IFE, IFB, IFNB. String Instructions: MOVS, MOVSB, MOVSW, MOVSD. REP Prefix. | Chapter 9 & 12 (Assembly) | **Shift Rotate Instructions**<br>• SHL, SHR<br>• SAR, SAL<br>• ROL, ROR<br>• RCL, RCR<br>• Binary number manipulation<br>• |
| | 29 | String Instructions: CMPS, CMPSB, CMPSW, CMPSD, SCAS, SCASB, SCASW, SCASD, LODS, LODSB, LODSW, LODSD, STOS, STOSB, STOSW, STOSD. REPE, REPNE Prefixes. Applications of string instructions. | | |
| | 30 | Input/Output: IN and OUT instructions, Direct and Indirect I/O. Introduction to Interrupts. Difference between interrupts and procedures. Types of Interrupts: Hardware. Software, Exceptions. Maskable and non-maskable Interrupts, Interrupt Flag, Trap Flag, Interrupt Vector Table, Interrupt Mechanism, Bios and DOS interrupts. Interrupt Instructions: INT, IRET, INTO. | | |
| | | | | |
| 11 | 31 | CPU Design: Control unit and Data Path. Register Transfer. Data-Path Design. Connecting Registers using Multiplexers. | Chapter 2 & 4 (Organization) | **Subroutine Handling Instructions and Macros**<br>• Stack<br>• Stack handling Instructions<br>• Subroutines and Subroutine Calls<br>• External Subroutines |
| | 32 | Connecting Registers using a tri-state bus. Examples of register transfer: MOV, XCHG. | | |
| | 33 | Single-Bus CPU Data path design. Single-Bus CPU design: Fetch Control Sequence, Synchronous vs. Asynchronous Memory Transfer | | |

| | | | | |
|---|---|---|---|---|
| 12 | 34 | Execution Control Sequence for MOV, ADD, ADD with register indirect addressing mode. Execution Control Sequence for unconditional Jump. | Chapter 2 & 4 (Organization) | **String Handling Instructions**<br>• Direction Flag<br>• Transfer Instructions<br>• Compare Instructions<br>• Repeat Prefixes<br>• |
| | 35 | Execution control Sequence for conditional Jump (JS), CMP, LOOP, SHR, INC[BX], MOVSB. | | |
| | 36 | Register Transfer Timing: Estimating Minimum Clock Period. Performance considerations. Two-Bus CPU: Fetch Control Sequence, Execution Control Sequence for ADD. Speedup of 2-bus vs. 1-bus CPU. | | |
| | | | | |
| 13 | 37 | Two-Bus CPU: Execution Control Sequence for unconditional and conditional Jump Instructions. Three-Bus CPU. | Chapter 2 & 4 (Organization) | **Interrupts**<br>• Conceptes of Interrupts<br>• Software vs. Hardware Interrupts<br>• BIOS<br>• Writing Interrupt service routines (ISR)<br>• Writing TSR programmes |
| | 38 | Three-Bus CPU: Fetch Control Sequence, Execution Control Sequence for ADD, unconditional and conditional Jump Instructions. | | |
| | 39 | Control Unit Design: Hardwired Control Unit, General Hardwired Control Unit Organization, Generation of Control Signals. Deriving Rout & Rin Signals. | | |

| | | | | |
|---|---|---|---|---|
| 14 | 40 | CPU-Memory Interface Circuit. ***Microprogrammed Control Unit Design:*** Control Word, Control Store, Microinstruction, Microroutine. | Chapter 2&4 (Organization) | **Accessing Video Memory**<br>• Video Memory<br>• INT 10H<br><br>**Using The Mouse**<br>• INT 33H<br>• Mouse Basic Functions |
| | 41 | General Organization of Microprogrammed Control Unit, Micro-Program Counter, Micro-Instruction Register, Sequencer, Branching Address. Horizontal, Vertical, Field-encoded control store. | | |
| | 42 | Microroutine for Add instruction with 8 addressing modes. Wide branch addressing and Multiway branching. Bit Oring. Comparison of Hardwired vs. Microprogrammed control unit. Simple CPU Design: datapath design, hardwired and microprogrammed control unit design. | | |
| | | | | |
| 15 | 43 | Review ? | | • **Project Submission** |
| | 44 | Review ? | | |
| | 45 | Review ? | | |