NEW YORK CITY COLLEGE OF TECHNOLOGY
The City University of New York

DEPARTMENT: Electrical and Telecommunications Engineering Technology

SUBJECT CODE: EET2262
AND TITLE: Digital Electronics II

COURSE DESCRIPTION: Theory and practice using the hardware and software of an Intel microprocessors are the main topics of the course. The microprocessor is integrated into a single board to form a prototype of a digital Von Neumann computer prototype. The hardware (building blocks) of a general digital computer prototype is studied with details, such as the CPU, memory (ROM and RAM), inputs and outputs.
The internal components of the CPU (microprocessor) are analyzed, such as the ALU (Arithmetic Logic Unit), clock, CPU registers, PC, Interrupt Services, Stack memory, etc.
Machine and Assemble languages are introduced and used, as well as the standard Assembly mnemonics and commands used by the industry (Intel Corp). Practical applications such as: digital counters, clocks, delays, displays, multiplexers, encoders/decoders, controllers are part of many laboratory exercises.

PREREQUISITE: EET2162/ET 382
PRE- or COREQUISITE:
TEXTBOOK:
1. Class Notes/ by Misza Kalechman.
2. Laboratory Manual/by Misza Kalechman.
3. Simulators
4. 8080/8085 Assembly Language Programming Manual /Internet.
Supplemental:
5. Microcomputer Theory & Applications by Rafiquzzaman, Publisher Prentice Hall.
6. The 8088 and 8086 Microprocessors, by Walter Triebel and Avtar Singh, Second Edition, Prentice Hall,1997.

COURSE OBJECTIVES: Upon completion of this course the student will possess the ability to:

COURSE OUTCOMES: 1. Know the structure and components of a typical microprocessor and/or **(ETAC/ABET)** minicomputer. (ABET Criteria: 3.1,3.4,3.5, criterion 5)
2. Write, run, and test (single step/analysis) a flowchart and a source code program (ABET Criteria:3.1,3.4,3.5, criterion 5)
3. Be able to follow the logic and flow of information of a program. (ABET Criteria:3.1,3.4,3.5, criterion 5)
4. Use the many diagnostic facilities (hardware and software) of typical microprocessor training unit. (ABET Criteria: 3.1,3.4,3.5, criterion 5)
5.Use the microprocessor to solve an array of typical practical

problems such as timing, synchronization, control of the

displays: address and data fields (using the 7 segments led), and encoding/decoding information (ABET Criteria 3.1,3.4,3.5, criterion 5)

6. Using the microprocessor clock be able to calculate the execution/processing time of an application program.
7. Design delays, and use standard delay techniques in controlling a variety of possible applications associated with different synchronous applications (ABETCriteria 3.1,3.4,3.5, criterion 5)
8. Set conditions, assumptions and limitations of systems specs and implementation. (ABET Criteria: 3.1,3.4,3.5, criterion 5)

TOPICS: Topics include implementation and timing calculations, controlling and/or outputting a display. Block and schematic diagrams including the wiring diagram, of the hardware and software implications of a typical microprocessor and/or a minicomputer using the Von Neumann model are explored; and a variety of practical applications are studied. Memory consideration (ROM, RAM), stack organization, internal wiring and organization of a microprocessor and the three busses (address, data and control) are analyzed with details, as well as the Maskable and hardware interrupt services. CPU registers, and the general architecture of the microprocessor including the status flags (F register) are studied and used in different practical lab applications. Lab practices include; clock considerations, displaying a moving and flashing message, decimal counters (up and down) with variable timing, synthesizers, routers, encoders and decoders, modulators, multiplexers, etc.

CLASS HOURS: 2

LAB HOURS: 3
CREDITS: 3

Prepared by: Professors M. Kalechman,
Spring 2025

Course Coordinator: Misza Kalechman
(718) 260-5318
Mkalechman@citytech.cuny.edu

Description and details of the laboratory course work:

The student is first introduced to the (Intel) microprocessor architecture and its basic components, and a simple model of a minicomputer prototype using the von Neumann processing model. The basic components are presented and discussed: such as input devices (keyboard) and the outputs given by the data and address fields, memories (R0M and RAM), the arithmetic logic unit (ALU), the monitor System,

the three communication busses, compilers and translators. Software is introduced starting with machine language, followed by assembly language, followed by high level languages, and its corresponding interfaces, overheads and costs are discussed. Basic hardware and software design applications such as counters (up and down), digital clocks, moving and flashing messages, routes, encoders, etc. are designed-implement and tested in the laboratory.

Contribution of course to meet the requirements of Criteria 5
EET2262, Microprocessor interface, meets Criterion 5, by providing students with a strong foundation of theoretical and practical laboratory principles and skills needed to analyze and design practical digital electronic (microprocessor) applications in diverse areas such as calculators, controls, modulators, multiplexers, etc. Academic benchmarks, course outcomes, and assessment requirements have been established to evaluate student comprehension of concepts, and skills. By also fostering critical thinking, communication (verbal and written), as well as team work, student develop the knowhow needed to solve problems in an academic/industrial environment, which will later serve them well in their professional workplace.

GRADING POLICY:

Exam #1	25%
Exam #2	10%
Lab Reports	25%
Final /Exam	40%

Letter Grade	Numerical Grade Ranges	Quality
A	93-100	4.0
A-	90-92.9	3.7
B+	87-89.9	3.3
B	83-86.9	3.0
B-	80.82.9	2.7
C+	77-79.9	2.3
C	70-76.9	2.0
D	60-69.9	1.0
F	59.9 and below	0.0

Assessment The following assessment techniques are correlated to the course objectives as follows: In addition, each assessment technique incorporates one or more of the following ABET Criteria 3 outcomes: 3.1,3.4,3.5, criterion 5. **Course Objectives** 1. Understand the main building blocks of a microprocessor and a digital van Neumann model computer:3.1,3.4,3.5, criterion 5. 2. Know the electronic building blocks of a microprocessor 3. Know the building blocks of a minicomputer 4. Write, run and test (be able to analyze) an Assembly programs 5. Know the interrelation between the hardware and the software using machine language when using the Arithmetic and logic groups 6. Be able to follow the logic and flow of Information in a program using mnemonics, OC and Assembly language.	**Assessment** Using an Intel chip (8085/8086/8088) microprocessor the student should be able to: 1.1 Identify the electronic chips and the general wiring diagram of a typical minicomputer. 1.2 Know the functions and components of the CPU (ALU, general CPU Registers, memory address registers, status registers, the clock, PC) 1.3 Know the memory organization (ROM-RAM) and the protocol/procedure to access information 2. Draw a block box diagram of a microprocessor (hardware/ software). 2.1 Draw a schematic diagram of the 8 and 16 bits microprocessor. 2.2 Draw a logic diagram of the microprocessor. 3. Draw a block diagram of a typical minicomputer/van Neumann model 4. Use the data transfer group to: 4.1 Move information from a CPU register to memory. 4.2 Move information from memory to a CPU register 4.3 Move information from a register to another register 4.4 Draw a flow-chart of the data transfer group 4.5 Define and use a memory Address Register to move information. 5. Use the Arithmetic and Logic transfer groups to: 5.1 Understand and be able to use the Arithmetic instructions: ADD, SUB 5.2 Understand the function of the status F register. 5.3 Demonstrate skills by checking the flags such as the

	carry bit (overflow), parity, sign, zero and Ac bits. 5.4 Know that the negative numbers are represented in complement form 5.5 Implement the Boolean Operations: And, Or, Not-Xor 6. Integrate all the transfer, Arithmetic and Logic instructions into a program
7. Decision making using machine language and delay calculations 8.System specs machine control instructions	7. Know the (conditional and unconditional) branch instructions 7.1 Implement and analyze a delay subroutine (status flag) 7.2 Construction of 0.5 sec. delays 7.3 Construction of larger delays 8.Machine Control instructions 8.1 Stack pointer operations 8.2 Push and Pop operations 8.3 Vector Interrupt services 8.4 Learn skills in analyzing system specs, limitations and errors

s

WEEK	THEORY	Research	LAB
1	1. The hardware and software components of the microprocessor trainer: SDK-85 2. Numbers: binary, decimal, octal, hexadecimal 3. The Binary System, Boolean operators 4. Combinational circuits (analysis /synthesis) 5. Sequential circuits (Analysis /synthesis) 6. Flip flops.	Review binary & hexadecimal number systems Combinational circuits/Truth tables SOP and POS implementations. Memories FF (SR, JK, T, D) Sequential machines/State tables and state diagrams Reading Assignment: Class Notes, Chapter #6	Explore and discuss the building blocks of a general purpose digital computer where the Intel 8085/8086/8088 microprocessor are the CPUs.
2	1..The main building blocks of a digital computer: CPU, MEM, I/O, 2.CPU functions: ALU, CU. 3. Inputs and outputs 4.Languages (Machine, Assembly, High Level) 5. Translators and Compilers	The assembler table/ Appendix F. Storing data in the CPU Registers and RAM locations. Explore the keyboard and outputs: Control and data fields Reading Assignment: Class Notes, Chapter #1/section 1-6 Reading Assignment: Class Notes, Chapter #7	Experiment 1 Introduction to a typical training unit of a minicomputer and the corresponding simulation software HMW#1
3	1. CPU registers: A (accumulator), B, C, D,E,H,L 2. The program counter, Stack pointer. 3. Data transf. group and the ADD instruction 4. Utilizing the CPU registers to perform addition	Write a program that adds numbers Review complements in different bases (10,8,16,2), use the 2's or the 1'sxcomplement arithmetic to handles negative numbers. Reading Assignment: Class Notes Chapter #1/section 7-10	Experiment 2 Addition Program and single stepping.
4	1. Relationships between binary, 2's 1's complement, 2. The complements in decimal, and hexadecimal bases. (and any base). 3. The 1's complement and 2's complement to represent negative numbers 4. Illustrate using arithmetic operations	The data transfer group MOV,MVI,STA,LDA, LXI,STAX,LDAX, XCHG Reading Assignment/ Class Notes Chapter #13/section 8-9.	Experiment 3, The Double precision addition/ DAD, Single Stepping a Program.
5	1. Review single stepping and the concept of examining registers and memory locations after each instruction of a given program is executed	The Arithmetic group: ADD, SUB, The F Register to check results. Negative numbers. Reading Assignment/ Class Notes Chapter #13/section 9-11.	Single Stepping cont'd-use of the DAA command to perform BCD operations. Experiment 4/Negative numbers
6	1. Subroutines 2. The logic group (ANA, ORA, XRA, CMP, ROTATE with and without Cy) 3. The F register. Flags (C, Ac, P, Z, S) 4. Conditional and unconditional Jumps. 5. The stack organization (PUS, POP)	Typical Subroutines: UPDAD, UPDAT, OUTPUT, RDKBRD and DELAY Reading Assignment/ Class Notes Chapter #12/section 1-5.	Experiment 5. The Delay Loops (Flags and Decision Making) Display Subroutines.

7	1. Flowcharts and assembly language programs for a time delay. 2. Calculating the execution time of a delay program from the clock's frequency and cycles per instruction. 3. Subroutines. 4. **Midterm Exam**	Subroutine Reading Assignment Chapter #13/section 12-13 Chapter #12/section 6-9	Experiment 6. Delay subroutines

S

WEEK	THEORY	HOMEWORK	LAB
8,9	1. Details of Controlling the SDK-85 (7-segment displays) 2. Using subroutines 3. Passing parameters to subroutines via CPU registers. 4. The DELAY and OUTPT subroutines.	Using the Output and Delay subroutines to display the Messages PASS, FAIL, BLANKS by calling Output just once, with different delays Reading Assignment/ Class Notes Chapter #13/section 16-18.	Experiment 7 Manipulations of blocks of bytes Experiment 8 Simulators
10	Up counters, down counters (Push, Pop, Output, delays) Exam #2	Counters flow chart and implementation	Experiment 9 The Outpu subroutine PASS, FAIL, BLANKS
11/12	Applications of the processor in telecom. (MUXIS, Sinusoidal, PAM, Synthesis of Signals)	4X1 MUX 4X1 de max PAM signals NYQUIST Sampling Reading Assignment/ Class Notes Chapter #14 section 1-4.	Experiment 12 Up and down count with variable delays (project)
13	1. INTERRUPT 2. TRAP, RSST 5.5, 6.5. 7.5 3. EI, Interrupt masks 4. SIM, RIM. INSTRUCTIONS Exam	INTERRUPT Services Reading Assignment / Class Notes Chapter #14/section 9-10	Experiment 13 Counter with variable delays
14	Parity Encoder. Parity decoder	• Flowcharting • Communication Protocols • BSC Model	Experiment 14 Interrupt services/ chronometer
15	Review/final Exam		Final Exam