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

DEPARTMENT: Electrical and Telecommunications Engineering Technology

SUBJECT CODE AND TITLE: TCET2242/TC 430 Microcomputer interfacing

COURSE DESCRIPTION: An introduction to assembly language and programming of microprocessors. The design of memory and microprocessor I/O interfaces. Laboratory experiments provide training in interfacing the microprocessor to D/A and A/D converter. Serial and parallel ports are studied and practical experiments utilizing switches, LED's and speakers are performed.

PREREQUISITE: EET2162/ET 382

TEXTBOOK: Microcomputer Theory & Applications
By Rafiquzzaman, Publisher Prentice Hall

COURSE OBJECTIVES/ COURSE OUTCOMES: Upon completion of this course the student will posses the ability to:

1. Know the basic components of a microprocessor and minicomputer. (ABET Criteria 2a, 2b, 2c, 2d, 2e, 2f, 2h,2 k, 2I, 2m)
2. Write, run, single step and flow chart a source code program(ABET Criteria 2a,2b,2c,2d,2h,2I,2m)
3. Follow the logic and flow of information in a program. (ABET Criteria 2a, 2b, 2c, 2d, 2e, 2h, 2I, 2m)
4. Use the many facilities (hardware and software) capabilities of the microprocessor. (ABET Criteria 2a, 2b, 2c, 2d, 2e, 2f, 2h, 2I, 2m)
5. Use the microprocessor to solve an array of typical practical problems (timing, control and output). (ABET Criteria 2a, 2b, 2c, 2d, 2e, 2f, 2I, 2m)
6. Calculate the delay, and output considerations associated with different synchronous applications(ABET Criteria 2a, 2b, 2c, 2d, 2e, 2f, 2I, 2m)
7. Set conditions, assumptions and limitations of systems specs and implementation. (ABET Criteria 2a, 2b, 2c, 2d, 2e, 2I, 2m)

TOPICS: Topics include timing, controlling, outputting and displaying considerations. Block and schematic diagrams,

of the hardware and software. Memory consideration (ROM, RAM), stack organization, internal wiring of the microprocessor and busses. Registers, and the general architecture of the microprocessor including the status flags and interrupts are analyzed. Applications include; clock design, moving and flashing messages, counters (up and down) with variable times, synthesizers, encodes/decoders muxs/de-muxes, routing information, etc.

CLASS HOURS: 2
LAB HOURS: 3
CREDITS: 3

Prepared by: Professors M. Kalechman
November 2006

Course Coordinator: Misza Kalechman
mkalechman@citytech.cuny.edu

Description details for laboratory course work:
The student is first introduced to the microprocessor, and the simple model of a minicomputer. The basic components are introduced :such as input /output memory (R0M and RAM), ALU, Operating System, busses, compilers and translators. Software is introduced such as machine and assembly language, and extensively used to design practical applications such as counters (up/down), digital clocks, moving and flashing messages, routes, etc.

GRADING POLICY:

Exam #1	20%
Exam #2	20%
Lab Reports	15%
Term Project	10%
Final Examination	35%

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 2 outcomes (2a, 2b, 2c, 2d, 2e, 2f, 2k, 2m).	
Course Objectives	**Assessment**
1.Understand the main building blocks of a microprocessor and digital minicomputer.	Using the Intel 8085 (SDK-85) microprocessor the student should be able to: 1.1 Identify the chips and the general wiring diagram that make up the SDK-85. 1.2 Know the functions of the CPU (ALU, CPU Registers, Clock) 1.3 Know the memory organization 1.4 Draw a block diagram of the microprocessor (hardware/ software). 1.5 Draw a schematic diagram of the microprocessor. 1.6 Draw a logic diagram of the microprocessor.
2. Write, run and analyze Assembly programs	2.1 Move information from a register to memory. 2.2 Move information from memory to registers. 2.3 Draw a flow chart 2.4 Define and use a memory Address Register.
3. Follow the logic and flow of information in a program..	3.1 Understand the Arithmetic instructions ADD, SUB 3.2 Understand the status register. 3.3 Demonstrate their skill by check the carry bit (overflow), parity and Ac bits.

6.Delay calculations	6.1 Comprehension of a delay loop(status flag) 6.2 Construction of 0.5 sec. delays 6.3 Construction of larger delays 6.4 Stack pointer operations 6.5 Push and Pop operations 6.6 Vector Interrupt routines
7. System specs machine control instructions	7.1 Display skills in analyzing system specs, limitations and errors

WEEK	THEORY	HOMEWORK	LAB
1	1. The main building blocks of a digital computer: CPU, MEM, I/O, 2. CPU functions: ALU, CU. 3. ALU functions: Boolean Logic, addition, complementing. 4. Difference between a μP, and a computer. 5. Machine, assembly, and high order languages.	Read Chapter 1 Chapter 2, pgs 17-23 Chapter 4, pgs. 136-144 Review binary & hexadecimal number systems(see Appendix E)	Observe how the SDK-85 implements the bldg. Blocks of a digital computer.
2	1. Memory: RAM, ROM 2. Bits, bytes, and words. 3. SDK-85 word size=8 bits 4. SDK-85 address size = 16 bits 5. The SDK-85 display has a 4-HEX digit address and a 2-HEX digit data display field. 6. 3-BUS architecture 7. SDK-85 keyboard-4 modes of operation: SUBST MEM, EXAM REG, GO, SING STEP	Read: Chapter 2 pgs. 23 (bottom)-33, 38-43, 47-63 Chapter 4 pgs.119-124, 148-161, Appendix F. Write a program that will add numbers located in memory locations 2050 and 2051 and place the sum at address 2052	Experiment 1 Operating the SDK-85.
3	1. CPU registers: accumulator, instruction register, program counter, data counter. 2. Utilizing the 8085 CPU registers to perform addition 3. The Assembler (see Appendix F pages 638-647)	Review 2's complement arithmetic and chapter 2-how a computer handles negative numbers. Read: Chapter 3 pages 97-115	Experiment 2 A Simple Addition Program.
4	1. Relationships between binary, 2's 1's complement, decimal, and hexadecimal numbers. (and only base). - 1's complement and 2's complement - negative numbers	Read: Chapter 2 pgs.33 (bottom)-36. Chapter 3 pg.97(flowcharts) Chapter 4 pgs.126-127 (jumps). Chapter 5 pgs.210-211 (key bounce) Chapter 3/21,23 Chapter 4/16, 17	Experiment 3, Single Stepping A Program.
5	1. Review single stepping and the concept of examining registers and memory after each instruction	Read: Chapter 3, pgs. 86-91 Chapter 2, pgs. 36-37 (stack pointer) Appendix F pgs. 686-699) Chapter 4 pgs. 172-176.	Single Stepping cont'd-use of the DAA command to perform BCD operations.
6	1. The F register 2. Flags (C, Ac, P,Z,S) 3. Conditional Jumps. 4. Introduction to flowcharting and flowchart symbols.	Read the description of the OUTPT subroutine given in Exp. 5	Experiment 4 Delay Loops (Flags and Decision Making) Display Subroutines.
7	1. Flowcharts and assembly language programs for a time delays. 2. Calculating a delay program's run-time from clock frequency and cycles per instruction data. 3. Subroutines. Midterm Exam	Develop a flowchart to output a 4-letter message to the address field of the display. (use is to be made of the DELAY subroutine in ROM to accomplish this timing control).	Experiment 5 Creating A Flashing Message (Utilizing Subroutines) (PASS, FAIL, BLANKS)

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.	Read: Chapter 4 pgs. 146-158.	Experiment 6 Creating a moving message. COLA
10	Up counters, down counters (Push, pop, output, delays) Exam #2	Flowchart counters	Experiment 7 MOD IO up counters with fix delays
11/12	Applications of the N processor in telecom. (MUXIS, PAM, Synthesis of Signals)	4X1 MUX 4X1 de max PAM signals NYQUIST Sampling	Experiment 8 MOD IO Up counter 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 #3	Read pg. 40 (INTERRUPT) Chapter 6, pgs. 223-257	Experiment 9 Up/down counter with VI
14	I/o, Parity Encoder. Parity decoder Communication Project	• Flowcharting • Communication Protocols • BSC Model	Experiment 10 Even Parity Encoder. Experiment 11 Even Parity decoder
15	Review		Final Exam