

COLLEGE OF SAN MATEO

COURSE OUTLINE

COURSE TITLE Introduction to Microprocessors DEPT./NUMBER ELEC 310UNITS OF CREDIT 3.0 LEC. HOURS/WEEK 2 COURSE LENGTH SemesterDEGREE APPLICABLE yes LAB. HOURS/WEEK 3NON-DEGREE APPLICABLE OTHER HOURS/WEEK na1. CATALOG DESCRIPTION

A study of the 8-bit microprocessor. The CPU instruction set, basic system hardware, chip select systems, memory, and direct I/O are the topics covered. Assembly language programming and software control of hardware are stressed.

2. CLASS SCHEDULE DESCRIPTION (125 characters)

8-bit CPU architecture, assembly language programming, hardware design, memory, and basic I/O. Extra supplies required. Prereq: ELEC 260 or equivalent with grade C or better.

3. PREREQUISITES AND COMPETENCY LEVELS REQUIRED (Title 5 Plan, p. 19)

ELEC 260 or equivalent with grade C or better.

4. COURSE OBJECTIVES (Must include objective(s) which involve "critical thinking," Title 5 Plan, p. 20.)

See attachment entitled ET-310 Course Objectives

5. RECOMMENDED TEXT MATERIALS

Brixen, ET-310 Lab Manual, CSM
Goody, The Intelligent Microcomputer, SRA
INTEL, MCS80/85 Family User's Manual, 1987

6. SUPPLIES NEEDED

ET310 parts kit available from Jameco Electronics in Belmont.

7. SCOPE OF THE COURSE (Attach topical outline)

See attachment entitled ET-310 Course Outline

8. EVALUATION (Include examinations, skills demonstrations, writing assignments, and other measures of ability to apply critical analysis. Include standards of evaluation where appropriate. Examples to be on file in Division Office.)

Eight reading/homework assignments, 14 lab activities, four exams, one midterm exam and one final exam.

PREPARED BY:



COURSE OBJECTIVES
ELEC-310 INTRODUCTION TO MICROPROCESSORS (11/87)

Upon completion of this course, the student will be able to:

1. identify the purpose and function of the main hardware elements of a single board microcomputer system including the central processor unit, the clock generator, the bus controller, tri-state data bus drivers, I/O ports, read/write memory, read only memory, address bus decoders, address bus buffers, display devices, and the keypad.
2. explain the effects of executing the major 8080/8085 instructions in the move, arithmetic, logic, branching, and machine control instruction groups.
3. indicate the correct CPU status word flag settings following the execution of an arithmetic or logic instruction.
4. write, from a flowchart, a simple assembly language program in standard form.
5. accurately translate the mnemonics and operands of an 8080/8085 source code program into 8080/8085 hexadecimal object code.
6. correctly use symbolic addresses, symbolic variables, and labels following Intel assembler requirements.
7. calculate time delays for single and nested delay loops.
8. identify the purpose and function of the principle hardware pins of the 8080A CPU, the 8224 clock generator chip, and the 8228 system bus controller chip.
9. identify, given proper Intel documentation, the binary number present on the address bus and the data bus, as well as which control bus signal is true, for each machine cycle of any given CPU instruction.
10. accurately interpret a CPU system timing diagram for read and write machine cycles.
11. analyze a program and the hardware it runs on to determine the correct sequence of chip select signals.
12. analyze a program and the hardware it runs on to determine if the microcomputer system uses accumulator I/O or memory mapped I/O.

COURSE OUTLINE
INTRODUCTION TO MICROPROCESSORS (11/87)

- I. Basic CPU architecture and operation
 - A. Parts of a microcomputer
 1. Central processing unit
 2. Memory
 3. Input and output ports
 - B. Register based CPU architecture
 1. General purpose registers
 2. Instruction register and instruction decoder
 3. Arithmetic logic unit and the processor status word (flags)
 4. Timing and control logic
 5. Four bus system
 - a. Address bus
 - b. Control bus
 - c. Data bus
 - d. Power bus
 - C. Basic system timing
 1. Instruction cycle
 2. Machine or bus cycle
 3. t-state
 4. Basic read and write timing
 - D. Computer Languages
 1. High level languages
 2. Assembly language--source code
 - a. Mnemonics
 - b. Operands
 - c. Labels
 - d. Comments
 3. Machine language--object code
 - a. Binary format
 - b. Hexadecimal format
- II. 8080/8085 Assembly Language Programming
 - A. Move instruction group
 1. General format of instructions
 2. Deriving binary and hexadecimal machine codes for a given move instruction
 3. Addressing modes
 - a. Immediate
 - b. Direct
 - c. Register
 - d. Register indirect
 4. Basic instruction timing
 - a. Fetch cycle
 - b. Memory read cycle
 - c. Memory write cycle
 5. Basic assembly language programming techniques
 - a. Flowcharting

- b. Program coding
 - 1) Address notation
 - 2) Machine code
 - 3) Labels
 - 4) Mnemonics
 - 5) Operands
 - 6) Comments
 - c. Use of symbols and labels
 - 6. Effect of the move instructions on the CPU flag register
 - B. Arithmetic instruction group
 - 1. General format of instructions
 - 2. Addition and 2's complement subtraction
 - 3. Effect of arithmetic instructions on the CPU flag register
 - 4. Double precision addition and subtraction
 - 5. Basic multiplication and division
 - 6. Basic instruction timing
 - a. Fetch cycle
 - b. Addition cycle
 - c. Subtraction cycle
 - d. Overlapping of arithmetic operations
 - C. Logic instruction group
 - 1. General format of instructions
 - 2. Masking and bit-setting techniques
 - 3. Software simulation of multi-input logic gates
 - 4. Effect of logic instructions on the CPU flag register
 - 5. Basic instruction timing
 - a. Fetch cycle
 - b. AND cycle
 - c. OR cycle
 - d. EXOR cycle
 - e. COMPLEMENT cycle
 - f. Rotate cycle
 - g. Overlapping of logic operations
 - D. Branching instruction group
 - 1. General format of instructions
 - 2. CPU decision making techniques
 - 3. The unconditional jump
 - 4. The conditional jump
 - 5. Subroutines
 - a. The stack and the stack pointer
 - b. The CALL instruction
 - c. The RETURN instruction
 - d. The conditional call
 - e. The conditional return
 - 6. Time delay loops
 - a. single loops
 - b. nested loops
 - c. decrement and OR loops
 - 7. Effect of the branching instructions on the CPU flag register
 - 8. Basic instruction timing

- a. The fetch cycle
 - b. Stack write and read cycles
 - c. Unconditional and conditional jumps
 - d. Unconditional and conditional calls
 - e. Unconditional and conditional returns
- E. The machine control instruction group
- 1. General format of instructions
 - 2. The stack
 - a. The stack pointer register
 - b. Pushing data onto the stack
 - c. Popping data off the stack
 - d. Use of the stack
 - e. Using the stack for data transfer
 - 3. Direct or Accumulator I/O instructions
 - 4. Effects of the machine control instructions on the CPU flag register
 - 5. Basic instruction timing
 - a. The fetch cycle
 - b. The stack read and write cycle
 - c. The input cycle
 - d. The output cycle

III. System Hardware

- A. 8224 clock generator chip
 - 1. Crystal timebase oscillators
 - 2. Phase 1 and phase 2 clock
 - 3. Reset input
 - 4. Ready input
 - 5. Sync input and status strobe output
- B. 8228/8238 bus controller chip
 - 1. Status word latching and decoding
 - 2. Bidirectional bus driver operation
- C. 8080A CPU
 - 1. Address generation
 - 2. Data bus multiplexing
 - 3. Status word generation
 - 4. Instruction fetch and decoding
 - 5. Data read/write
 - 6. General purpose registers
 - 7. CPU operation during a wait state
 - 8. CPU operation during a hold state
 - 9. CPU operation during a halt state
- D. Detailed read and write timing charts
- E. Address bus buffering and decoding
 - 1. Basic memory organization techniques
 - 2. Bus buffering
 - 3. Basic memory devices
 - a. Static RAM
 - b. Dynamic RAM
 - c. ROM
 - d. EPROM
 - e. EEPROM
 - f. EAPROM
 - 4. Data bus connections

5. Generation of chip select signals
6. HOLD and READY operations
7. Basic hardware and software techniques for accumulator I/O
8. Basic hardware and software techniques for memory mapped I/O
9. Design of a basic system