



Institiúid Teicneolaíochta Chorcaí
Cork Institute of Technology

ELTR6016: Microprocessor Systems 1

Module Details

| | | |
|-----------------------|--------------------------|------------------------|
| Short Title: | Microprocessor Systems 1 | APPROVED |
| Full Title: | Microprocessor Systems 1 | |
| Module Id: | 4831 | |
| Official Code: | ELTR6016 | NFQ Level: 6 |
| | | ECTS Credits: 5 |

| | |
|---------------------|----------------|
| Coordinator: | JOSEPH CONNELL |
|---------------------|----------------|

| | |
|---------------------|--|
| Description: | This module introduces the learner to the basic building blocks of modern microprocessor systems with the focus on these as controlling devices incorporated (embedded) into various electronic systems. Assuming a knowledge of basic digital systems, the module begins with the overall structure of simple microcomputers, introduces the main hardware building blocks and highlights the role of the microprocessor (mpu) as the system's central controlling device. The learner will also study the action of the microcomputer as it executes programs in its own native language, (machine programs), and will learn how to write such programs in assembly language using a microprocessor development system |
|---------------------|--|

Learning Outcomes:

On successful completion of this module the learner will be able to...

1. draw a block diagram of a typical microcomputer, identify each block and bus and explain the function and operation of the overall system
2. draw a block diagram of a microprocessor, identify each block and register set and explain their operation in terms of fetch-execute cycles for typical machine instructions
3. list the main microprocessor instruction types and addressing modes, and use these in the writing of simple assembly language program segments and routines
4. perform calculations on signed and unsigned numbers in binary and hexadecimal formats and become proficient in the use of mpu status flags to process such numbers
5. work alone and in teams to analyse and test various microcomputer hardware circuits/modules and to write, test, debug and document simple assembly language routines

Pre-requisite learning

Module Recommendations

This is prior learning (or a practical skill) that is strongly recommended before enrolment in this module. You may enrol in this module if you have not acquired the recommended learning but you will have considerable difficulty in passing (i.e. achieving the learning outcomes of) the module. While the prior learning is expressed as named CIT module(s) it also allows for learning (in another module or modules) which is equivalent to the learning specified in the named module(s).

Digital Systems I and Digital Systems II or equivalent

Incompatible Modules

These are modules which have learning outcomes that are too similar to the learning outcomes of this module. You may not earn additional credit for the same learning and therefore you may not enrol in this module if you have successfully completed any modules in the incompatible list.

No incompatible modules listed

Module Requirements

This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed. You may not enrol on this module if you have not acquired the learning specified in this section.

No requirements listed



Module Content & Assessment

Indicative Content

- **Basic microcomputer architecture**

Principle component parts of a computer, CPU, memory, RAM, ROM, I/O unit, peripherals. Von Neumann architecture. Data, address and control busses. Implications of data and address bus size. Programs, machine code and its relation to high-level languages. Fetch-execute cycles. Microprograms.

- **The microprocessor unit**

Microprocessor block diagram - control unit, ALU, general-purpose and special-purpose registers, bus drivers. Data structures, arrangement of data in registers, memory addressing capabilities. Structure of 16-bit memory - storage of bytes, words and longwords. MPU as a cpu - CISC and RISC. The machine cycle - fetch and execute cycles for specific machine instructions

- **Assembly language programming**

Instruction types - data transfer, arithmetic, logic, transfer of control. Addressing modes, register, memory direct, immediate, memory indirect modes. The status register and its uses. Jumps and branches, control of loops. The flowchart. Resident and cross assemblers. Comprehensive programming examples

- **Number systems**

Review of signed and unsigned binary number systems, sign extension, range of possible values, BCD system, hexadecimal notation, use of status flags in number processing

- **Laboratory assignments**

Identification and evaluation of microcomputer hardware devices/modules of a PC and 68000-based system, flowcharting, writing, assembling, testing, debugging and documenting assembly language routines, familiarisation with resident and cross assemblers.

| Assessment Breakdown | % |
|------------------------------------|-----|
| Course Work | 40% |
| End of Semester Formal Examination | 60% |

| | Outcome addressed | % of total | Assessment Date |
|---|-------------------|------------|-----------------|
| Formal End-of-Semester Examination | 1,2,3,4 | 60% | Semester End |

| Coursework Breakdown | | | | |
|-----------------------------|---|-------------------|------------|-----------------|
| Type | Description | Outcome addressed | % of total | Assessment Date |
| Practical/Skills Evaluation | A two-hour open-book exam on topics covered to date and also involving the writing, testing and documenting of a simple assembly language routine | 1,2,3,4,5 | 20 | Week 6 |
| Practical/Skills Evaluation | Weekly lab-based assignments on hardware and software | 5 | 20 | Every Week |

The institute reserves the right to alter the nature and timings of assessment



Module Workload & Resources

| Workload | | Full-time mode | | |
|---|---|----------------|------------|---------------------------------|
| Type | Description | Hours | Frequency | Average Weekly Learner Workload |
| Lecture | Two one-hour theory lectures | 2 | Every Week | 2.00 |
| Lab | A two-hour practical on various hardware and software assignments | 2 | Every Week | 2.00 |
| Independent & Directed Learning (Non-contact) | Review of lecture notes and recommended material and preparation of reports for selected laboratory sessions. | 3 | Every Week | 3.00 |
| Total Weekly Learner Workload | | | | 7.00 |
| Total Weekly Contact Hours | | | | 4.00 |

| Workload | | Part-time mode | | |
|---|--|----------------|-------------------|---------------------------------|
| Type | Description | Hours | Frequency | Average Weekly Learner Workload |
| Lecture | One two-hour lecture | 2 | Every Week | 2.00 |
| Lab | A two-hour practical on various hardware and software assignments | 2 | Every Second Week | 1.00 |
| Independent & Directed Learning (Non-contact) | Review of lecture notes and recommended material and preparation of reports for selected laboratory sessions | 4 | Every Week | 4.00 |
| Total Weekly Learner Workload | | | | 7.00 |

Resources

Recommended Book Resources

- Heffer, King, Keith, *Basic Principles & Practice of Microprocessors*, Arnold [ISBN: 0-7131-3569-7]
- Blighe, A B, *The Flight-68K - User's manual*, Flite Electronics Ltd Southampton

Supplementary Book Resources

- Brey, Barry B, *Microprocessors and Peripherals*, Merrill Publishing [ISBN: 0-675-20884-X]
- Kane, Hawkins, Leventhal, *68000 Assembly Language Programming*, McGraw-Hill [ISBN: 0-931988-62-4]

Other Resources

- Websites: Various websites for relevant data sheets and other material as recommended during the module