



# COMP226

## Computer Architecture

S2 Day 2013

*Computing*

### Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	2
<u>Assessment Tasks</u>	3
<u>Delivery and Resources</u>	5
<u>Unit Schedule</u>	6
<u>Learning and Teaching Activities</u>	7
<u>Policies and Procedures</u>	7
<u>Graduate Capabilities</u>	8
<u>Grade Requirements</u>	12
<u>More About This Unit</u>	13

#### **Disclaimer**

Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.

## General Information

Unit convenor and teaching staff

Unit Convenor

Len Hamey

[len.hamey@mq.edu.au](mailto:len.hamey@mq.edu.au)

Contact via [len.hamey@mq.edu.au](mailto:len.hamey@mq.edu.au)

Other Staff

Michael Johnson

[michael.johnson@mq.edu.au](mailto:michael.johnson@mq.edu.au)

Contact via [michael.johnson@mq.edu.au](mailto:michael.johnson@mq.edu.au)

Credit points

3

Prerequisites

COMP225(P) or COMP229(P) or COMP125(Cr)

Corequisites

Co-badged status

Unit description

This unit is about modern computer architecture. The goal is to understand the design of the machine rather than just know how to use it. We focus primarily on reduced instruction set computers (RISC) and include topical issues such as pipelining and memory hierarchies. Practical work uses assembly language programming and the C programming language.

## Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at <https://www.mq.edu.au/study/calendar-of-dates>

## Learning Outcomes

On successful completion of this unit, you will be able to:

Experience in assembly programming (SPARC) and understanding of the relationship between assembly code and machine code, including understanding of basic concepts of machine instruction sets.

Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.

Understanding of memory hierarchy and key design issues for cache and virtual memory systems.

Understanding of I/O mechanisms (for example bus operation and hard disk operation) and their access via system calls

Basic understanding of digital logic design - gates, half-adder, etc.

Understanding of machine level representations of recursion, stack discipline, etc. as part of a developing understanding of the relationships between high level languages and their realisation on low level architectures.

## Assessment Tasks

Name	Weighting	Due
<a href="#"><u>Assignment 0</u></a>	0%	Week 3
<a href="#"><u>Assignment 1</u></a>	15%	Week 5
<a href="#"><u>Assignment 2</u></a>	15%	Week 10
<a href="#"><u>Tutorial/Practical submissions</u></a>	10%	Each week
<a href="#"><u>Final Examination</u></a>	60%	TBA

### Assignment 0

Due: **Week 3**

Weighting: **0%**

This is a diagnostic assignment to help students check that they are learning the most important new material at the required rate. Your prac work from Practicals 2 and 3 must be submitted for marking in week 3.

Completing this assignment is a required part of the work for comp226.

On successful completion you will be able to:

- Experience in assembly programming (SPARC) and understanding of the relationship between assembly code and machine code, including understanding of basic concepts of machine instruction sets.
- Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.

### Assignment 1

Due: **Week 5**

Weighting: **15%**

Assignment 1: Assembly programming

On successful completion you will be able to:

- Experience in assembly programming (SPARC) and understanding of the relationship between assembly code and machine code, including understanding of basic concepts of machine instruction sets.

## Assignment 2

Due: **Week 10**

Weighting: **15%**

Assignment 2: Memory-efficient programming

On successful completion you will be able to:

- Understanding of memory hierarchy and key design issues for cache and virtual memory systems.

## Tutorial/Practical submissions

Due: **Each week**

Weighting: **10%**

From week 2, some tutorial questions will be identified as "Hand-in" questions. You should complete these questions **before** the tutorial and submit your solution no later than 5:50pm Monday in the tutorial submission box for COMP226 on the ground floor of building E6A. You should keep a copy of each tutorial submission for your own records, and bring it to class with you for discussion.

On successful completion you will be able to:

- Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.
- Understanding of I/O mechanisms (for example bus operation and hard disk operation) and their access via system calls

## Final Examination

Due: **TBA**

Weighting: **60%**

On successful completion you will be able to:

- Experience in assembly programming (SPARC) and understanding of the relationship

between assembly code and machine code, including understanding of basic concepts of machine instruction sets.

- Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.
- Understanding of memory hierarchy and key design issues for cache and virtual memory systems.
- Understanding of I/O mechanisms (for example bus operation and hard disk operation) and their access via system calls
- Basic understanding of digital logic design - gates, half-adder, etc.
- Understanding of machine level representations of recursion, stack discipline, etc. as part of a developing understanding of the relationships between high level languages and their realisation on low level architectures.

## Delivery and Resources

### Classes

Each week you should attend three hours of lectures, a two-hour practical class (which may include some tutorial work) and be prepared to work outside of class times too. For details of days, times and rooms consult the [timetables webpage](#).

(But, to be a little more helpful than that, comp226 lectures take place on Wednesdays from 6 to 9 in the evening, and comp226 prac and tutorial classes are available on Mondays and Fridays starting at 6pm (you choose one day) for two hours and are held in E6A 123 or 121.)

Note that there will be no tutorial or practical classes on the Monday of week 1, but there is tutorial work for everyone to do in week 1 which will be discussed in the first lecture, and there is a Friday evening practical class.

Please note that you are **required** to attend prac classes to have work marked, and you are required to submit work including tutorial work.

### Required and Recommended Texts and/or Materials

You are expected to use the following **text book**:

- P&H: D.A. Patterson and J.L. Hennessy, *Computer Organisation and Design, The Hardware Software Interface*, Revised Fourth Edition, Morgan Kaufmann, 2011.

The following books are **reference books** and may be useful in some parts of the course:

- H&P: J.L. Hennessy and D.A. Patterson, *Computer Architecture - A Quantitative Approach*. Morgan Kaufmann, Third Edition, 2003, or later.
- M&H: Miles J. Murdocca and Vincent P. Heuring, *Principles of Computer Architecture*. Prentice-Hall, 2000.

- B.W. Kernighan and D.M. Ritchie, *The C Programming Language*. Second Edition, Prentice-Hall, 1988, or later.

There is also a [companion website](#) for the text maintained by the publisher [books.elsevier.com](#). They now also offer a link to Google Books from [this page](#) for a substantial preview.

## Unit Webpage and Technology Used and/or Required

### Unit Web Page

The web page for this unit can be found at <http://www.comp.mq.edu.au/~mike/comp226>. Note that the majority of the unit materials will be openly accessible from that page.

### Echo360 discouraged

Lectures are an interactive experience - we expect you to attend and to be involved. We don't think of lectures as some encapsulated information transmission. They should be useful learning experiences, and we'll do our best to make them that for you. If matters beyond your control force you to miss a lecture please review the material available, read another student's notes, and then come and talk to us -- we do much better reviews of lecture content than echo360, and we like to help. So, while some lecture content may be recorded, we make no guarantees that Echo360 recordings will be provided for each lecture, nor that they will prove useful to you.

## Unit Schedule

The following is the plan for lectures. NOTE: The actual presentation arrangements may vary - please refer to the Lecture topic list on the unit website for up-to-date information.

1	Introduction; assembly programming	P&H ch 1,2	M&H ch 1,2,4	
2	SPARC instruction set; Machine code	P&H ch 2	M&H ch 4,5	Assignment 1 out
3	Machine code; Subroutines and stacks; Recursion	P&H chap 2	M&H ch 4, 5, 10.4	Assignment 0 due
4	Pipelining; Data hazards	P&H sec 4.5	H&P appendix A in 3rd ed.	
5	Memory and cache	P&H ch 5	M&H ch 7; H&P ch 5	Assignment 1 due Assignment 2 out

6	Cache; Efficient programming	H&P ch 5		
7	Virtual memory; TLB	P&H ch 7	H&P ch 5	
	<b>Study Break - 2 weeks</b>			
8	Virtual memory; Alpha AXP architecture	H&P ch 5; M&H 7.7		Assignment 2 due
9	IO; Hard disks; Buses; DMA; Interrupts	P&H ch 8		
10	Interrupts; Systems programming	<i>C Programming Language</i> , Kernighan & Ritchie <i>Structured Computer Organisation</i> , Tanenbaum 7.4 Unix man pages, esp man make, man -s2 read		
11	The datapath; Control	P&H ch 4; P&H appendix D	M&H ch 6; H&P appendix A	
12	Control; Digital logic	P&H appendix C	M&H appendix A	
13	Digital logic; Review			

## Learning and Teaching Activities

### Lectures

Weekly lectures provide discussion of key concepts.

### Workshop

Practical experience and tutorial discussions.

### Assignments

In-depth individual projects that are marked.

## Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central](#). Students should be aware of the following policies in particular with regard to Learning and Teaching:

Academic Honesty Policy [http://www.mq.edu.au/policy/docs/academic\\_honesty/policy.html](http://www.mq.edu.au/policy/docs/academic_honesty/policy.html)

Assessment Policy <http://www.mq.edu.au/policy/docs/assessment/policy.html>

Grading Policy <http://www.mq.edu.au/policy/docs/grading/policy.html>

Grade Appeal Policy <http://www.mq.edu.au/policy/docs/gradeappeal/policy.html>

Grievance Management Policy [http://mq.edu.au/policy/docs/grievance\\_management/policy.html](http://mq.edu.au/policy/docs/grievance_management/policy.html)

Special Consideration Policy [http://www.mq.edu.au/policy/docs/special\\_consideration/policy.html](http://www.mq.edu.au/policy/docs/special_consideration/policy.html)

In addition, a number of other policies can be found in the [Learning and Teaching Category](#) of Policy Central.

## Student Support

Macquarie University provides a range of Academic Student Support Services. Details of these services can be accessed at: <http://students.mq.edu.au/support/>

### UniWISE provides:

- Online learning resources and academic skills workshops [http://www.students.mq.edu.au/support/learning\\_skills/](http://www.students.mq.edu.au/support/learning_skills/)
- Personal assistance with your learning & study related questions.
- The Learning Help Desk is located in the Library foyer (level 2).
- Online and on-campus orientation events run by Mentors@Macquarie.

## Student Services and Support

Students with a disability are encouraged to contact the [Disability Service](#) who can provide appropriate help with any issues that arise during their studies.

## Student Enquiries

Details of these services can be accessed at <http://www.student.mq.edu.au/ses/>.

## IT Help

If you wish to receive IT help, we would be glad to assist you at <http://informatics.mq.edu.au/help/>.

When using the university's IT, you must adhere to the [Acceptable Use Policy](#). The policy applies to all who connect to the MQ network including students and it outlines what can be done.

## Graduate Capabilities

### Capable of Professional and Personal Judgement and Initiative

We want our graduates to have emotional intelligence and sound interpersonal skills and to demonstrate discernment and common sense in their professional and personal judgement. They will exercise initiative as needed. They will be capable of risk assessment, and be able to handle ambiguity and complexity, enabling them to be adaptable in diverse and changing environments.

This graduate capability is supported by:

### Assessment tasks

- Assignment 1



- Assignment 2
- Final Examination

## Commitment to Continuous Learning

Our graduates will have enquiring minds and a literate curiosity which will lead them to pursue knowledge for its own sake. They will continue to pursue learning in their careers and as they participate in the world. They will be capable of reflecting on their experiences and relationships with others and the environment, learning from them, and growing - personally, professionally and socially.

This graduate capability is supported by:

### Assessment tasks

- Assignment 1
- Assignment 2

## Discipline Specific Knowledge and Skills

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

### Learning outcomes

- Experience in assembly programming (SPARC) and understanding of the relationship between assembly code and machine code, including understanding of basic concepts of machine instruction sets.
- Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.
- Understanding of memory hierarchy and key design issues for cache and virtual memory systems.
- Understanding of I/O mechanisms (for example bus operation and hard disk operation) and their access via system calls
- Basic understanding of digital logic design - gates, half-adder, etc.
- Understanding of machine level representations of recursion, stack discipline, etc. as part of a developing understanding of the relationships between high level languages and their realisation on low level architectures.

## Assessment tasks

- Assignment 0
- Assignment 1
- Assignment 2
- Tutorial/Practical submissions
- Final Examination

## Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

## Learning outcomes

- Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.
- Understanding of memory hierarchy and key design issues for cache and virtual memory systems.
- Basic understanding of digital logic design - gates, half-adder, etc.
- Understanding of machine level representations of recursion, stack discipline, etc. as part of a developing understanding of the relationships between high level languages and their realisation on low level architectures.

## Assessment tasks

- Assignment 1
- Assignment 2
- Tutorial/Practical submissions
- Final Examination

## Problem Solving and Research Capability

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

## Learning outcomes

- Experience in assembly programming (SPARC) and understanding of the relationship between assembly code and machine code, including understanding of basic concepts of machine instruction sets.
- Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.
- Understanding of memory hierarchy and key design issues for cache and virtual memory systems.
- Understanding of I/O mechanisms (for example bus operation and hard disk operation) and their access via system calls
- Basic understanding of digital logic design - gates, half-adder, etc.
- Understanding of machine level representations of recursion, stack discipline, etc. as part of a developing understanding of the relationships between high level languages and their realisation on low level architectures.

## Assessment tasks

- Assignment 0
- Assignment 1
- Assignment 2
- Tutorial/Practical submissions
- Final Examination

## Creative and Innovative

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

## Learning outcomes

- Experience in assembly programming (SPARC) and understanding of the relationship between assembly code and machine code, including understanding of basic concepts of machine instruction sets.
- Understanding of RISC processor design - instruction pipeline, CPU components, HDL, etc.
- Understanding of memory hierarchy and key design issues for cache and virtual memory systems.
- Understanding of machine level representations of recursion, stack discipline, etc. as

part of a developing understanding of the relationships between high level languages and their realisation on low level architectures.

## Assessment tasks

- Assignment 0
- Assignment 1
- Assignment 2
- Tutorial/Practical submissions
- Final Examination

## Effective Communication

We want to develop in our students the ability to communicate and convey their views in forms effective with different audiences. We want our graduates to take with them the capability to read, listen, question, gather and evaluate information resources in a variety of formats, assess, write clearly, speak effectively, and to use visual communication and communication technologies as appropriate.

This graduate capability is supported by:

## Assessment tasks

- Assignment 1
- Assignment 2
- Tutorial/Practical submissions
- Final Examination

## Grade Requirements

Grading requirements apply to this unit.

To achieve a pass grade, you must demonstrate pass level achievement of the relevant learning outcomes in the final examination and you must demonstrate overall achievement suitable for a pass level grade.

If you do not submit a satisfactory attempt for all of the assignments, we can require you to complete additional work in order to pass the unit. Similarly, if you do not submit a reasonable attempt to at least ten tutorials and at least ten practicals, we can require you to complete suitable additional work in order to pass the unit. If you do not complete the required additional work, you will fail the unit.

All work submitted should be readable and well presented.

You should never commit plagiarism in any of your submitted work, including tutorial and practical answers.

From week 2, some tutorial questions will be identified as "Hand-in" questions. You should complete these questions **before** the tutorial and submit your solution no later than 5:50pm on

Mondays in the tutorial submission box for COMP226 on the ground floor of building E6A. Always keep a copy of your work that you can bring to the tutorial with you for the discussion. Tutorial 2 is due on the Monday of week 2, Tutorial 3 on the Monday of week 3, and so on.

## More About This Unit

This information does not fit neatly into the Units structure, so it appears here in a separate section. It is just as important as the rest of the unit outline.

## Why Study Computer Architecture?

In addition to the obvious fact that "computer architecture is fun and interesting" we'd like to point out that understanding computer architecture is helpful for efficient programming (which is discussed in this unit), for Operating System design (see COMP342: Operating Systems) and for compiler design (see COMP332: Programming Languages). The unit treats assembly-language programming in detail.

## Learning Outcomes

Above, we have identified the specific learning outcomes of COMP226, but we really want you to understand the key learning objective of this unit which is:

- To understand the computer (not just to know how to make it do things).

Of course, that involves the specific outcomes listed above, but it is a more fundamental concept. We want you to be able to work with computers, knowing how they are designed, how they work, and how that impacts what they can do.

## Broader Understandings

In addition to the specific learning outcomes listed above, this unit will help you develop a broader understanding of computing including the following:

1. Developing a detailed understanding of many of the driving forces in the profession of computing, and how they have led to the hardware we use.
2. A deeper insight into professional practice. As professionals, computing graduates are expected to have much deeper knowledge than merely how to make things work. Professionals need to *understand* the subject in a broad sense. This understanding equips graduates to be able to make decisions many years hence, long after current technology has been superseded.
3. A view of some of the social issues related to computing, especially those related to hardware and operating systems.
4. A generic understanding of assembly language and machine code. We will study SPARC, but the concepts you learn will be applicable to many other extant and future architectures.

5. And a very important technical generic skill: You will learn precisely how what you write in a program is executed by a machine, and hence how apparently reasonable high level language (HLL) code can be highly inefficient or can introduce intermittent and very hard to correct bugs.

## Graduate Capabilities - More Details

In addition to the discipline-based learning objectives, all academic programs at Macquarie seek to develop students' Graduate Capabilities in a range of areas. As listed above, one of the aims of this unit is that students develop their

- Critical analysis skills;
- Problem-solving skills;
- Creativity and innovation;

But you will develop other useful skills also. For example, we expect you to be proficient note-takers in lectures -- that is a very important skill for your future because you will frequently need to take concise but precise notes in meetings, discussions, or even just of your own plans and ideas. In comp226 we will also try to get you a little used to dealing with ambiguity. In your future life, specs will rarely be as carefully worked out as they are for first year assignments. Now we expect you to apply intelligence in interpreting specs. Similarly we provide a wide range of materials that are relevant to computer architecture study, and we expect you to take control of your learning and take advantage of those that are appropriate for your own state of knowledge.

In addition, as with most computing units you will develop your project management skills, communication skills, and quality assurance skills in completing assignment work, and you will gain insight into yourself, and how you work most effectively.

## Assessment Tasks

Deadlines are firm in this unit and you should ensure that you can submit your assignment work (even if it is incomplete) on time. Late work will **not be accepted** unless due to illness or misadventure. If you cannot submit on time because of illness or other circumstances, please contact the lecturer **before** the due date.

If serious illness or misadventure prevents you completing your work on time, you should contact us as soon as possible. We may make special arrangements for you to meet the requirements of the work, or we may set you alternative work or we may set you an extra exam. In any case, you should make every effort to submit, before the due date, what you have achieved and you should contact the lecturer **before** the due date.

You are encouraged to:

- set your personal deadline earlier than the actual one;
- keep backups of all your important files;
- make sure that no-one else picks up your printouts.

Late work will not be accepted. If you cannot submit on time because of illness or other

circumstances, please contact the lecturer **before** the due date.