MODULE DESCRIPTOR

TITLE	Fundamentals of Computer Architecture			
SI MODULE CODE	12-4473			
CREDITS	20			
LEVEL	4			
JACS CODE	*G411*			
SUBJECT GROUP	*SEGM*			
DEPARTMENT	*Computing*			
MODULE LEADER	*Dr Adrian Oram*			
NOTIONAL STUDY HOURS BY TYPE	Tutor-led	Tutor-directed	Self-directed	Total Hours
	48	*48*	*104*	*200*

MODULE AIM(S)

This module introduces the fundamental principles and features that underpin all modern computing systems design, and describes how the execution of software is influenced by the processor architecture being used. The aim is to inculcate in the student a clear understanding of the relationship between software and the hardware on which it executes.

MODULE LEARNING OUTCOMES

By engaging successfully with this module a student will be able to

1. List the common elements of typical computer hardware and explain the rationale for their design and inclusion, using the appropriate technical language.
2. Describe and manipulate simple logic circuits using Boolean algebra.
3. Describe how numbers/data are represented and manipulated in a typical processor.
4. Explain how code is executed, both in a general way and specifically on contemporary processors.
5. Describe how some common programming language structures are represented in the machine code of the target processor and be able to implement them in a low-level assembly language.
6. Use debugging tools to inspect and repair small programs.

INDICATIVE CONTENT

- **Basics of computer systems**: • Review of hexadecimal and binary number systems • Basic logic gates • Elementary Boolean algebra • Simple combinatorial and sequential logic circuits • Applications of logic circuits. • Floating point and other number systems.
- **Generalised computer systems architecture**: • Overview of typical computer systems • The Central Processing Unit (CPU) • Machine code and programming in a low-level assembler language • The fetch-execute cycle; how code is executed.
- **Contemporary computer systems**: • Improving CPU performance • Modern processor designs • General Purpose Graphical Processing Units (GPGPUs) • Device controllers • Memory systems, cache memory and buses • Input & output methods and devices • Device controllers • The role of the OS.

LEARNING AND TEACHING METHODS

Students will be supported in their learning, to achieve the above outcomes, in the following ways:

The module will be delivered in a mix of lecture and tutor-led practical sessions. The practical and lecture material is generally interdependent and mutually reinforcing. Practical sessions consist of hands-on circuit building, assembler programming, paper based exercises, assignments, and will include informal discussions, mini-briefings by the tutor, question and answer sessions, and the presentation of solutions to the problems posed.

ASSESSMENT STRATEGY AND METHODS

Task No.	TASK DESCRIPTION	SI Code	Task Weighting %	Word Count / Duration	In-module retrieval available
1	Coursework	CW	65%	N/A	Y
2	Exam	EX	35%		N

ASSESSMENT CRITERIA

An overall pass mark of 40% must be achieved to pass.

Students must demonstrate an understanding of number systems, some skill in the use of programming tools to develop and debug low-level assembler code, and be able to describe how the memory hierarchy, especially cache memory, influences the performance of code. They will also need to be able to manipulate Boolean expressions using algebra and use the correct nomenclature when describing modern computer systems.

FEEDBACK

Students will receive feedback on their performance in the following ways

- Individual feedback will be via written marks and comments, and group de-briefing sessions will ensure that both specific and generalised feedback is available.
- Phase test 1 will be worked in class after the event as feedback to all students.
- Assignment feedback is given both individually and collectively, so that lessons learnt are shared. Sufficient time is allocated to assessed work for feed-forward and advice to be given when required or asked for.
- At all times verbal feedback from tutors can be sought whenever appropriate during practical sessions.

LEARNING RESOURCES (INCLUDING READING LISTS)

PC labs with Visual Studio development environment.

Intel hardware manuals (available online and on the module Blackboard site).

- HENNESSY, J., L., PATTERSON, D., A.. (2003), Computer Architecture: A Quantitative Approach, 3rd ed., Morgan Kaufmann

- WILLIAMS, R., (2006), Computer Systems Architecture - A Networking Approach, 2nd ed. Prentice Hall

REVISIONS

Date	Reason
July 2012	Assessment Framework review

SECTION 2 'MODEL A' MODULE (INFORMATION FOR STAFF ONLY)
MODULE DELIVERY AND ASSESSMENT MANAGEMENT INFORMATION

MODULE STATUS - INDICATE IF ANY CHANGES BEING MADE

NEW MODULE	N
EXISTING MODULE - NO CHANGE	Y
Title Change	N
Level Change	N
Credit Change	N
Assessment Pattern Change	Y
Change to Delivery Pattern	N
Date the changes (or new module) will be implemented	**09/2011**

MODULE DELIVERY PATTERN - *Give details of the module delivery pattern. If the course has more than one intake, for example, September and January, please give details of the module start and end dates for each intake.*

	Module Begins	Module Ends
Course Intake 1	DD/MM/YYYY	DD/MM/YYYY
Course Intake 2	DD/MM/YYYY	DD/MM/YYYY
Course Intake 3	DD/MM/YYYY	DD/MM/YYYY

Is timetabled contact time required for this module?	Y

Are any staff teaching on this module non-SHU employees?	N
If yes, please give details of the employer institution(s) below	
What proportion of the module is taught by these non-SHU staff, expressed as a percentage?	

MODULE ASSESSMENT INFORMATION

Does the Module (using Model A Assessment Pattern) Require Either (why cap	
Overall Percentage Mark of 40%	Y
Overall Pass / Fail Grade	N

NB: Choose one of the above – Model A module cannot include both percentage mark and pass/fail graded tasks

FINAL TASK

According to the Assessment Strategy shown in the Module Descriptor, which task will be the LAST TASK to be taken or handed-in? (Give task number as shown in the Assessment Strategy)	Task No. 3

MODULE REFERRAL STRATEGY

Task for Task (as shown for initial assessment strategy)	Y
Single Referral Package for All Referred Students	N