Xi'an Jiaotong-Liverpool University
西交利物浦大学

Department of Computing

MODULE HANDBOOK

CPT101

Computer Systems

Steven Guan / Kok Hoe Wong

Semester 1

2021/2022

SECTION A: Basic Information

- **Brief Introduction to the Module**

This module introduces students to the various architecture levels of computer systems, including the digital logic level, the microarchitecture level, the instruction set architecture level, and the assembly language level.
It also introduces students to the components of computer systems and
a common CPU architecture (e.g. AMD, ARM, or Intel) so that students can relate theoretical concepts with their reali
in state-of-the-art processors
Students will be encouraged to reflect upon the purpose of knowledge creation including their own motivations for engaging with computer systems related research.

- **Key Module Information**

Module name:
Computer Systems

Module code: *CPT101*

Credit value: *5*

Semester in which the module is taught: *S1*

Pre-requisites needed for the module: *None*

Programmes on which the module is shared: *BEng Digital Media Technology*, *BEng Computer Science and Technology, BSc Information and Computing Science*

- **Delivery Schedule**

Seminar room: *TBA*

Seminar time: *TBA*

- **Module Leader and Contact Details**

Name: *Prof. Steven Guan /Dr Kok Hoe Wong*

Brief Biography of Steven Guan: *Steven Guan received his M.Sc. & Ph.D. from the University of North Carolina at Chapel Hill. He is currently a Professor and the Director for Research Institute of Big*

Data Analytics at Xi'an Jiaotong-Liverpool University (XJTLU). He served the head of department position at XJTLU for 4.5 years. Before joining XJTLU, he was a tenured professor and chair in intelligent systems at Brunel University, UK. Prof. Guan has worked in a prestigious R&D organization for several years, serving as a design engineer, project leader, and department manager. After leaving the industry, he joined Yuan-Ze University in Taiwan for three and half years. He served as deputy director for the Computing Centre and the chairman for the Department of Information & Communication Technology. Later he joined the Electrical & Computer Engineering Department at National University of Singapore as an associate professor.

Prof. Guan's research interests include: machine learning, modelling, security, networking, and pseudorandom number generation. He has published extensively in these areas, with 130+ journal papers and 180+ book chapters or conference papers. He has chaired and delivered keynote speech for 50+ international conferences and served in 180+ international conference committees and 20+ editorial boards.

Brief Biography of Kok Hoe Wong: *Upon completion of his Ph.D. in 3-D imaging, Dr. Kok Hoe WONG has worked for several renowned Multi-National Corporations (MNCs) before embarking into academia the last 10 years. His forte is in software engineering, project management, teaching and academic management. He has extensive experiences in architecting and managing enterprise-level IT projects, working with stakeholders from different parts of the world. Upon his arrival in China in 2007, he has progressed from being a Senior Lecturer to Vice President at a local institution, overseeing a successful Sino-Foreign partnership with Staffordshire University, UK. In recognition of his contributions, he had received several awards from the SuZhou government that include the "SIP Education Talent Award".*

Email address: *steven.guan@xjtlu.edu.cn* / *kh.wong@xjtlu.edu.cn*

Office telephone number: *0512-8816-1501/ 8188-4951*

Room number and office hours: *SD425 / SD431, by appointment*

Preferred means of contact: *e-mail*

SECTION B: What you can expect from the module

❑ Educational Aims of the Module

To introduce students to the components of computer systems.
To introduce students to the various architecture levels of computer systems, including the digital logic level, the microarchitecture level, the instruction set architecture level, and the assembly language level.
To introduce a common CPU architecture (e.g. AMD, ARM, or Intel) so that students can relate theoretical concepts with their realisation in state-of-the-art processors.

❑ Learning Outcomes

	LEARNING OUTCOMES
1	To understand the components of computer systems, their function, and interaction;
2	To understand the various architecture levels of computer levels of computer systems, their function, and interaction.

❑ Assessment Details

Initial Assessment

Sequence	**Method**	**Assessment Type (EXAM or CW)**	**Learning Outcomes Assessed** *(use codes under Learning Outcomes)*	**Duration**	**Week**	**% of Final Mark**	**Resit (Y/N/S)**
1	*Final Exam*	*EXAM*	*ALL*	*2 hours*		*80*	*S*
2	*Assessment Task 1*	*CW*	*ALL*			*10*	*S*
3	*Assessment Task 2*	*CW*	*ALL*			*10*	*S*

Resit Assessment

Assessment Type (EXAM or CW)	**Learning Outcomes Assessed** *(use codes under Learning Outcomes)*	**Duration**	**Week**	**% of Final Mark**
EXAM	*ALL*		*2 hours*	*100*

ASSESSED COURSEWORKS (1&2): (10% of the module mark each).

The task: *Using the techniques learnt from the Computer Systems module, design relevant inline assembly programs that include the methods prescribed in the assignment sheet. Specifically, develop an algorithm for each method. Finally, write in C and mainly in inline assembly language a program that implements all of these methods enabling specified input that accepts user input to test these methods with expected output.*

Deadline: *The deadline for coursework 1 is generally set during Week 7， 8 or 9 while for coursework 2 it is generally Week 12 or 13. Each student needs to attend an on-spot assessment session on the due date.*

Final Exam: (80% of module mark).

There will be Multi-Choice Questions, each worth 2.5 marks. These questions aim to examine students' knowledge on computer systems. The remaining exam questions will examine in further depth students' ability to apply knowledge learnt in the study of this module, make appropriate choices and design in this area to solve related problems in computer systems.

- **Methods of Learning and Teaching**

Students will be expected to attend three hours of formal lectures in a typical week as well as to participate in pre-scheduled, supervised help sessions in a computer lab. Lectures will introduce students to the academic content and practical skills which are the subject of the module, while help sessions will allow students to interact with teaching assistants, develop and practice those skills relevant to coursework solving. In addition, students will be expected to devote three hours of unsupervised time to solving continuous assessment tasks and private study. Continuous assessment will be used to test to what extent practical skills have been learnt, in particular, assessment tasks will be solved individually and each solution comprises the resolution, using inline assembly and sound problem solving techniques, of the given problems expressed in terms of a requirements statement.

- **Syllabus & Teaching Plan**

Week number and/or date	***Lecture/Seminar/ Field trip/other***	***Topic/Theme/Title***	***Pre-reading (ch. refer to Required Textbook)***
Week 1	*Lectures*	Overview and history of computer architecture; Design Principles for Modern Computers; Fundamental Buildings Blocks; *Principal components of a computer*	*Ch. 1, 2.1-2.3*
Week 2-3	*Lectures*	Processors (CPU Organization, instruction execution, RISC versus CISC, Harvard Architecture); *Coprocessors* Example Processors (Pentium); Motherboards; Chipsets and Controllers; BIOS and Boot Processor ; Data Codes, ASCII, Unicode;Operating Systems;Onion Ring Model; Computer Networks; Client-Server Computing	*Ch. 2.4-2.8, 5.3, 2.9-2.10*
Week 4	*Lectures*	C Programming Primer; Processor and Registers; Buses;Registers; Machine Cycle; Output Hardware; Execution of instructions; Assembly language; Main memory, RAM; Words, bytes and bits; Base register; Instruction pointer	*Ch. 3.1-3.2, 3.6, 7.1-7.2*
Week 5	*Lectures*	*CPU Registers; CPU status flags Inline assembler; Stack; PUSH and POP; Adjusting stack pointer; Stack as the temporary store; Passing parameters; Structure of instructions; Addressing modes; Register Indirect mode; Indexed Register*	*Ch. 3.6, 7.3-7.6, 8.2, 8.4, 8.5*

		Indirect with Displacement mode	
Week 6	*Lectures*	*Output in inline assembly; Input in inline assembly; More about printf; More about scanf; Controlling program flow; Jumps; Unconditional jumps; Conditional jumps; Controlling program flow: Loops; Loops with additional tests; Implementing higher-order constructs: conditional statements; Implementing higher-order constructs: the for statement* *Implementing higher-order constructs: the while statement* *Higher-order constructs: do-while statement* *switch-case statement*	*Ch. 8.4-8.5,*
Week 7	*Lectures*	*Subroutines; Return addresses; Subroutines in assembly language; Return from subroutine call; How does CALL work; Nested calls; CALL and RET working together Using the stack; Value parameters; Reference parameters; Passing parameters via registers; Local variables; Passing parameters via stack; Nested subroutine calls*	*Ch. 8.1-8.3, 8.5-8.6*
Week 8	*Lectures*	*Stack frame; Recursive subroutine* *Recursive method for factorial function in Java; Implementation of recursive function in the assembly language; Data codes –*	*Ch. 8.5-8.6*

		numeric and character; Number representations; Unsigned integers; Binary vs. BCD representation; Signed integers; Sign-and-magnitude representation; Complementary representations; 10's complementary coding	
Week 9	*Lectures*	*Stacks/queues and their implementation; LIFO, Creating a Stack using a Linked List with a header; FIFO, Creating a Queue using a Linked List with a header; Application of Queues, User job queue, Print spooling queue, I/O event queue*	*Ch. 3.6, 3.7, 6.2*
Week 10	*Lectures*	*Addition; Subtraction; Overflow testing Two's complement; Numerical types in Java; Algorithms; Floating Point Numbers; Floating Point Formats Excess-n notation; Normalisation of floating point numbers Floating point in binary; IEEE standard 754*	*Ch. 5.6, IEEE 754 standard*
Week 11	*Lectures*	*Data storage; Main memory Words, bytes and bits; RAM ROM; Memory hierarchy; Mass storage; Storage requirements for digital audio & video; Memory parameters ; Address width; Memory modules; Memory mapping Memory address decoding; Registers Cache memory; Memory hierarchy*	*Ch. 3.6, 12.7,12.8-12.10, 3.6, 6.7, 6.2 - 6.4, 12.1,12.3, 12.2*
Week 12	*Lectures*	*Hard disk drives; Tracks, sectors and cylinders; Disk*	*Ch. 12.5, 12.7, 4.1-4.2, 4.3-4.4, 4.6-4.7,*

		addressing *Hard disk vs. main memory* *Files, records, fields, keys; Virtual memory; Building computers from logic; Digital systems; Digital electronic circuits; Boolean operations and Boolean gates; Truth tables for basic logic operations; Boolean circuits; Selector circuits* *Selector circuit; Multiplexer* *Two-line decoder; Data selector with two-line decoder; Implementing a function with logic gates*	
Week 13	*Revision*	*Revision & info for the final exam*	

- **Reading Materials**

Required (Essential) Textbook:

Title	Author	ISBN/Publisher
COMPUTER SYSTEMS ARCHITECTURE A NETWORKING APPROACH	R. WILLIAMS	*ADDISON-WESLEY*

Additional Readings:

In addition to the required and recommended readings given above, you are encouraged to identify appropriate further resources for your study, e.g. articles in computer systems related academic journals

SECTION C: Further Information

- **Student Feedback**
  The University requires student feedback to be obtained and evaluated by Departments for each module in every session. It is University policy that the preferred way of achieving this is by means of an Online Module Evaluation Questionnaire Survey. Students will be invited to complete the questionnaire survey for this module at the end of the semester.

You are strongly suggested to read policies mentioned below very carefully, which will help you better perform in your academic studies.
All the policies and regulations related to your academic study can be found in Student Academic Services section under the heading "Policies and Regulations" on E-bridge.

- **Plagiarism, Cheating, and Fabrication of Data.**
  Offences of this type can result in attendance at a University-level committee and penalties being imposed. You need to be familiar with the rules. Please see the "Policy for Dealing with Plagiarism, Collusion and Data Fabrication" document available on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- **Rules of submission for assessed coursework**
  The School has detailed rules and procedures governing the submission of assessed coursework. You need to be familiar with them. Details can be found in the "Code of Practice for Assessment" document available on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- **Late Submission of Assessed Coursework**

  The University attaches penalties to the late submission of assessed coursework. You need to be familiar with the University's rules. Details can be found in the "Code of Practice for Assessment" document available on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- **Mitigating Circumstances**

  The University is able to take into account mitigating circumstances such as illness or personal circumstances which may have adversely affected student performance on a module. It is the student's responsibility to keep their Academic Adviser, Programme Director or Head of Department informed of illness and other factors affecting their progress during the year and especially during the examination period. Students who believe that their performance on a examination or assessed coursework may have been impaired by illness, or other exceptional circumstances should follow the procedures set out in the Mitigating Circumstances Policy, which can be found on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- **ICE**

  Copies of lecture notes and other materials are available electronically through ICE, the University's virtual learning environment.