MULTIMEDIA UNIVERSITY®

COURSE INFORMATION

1 .	**Name of Course**	Computer Architecture and Organization
2 .	**Course Code**	TSN1101
3 .	**Type of Course** **(e.g. : Core, major, elective etc.)**	Core
4 .	**Synopsis**	Computer architecture and organization addresses the fundamental principles in computer design, from the basic buidling blocks of number systems, data representation and digital systems to microprocessor design, memory and input/output strategies.
5 .	**Version** **(State the date of theSenate's approval - previous and the current approval date)**	Current: January 2018 Previous: June 2016
6 .	**Name(s) of Academic Staff**	Ramakrishnan Kannan Timothy Yap Tzen Vun Ng Hu Goh Hui Ngo Tan Saw Chin
7 .	**Semester and Year Offered**	Trimester 2 (Beta)
8 .	**Credit Value**	4 credit hours
9 .	**Pre-Requisite**	Nil

10 . **Objective of the course in the programme:**

- To provide the knowledge of how the data can be represented and how the different arithmetic operations can be done using different types of number systems and codes.
- To provide a good knowledge of the fundamentals for the logic design of combinational and sequential logic circuits.
- To provide discussions on the fundamentals of computer organization and architecture.
- To describe the internal structure and operation of a Central Processing Unit and Control Unit.
- To understand the improvement in processor performance by using pipelining and branch prediction techniques.
- To discuss instruction set architecture and design, different types of instructions, addressing modes and instruction formats.
- To assimilate the need for memory hierarchy and different characteristics of memory systems.
- To discuss the different ways of mapping cache memory and the replacement algorithms.
- To explain the different I/O data transfer schemes.
- To gain knowledge of how the performance can be improved by using multiprocessors, multicore and multithreaded processors.

11 . **Justification for including the course in the programme:**

Understanding of the fundamental principles behind the design and operation of computer system, performance characteristics, and interfacing techniques are essential for a student in the field of computer science because of the following reasons:

- To use software tools effectively in order to create fast and efficient software and also to identify errors.
- To write fast and efficient compilers
- To write device drivers for I/O devices
- To work on embedded systems
- To design a good operating system
- To model large, complex, real world systems
- To appreciate the complex trade-offs between CPU clock speed, cost, portability, power consumption, reliability, programmability, cache size, number of core processors, and so on.
- To understand the other topics of computer science such as operating systems, high-level language concepts such as pointers, parameter passing etc.

12 .

Course Learning Outcomes (CLO)		**Domain**	**Level**
CLO1:	Describe how the data can be represented in different ways and how the different arithmetic operations can be performed inside the computer.	Cognitive	2
CLO2:	Design combinational and sequential logic circuits using basic logic gates and construct them experimentally using logic function integrated circuits	Cognitive	6
CLO3:	Illustrate the structures and functions of the primary components of computer and the fundamental differences between the organization of multiprocessors, multicore, and multithreaded processors.	Cognitive	4
CLO4:	Write simple assembly language programs, by using instruction set of a typical microprocessor.	Cognitive	3

13 . **Mapping of the Course Learning Outcomes to the Programme Learning Outcomes, Teaching Methods and Assessment:**

Course Learning Outcomes (CLO) (Must tally with CLOs in item 12)	**Programme Learning Outcomes (PLO)**												**Teaching Methods**	**Assessment Method**
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12		
CLO1		✓		✓									Lecture/Practical	Test/Final Exam
CLO2		✓	✓	✓									Lecture/Practical	Lab Experiments/Test/Final Exam
CLO3		✓	✓	✓									Lecture/Practical	Test/Final Exam
CLO4	✓	✓	✓	✓									Lecture/Practical	Lab Experiments/Test/Final Exam
Total	**1**	**4**	**3**	**4**									*Indicate the relevancy between the CLO and PLO by ticking "✓" the appropriate relevant box (This description must be read together with standards 2.1.2, 2.2.1, and 2.2.2 in Area 2 – pages 16 & 18 of COPPA 2.0)*	

14 . **Transferable Skills:**

Critical thinking

15 . **Distribution of Student Learning Time (SLT)**

	Course Content Outline	****CLO**	**Teaching and Learning Activities**						**Total SLT**
			Guided Learning (F2F)*				**Guided Learning (NF2F)***	**Independent Learning (NF2F)***	
			***L**	***T**	***P**	***O**			
1	**Introduction to Digital Logic and Boolean algebra** Digital and analog systems, Logic gates, Boolean algebra, Simplification using boolean algebra, Boolean analysis of logic circuits		3		4			6	13
2	**Combinational Logic Circuits** Standard forms of boolean expressions, Simplification using Karnaugh map, Design of combinational logic circuits, Standard combinational logic circuits		2		4		4	5	15
3	**Sequential Logic Circuits** Latches and flip-flops, Asynchronous and synchronous counters, Basic functions and types of shift registers, Shift register counters		2		2		4	3	11

4	**Number Systems, Codes and Machine Level Representation of Data** Number system conversions and arithmetic, Numeric and alphanumeric codes, Integer representation and arithmetic, Floating-point representation and arithmetic.		2				4	2	8
5	**Computer Evolution, Performance, and System Buses** Basic concepts and computer evolution, Introduction to embedded systems and the Internet of Things (IoT), Performance issues, Computer components and functions, Instruction cycle and interrupts, Bus interconnection structure, Case Studies: Quick path interconnect and peripheral component interconnect approaches.		4					4	8
6	**Structure and Function of Central Processing Unit and Operation of Control Unit** Internal structure of CPU, Organization of registers, Instruction cycle with indirect stage, Instruction pipelining: strategy, performance, source, data and control hazards, Control unit operation Case Studies: register organization and interrupt processing of Intel x86 and ARM processors.		6		2			7	15
7	**Instruction Set Architecture and Design** Characteristics and functions of instruction sets, types of operands, types of operations, Addressing modes, Instruction formats, Case Studies: Intel x86 and ARM data and operation types, addressing modes, and instruction formats. Assembly language programming with Intel x86.		6		14			16	36
8	**Memory System Organization and Architecture** Characteristics of memory systems, Memory hierarchy, Cache memory: principles, mapping functions, replacement algorithms, write policy, and multilevel caches, Semiconductor main memory: types and characteristics		3					3	6
9	**Input / Output Interfacing Strategies** Structure and function of I/O modules, Data transfer techniques: programmed I/O, interrupt-driven I/O, direct memory access, Direct cache access strategies, I/O Channels and Processors, Case Studies: USB, Firewire, SCSI, Thunderbolt, InfiniBand.		3		2			5	10
10	**Multiprocessing, Alternative Architectures, and Performance Enhancements** Flynn's taxonomy: Multiple Processor Organizations, Cache Coherence problem, Multi-threading, Multi-Core Computing, General-Purpose Graphic Processing Units (GPUs), Instruction Level Parallelism and Superscalar processors		5					5	10
								Total SLT	**132**

SUMMATIVE ASSESSMENT

1. Continuous Assessment	**Percentage %**	**Total SLT**	
Tests	20%	5	
Lab Experiments	20%	9	
	Total SLT for Continuous Assessment	**14**	
2. Final Assessment	**Percentage %**	**Total SLT**	
		F2F	**ILT**
Final Exam	60%	2	12
	Total SLT for Final Assessment (F2F + NF2F)	**14**	
Grand Total	**100%**	**160**	

****Indicate the CLO based on the CLO's numbering in Item 12.**
***L= Lecture, *T= Tutorial, *P= Practical, *O= Others, F2F*= Face to Face, NF2F*= Non Face to Face**

16 . **Identify Special Requirement to Deliver the Course (e.g., software, nursery, computer lab, simulation room):**
Microprocessor assembly language simulator

17 . **Main References:**
William Stallings. (2016). Computer Organization and Architecture (10th ed.). Prentice Hall.
Thomas L. Floyd. (2015). Digital Fundamentals (11th ed.). Prentice Hall.

18 . **Additional References:**
M. Morris Mano, Charles R. Kime and Tom Martin (2016). Logic and Computer Design Fundamentals (5th ed.). Prentice Hall.
Patterson and Hennessy. (2013). Computer Organization and Design: The Hardware/Software Interface (5th ed.). Morgan Kaufmann.
M.Moris Mano and Michael D. Ciletti. (2013). Digital Design (5th ed.). Prentice Hall.
Barry B. Brey. (2009). Intel Microprocessors (8th ed.). Prentice Hall.
Kris Schindler. (2013). Introduction to Microprocessor based systems using the ARM processor (2nd ed.). Prentice Hall

Note:

▨	Cells shaded light grey contain formulas / fixed values. Edit these formulas only if needed.