

# MODULE DESCRIPTION FORM

## نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Computer Organization & Architecture		Module Delivery
Module Type	Core		<input checked="" type="checkbox"/> Theory <input checked="" type="checkbox"/> Lecture <input checked="" type="checkbox"/> Lab <input type="checkbox"/> Tutorial <input type="checkbox"/> Practical <input checked="" type="checkbox"/> Seminar
Module Code	MU010602104		
ECTS Credits	5		
SWL (hr/sem)	125		
Module Level	UGII	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader	Sundos Abdulameer Hameed	e-mail	<a href="mailto:ss.aa.cs@uomustansiriyah.edu.iq">ss.aa.cs@uomustansiriyah.edu.iq</a>
Module Leader's Acad. Title	Assistant Prof Doctor	Module Leader's Qualification	PHD.
Module Tutor	Muna Abdulhussain Radi	e-mail	<a href="mailto:muna.ali@uomustansiriyah.edu.iq">muna.ali@uomustansiriyah.edu.iq</a>
Peer Reviewer Name		e-mail	
Scientific Committee Approval Date	20/04/2024	Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Logic Design (1)	Semester	2
Co-requisites module	None	Semester	

## Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

<p><b>Module Objectives</b> أهداف المادة الدراسية</p>	<p>This course will describe the basics of</p> <ol style="list-style-type: none"> <li>1. Topics include computer system performance, instruction set architectures, pipelining, branch prediction, memory-hierarchy design, and a brief introduction to multiprocessor architecture issues. General Introduction of Emulator 8086 and assembly language in the lab.</li> <li>2. Assembly language programming, including logic, comparing and branching, interrupts, macros, procedures, arrays, program design, testing, debugging, loading, and linking; combinational, arithmetic and logical circuits ALU; memory circuits, latches, flip-flops, registers; computer structure; fetch-execute cycles, clocks and timing; microprogramming and microarchitecture; data path, timing, sequencing; cache memory organization; RISC architectures</li> </ol>
<p><b>Module Learning Outcomes</b> مخرجات التعلم للمادة الدراسية</p>	<ol style="list-style-type: none"> <li>1. Understand the key components within a computer system and organization.</li> <li>2. Understand number systems and implement binary arithmetic with overflow handling and floating-point numbers.</li> <li>3. understand digital logics, data path, and control using logic gates, combinational and sequential components</li> <li>4. Analyze the cause of pipeline hazards and implement ways to overcome them.</li> <li>5. Understand memory hierarchy and apply the principle of locality in cache design.</li> <li>6. Understanding the need for the use of memory hierarchy (cache, main memory, storage devices) to ensure the design of a balanced computer system.</li> <li>7. Understanding the fundamental principles of CPU and control unit design.</li> <li>8. Discuss the relationship between the design of a computer system and the design of an operating system to operate it.</li> <li>9. Evaluate various trade-offs in designing the instruction set architecture.</li> </ol>
<p><b>Indicative Contents</b> المحتويات الإرشادية</p>	<p>Indicative content includes the following.</p> <ol style="list-style-type: none"> <li>1. GENERAL INTRODUCTION &amp; REVIEW:             <ul style="list-style-type: none"> <li>○ Basic concepts revision</li> <li>○ computer Architectures</li> <li>○ computer design,</li> <li>○ History of computers</li> <li>○ computer Components</li> </ul> </li> <li>2. DATA REPRESENTATION             <ul style="list-style-type: none"> <li>○ Data Type Representation Signed number, fixed, floating point, character</li> <li>○ Addition Ripple Carry Adder, Carry Look-ahead,</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>○ Multiplication Shift and Add, Booth Multiplier, Carry Save Multiplier</li> </ul> <ol style="list-style-type: none"> <li>3. GENERAL PURPOSE REGISTERS <ul style="list-style-type: none"> <li>○ 8086 Emulator</li> <li>○ Combinational and sequential circuit</li> <li>○ Instruction set</li> </ul> </li> <li>4. MAIN MEMORY ORGANIZATION <ul style="list-style-type: none"> <li>○ Type of computer memory</li> <li>○ Memory expansions</li> <li>○ Addressing modes and effective address</li> <li>○ Hierarchical memory organization</li> <li>○ Direct Memory Access</li> </ul> </li> <li>5. CENTRAL PROCESSING <ul style="list-style-type: none"> <li>○ CPU Organization &amp; Operation</li> <li>○ Hardware Implementation</li> <li>○ Performance of Computer</li> <li>○ Basic Measurement Metrics</li> <li>○ Pipelining: instruction</li> </ul> </li> <li>6. PARALLEL PROCESSORS <ul style="list-style-type: none"> <li>○ Parallel Program</li> <li>○ SISD, MIMD, SIMD, SPMD, and Vector</li> <li>○ HW Multithreading</li> <li>○ Multicore and Shared Memory Multiprocessors</li> </ul> </li> </ol>
--	--

<b>Learning and Teaching Strategies</b> استراتيجيات التعلم والتعليم	
<b>Strategies</b>	<p>Learning Strategies:</p> <ol style="list-style-type: none"> <li>1. <b>Understand Fundamentals:</b> Start with the basics. Learn about binary arithmetic, logic gates, and basic digital circuit design. Understanding these fundamental concepts will provide a solid foundation</li> <li>2. <b>Follow a Structured Curriculum</b> Look for structured courses or textbooks on computer architecture. These resources typically start with the basics and gradually introduce more complex topics.</li> <li>3. <b>Hands-On Practice:</b> Theory is essential, but hands-on practice can deepen your understanding. Experiment with circuit simulators, assembly language programming, and hardware description languages.</li> <li>4. <b>Utilize Visual Aids:</b> Computer architecture often involves abstract concepts that can be challenging to visualize. Look for diagrams, animations, or interactive simulations that illustrate key concepts like CPU pipelines, memory hierarchies, and instruction execution.</li> </ol> <p>Teaching Strategies:</p>

	<ol style="list-style-type: none"> <li><b>Start with Basics:</b> Begin by laying a solid foundation with fundamental concepts such as binary arithmetic, logic gates, and Boolean algebra. Use real-world examples to illustrate abstract concepts and make them more relatable.</li> <li><b>Use Visual Aids:</b> Computer architecture often involves intricate diagrams and structures. Utilize visual aids such as slides, animations, and interactive simulations to help students visualize abstract concepts like CPU pipelines, memory hierarchies, and instruction execution.</li> <li><b>Encourage Active Learning:</b> Instead of simply lecturing, engage students in active learning activities such as group discussions, problem-solving exercises, and hands-on projects.</li> <li><b>Provide Real-World Examples:</b> Relate computer architecture concepts to real-world applications and industry practices. Show how concepts like caching, pipelining, and parallel processing are used in modern processors and systems to improve performance and efficiency.</li> <li><b>Demonstrate with Simulations:</b> Use software tools and simulators to demonstrate how different architectural designs and optimizations impact system performance. Interactive simulations allow students to explore complex concepts in a controlled environment and see cause-and-effect relationships in action.</li> <li><b>Integrate Practical Exercises:</b> Design hands-on exercises and lab assignments that reinforce theoretical concepts and allow students to apply their knowledge in a practical setting.</li> </ol>
--	--

Student Workload (SWL)			
الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
<b>Structured SWL (h/sem)</b> الحمل الدراسي المنتظم للطالب خلال الفصل	79	<b>Structured SWL (h/w)</b> الحمل الدراسي المنتظم للطالب أسبوعيا	5
<b>Unstructured SWL (h/sem)</b> الحمل الدراسي غير المنتظم للطالب خلال الفصل	46	<b>Unstructured SWL (h/w)</b> الحمل الدراسي غير المنتظم للطالب أسبوعيا	3
<b>Total SWL (h/sem)</b> الحمل الدراسي الكلي للطالب خلال الفصل	<b>125</b>		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
<b>Formative assessment</b>	<b>Quizzes</b>	2	10% (10)	5 and 7	LO #1, #2, #3,#4
	<b>Assignments</b>	2	8% (8)	11 and 15	LO #5,#6, #7,#8

	<b>Projects / Lab.</b>	1	14% (14)	Continuous	All
	<b>Report</b>	1	8% (8)		LO #8, #9
<b>Summative assessment</b>	<b>Midterm Exam</b>	2hr	10% (10)	8	LO #1, #2, #4, #5, #6, #7
	<b>Final Exam</b>	4hr	50% (50)	16	All
<b>Total assessment</b>			100% (100 Marks)		

<b>Delivery Plan (Weekly Syllabus)</b> المنهاج الاسبوعي النظري	
	<b>Material Covered</b>
<b>Week 1</b>	General Introduction & Review
<b>Week 2</b>	Data Representation
<b>Week 3</b>	Combinational and sequential circuit
<b>Week 4</b>	Main Memory Organization
<b>Week 5</b>	Hierarchical memory organization
<b>Week 6</b>	Memory expansions and Instruction formats
<b>Week 7</b>	Central Processing Organization
<b>Week 8</b>	<b>Midterm Exam</b>
<b>Week 9</b>	Execute Of Complete Instruction
<b>Week 10</b>	Performance of Computer
<b>Week 11</b>	Pipelining processing
<b>Week 12</b>	Direct Memory Access
<b>Week 13</b>	Input-Output Organization
<b>Week 14</b>	RISC and CISC pipeline, SISD, MIMD, SIMD, SPMD, and Vector
<b>Week 15</b>	Huffman code instruction
<b>Week 16</b>	<b>Preparatory week before the final Exam</b>

<b>Delivery Plan (Weekly Lab. Syllabus)</b> المنهاج الاسبوعي للمختبر	
	<b>Material Covered</b>
<b>Week 1</b>	General Introduction of Emulator 8086
<b>Week 2</b>	Numbering System
<b>Week 3</b>	assembly language (part1)
<b>Week 4</b>	assembly language (part2)

<b>Week 5</b>	assembly language (part3)
<b>Week 6</b>	8086 interaction set1
<b>Week 7</b>	8086 interaction set2
<b>Week 8</b>	Examples Methods
<b>Week 9</b>	Examples Methods
<b>Week 10</b>	Variables
<b>Week 11</b>	Inside the CPU
<b>Week 12</b>	Memory Access
<b>Week 13</b>	XCHG Instruction
<b>Week 14</b>	Arithmetic Instruction
<b>Week 15</b>	Register Flag

<b>Learning and Teaching Resources</b> مصادر التعلم والتدريس		
	Text	Available in the Library?
<b>Required Texts</b>	<ul style="list-style-type: none"> <li>- "Wang, Shuangbao Paul. Computer Architecture and Organization: Fundamentals and Architecture Security. Springer Nature, 2021.</li> <li>- Zomaya, Albert Y. "Advanced computer architecture and parallel processing." (2021)..</li> </ul>	Yes
<b>Recommended Texts</b>	<ul style="list-style-type: none"> <li>- Hayes, John P. <i>Computer architecture and organisation</i>. Tata McGraw Hill, 2022.</li> <li>- Nisan, Noam, and Shimon Schocken. <i>The elements of computing systems: building a modern computer from first principles</i>. MIT press, 2021.</li> <li>- Null, Linda. <i>Essentials of Computer Organization and Architecture</i>. Jones &amp; Bartlett Learning, 2023.</li> </ul>	yes
<b>Websites</b>		

<b>Grading Scheme</b> مخطط الدرجات				
Group	Grade	التقدير	Marks %	Definition
<b>Success Group (50 - 100)</b>	<b>A</b> - Excellent	امتياز	90 – 100	Outstanding Performance
	<b>B</b> - Very Good	جيد جدا	80 – 89	Above average with some errors
	<b>C</b> - Good	جيد	70 – 79	Sound work with notable errors
	<b>D</b> - Satisfactory	متوسط	60 – 69	Fair but with major shortcomings

	E - Sufficient	مقبول	50 – 59	Work meets minimum criteria
<b>Fail Group (0 – 49)</b>	<b>FX – Fail</b>	راسب (قييد المعالجة)	(45-49)	More work required but credit awarded
	<b>F – Fail</b>	راسب	(0-44)	Considerable amount of work required

**Note:** Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.