MODULE DESCRIPTION FORM

Module Information					
Module Title	**Microprocessor**			**Module Delivery**	
Module Type	**Core**			☒Lecture ☒Practical	
Module Code	**IT2104**				
ECTS Credits	**6**				
SWL (hr/sem)	**150**				
Module Level	UG2		**Semester of Delivery**	1	
Administering Department	Information Technology		**College**	College of Science	
Module Leader	Maky H.Abdulraheem		**e-mail**	maky.h@uowa.edu.iq	
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**	Ph.D	
Module Tutor	Ali Abdul Hussein Ibrahim		**e-mail**	ali.abdulhussein@uowa.edu.iq	
Peer Reviewer Name	Dr .Hayder Ghanim		**e-mail**	hayder.alghanami@uowa.edu.iq	
Scientific Committee Approval Date	2025-09-1		**Version Number**	V01	

Relation with other Modules			
Prerequisite module	None	Semester	None
Co-requisites module	None	Semester	None

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	The purpose of the course is to teach and understand the main components and working principles of the 8086 processor. Understanding of basic computer architecture. Understanding memory organization and interaction with memory . Handling I/O units. The course analyzes the several components of a computing system: from the microprocessor internal architecture, up to system bus for peripheral devices management. The course also covers programming at assembly level.
Module Learning Outcomes	The course has following specific objectives: 1) Teaching the student about the microprocessor and its components and how to implement the instructions. 2) Learn assembly language 3) Knowing the methods and stages of converting an assembly language program into symbols. 4) Teaching the student, the principle of memory system and how it was divided the data into segments and how to link them. 5) To explain the principle of data flow.
Indicative Contents	Indicative content includes the following. ✔ Identifying the parts and components of the processor, the most important main units in the processor, identifying the memory, the most important signals that deal with it, and methods of accessing data inside the memory and the processor. ✔ A compiler design that uses specific algorithms in which data is entered in the manner of rules and laws are applied to it to know the results and to know the errors resulting in implementation and classify them according to their type and treat them

Learning and Teaching Strategies	
Strategies	The learning and teaching strategies for studying the microprocessor subject in an IT department involve a balanced approach of theoretical understanding and practical application. Lectures, interactive discussions, provide the necessary theoretical foundation. Practical exercises, group work, enable hands-on experience with microprocessor 8086. Giving lectures, carrying out assignments and practical issues inside the laboratories, conducting theoretical exams, discussions and scientific dialogues, and asking questions. These strategies ensure a comprehensive understanding of microprocessor and their relevance in the IT field.

Student Workload (SWL)			
Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	87	**Unstructured SWL (h/w)**	**6**
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	7	10% (8)	2,4,6,8,10	1,2,3,4
	Homework assignment	3	10% (7)	2,5,8,9,12	All
	Onsite Assignments	5	10% (5)	3,5,8,10,11	All
	Projects	1	10%(5)	12	All
	Lab	5	10%(10)	3,5,7,9,12	All
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
	Material Covered
Week 1	Fundamental to microprocessor
Week 2	Microprocessor 8086 internal Architecture . Bus Interface Unit, Execution unit, register organization. Address bus, data bus, control bus
Week 3	
Week 4	Memory unit and R/W timing diagram memory segmentation Instruction cycle
Week 5	
Week 6	Memory addressing mode Instruction format
Week 7	
Week 8	Input / output devices and R/W timing diagram
Week 9	
Week 10	Assembly Language Programming Data transfer instruction set
Week 11	
Week 12	Variable , Array and constant
Week 13	Arithmetic and Logical instruction set
Week 14	
Week 15	Rotate and shift instruction set
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)	
	Material Covered
Week 1	Lab 1: Setting up the emu8086 simulation
Week 2	Lab 2: the concept of Assembly Language
Week 3	Lab 3: Practical basic on assembly language
Week 4	Lab 4: learn to build a code using emu8086 simulation
Week 5	Lab 5,6 :Learn to create code for data transfer instruction set
Week 6	
Week 7	Lab 7,8: Learn to convert from Assembly language to machine language
Week 8	
Week 9	Lab 9,10: Learn to create code for arithmetic and logical instruction set
Week 10	
Week 11	Lab 11,12: Learn to deal with variable and array in emu8086 simulation
Week 12	
Week 13	Lab 13,14 : Learn to create code for rotate and shift instruction set
Week 14	
Week 15	Lab 15: implemented a code for preparing to the final exam

Learning and Teaching Resources		
	Text	**Available in the Library?**
Required Texts	• The 80x86 Family, Design, Programming and Interfacing, 3rd edition,Prentice Hall, 2002.	
Recommended Texts	• The Intel Microprocessors, Architecture, Programming and Interfacing, Barry B. Brey, Prentice Hall, 1994.	

Grading Scheme				
Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	Fail	(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.