MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Computer Organization & Architecture / تركيب ومعمارية الحاسوب**			**Module Delivery**	
Module Type	**S**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	CYBT208				
ECTS Credits	**5**				
SWL (hr/sem)	**125**				
Module Level			**Semester of Delivery**	4	
Administering Department	Department of Cybersecurity Technical Engineering		**College**	Technical Engineering College of Computer and Artificial Intelligence - Kirkuk	
Module Leader			**e-mail**		
Module Leader's Acad. Title			**Module Leader's Qualification**		
Module Tutor			**e-mail**		
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date	2/11/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Programming Fundamentals	**Semester**	1
Co-requisites module	Operating Systems	**Semester**	3

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	This module aims to introduce students to the fundamental principles of computer organization and architecture. It focuses on the structure and operation of the Central Processing Unit (CPU), Instruction Set Architecture (ISA), ALU design, computer arithmetic, control unit design (hardwired vs. microprogrammed), instruction execution cycles, pipelining and hazard resolution, memory hierarchy design (cache memory mapping and replacement algorithms, virtual memory), Input/Output organization, DMA, and RISC vs. CISC architectural trade-offs.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Understand the historical evolution of computer systems, Von Neumann vs. Harvard architectures, and performance evaluation metrics. Comprehend Instruction Set Architecture (ISA) design, register allocation, instruction formats, and addressing modes. Analyze computer arithmetic operations (signed integers, Booth's algorithm, IEEE 754 floating-point standard) and ALU implementation. Master CPU datapath and control unit design principles (hardwired and microprogrammed control units). Evaluate instruction-level parallelism, instruction pipelining, pipeline hazards (data, control, structural), and branch prediction. Understand memory hierarchy design, cache organization (direct, associative, set-associative mapping), replacement policies, and virtual memory. Analyze I/O organization, interrupts, Direct Memory Access (DMA), and write/debug assembly programs using CPU architectural simulators (MARS/SPIM/Logisim).
Indicative Contents المحتويات الإرشادية	The module will cover the following key areas: Computer Evolution, Functional Units, Von Neumann Architecture, System Buses, and Performance Metrics (CPI, Clock Speed, Amdahl's Law) Instruction Set Architecture (ISA): Register Organization, Instruction Types, Instruction Cycle, and Addressing Modes Computer Arithmetic: Integer Operations, Adder/Subtractor Design, Booth's Multiplication Algorithm, Division, and IEEE 754 Floating-Point Standard Datapath and Control Unit Design: Single-Cycle Datapath, Multi-Cycle Datapath, Hardwired Control, and Microprogrammed Control Instruction Pipelining: Pipeline Stages, Data Hazards, Control Hazards, Structural Hazards, Stalling, Forwarding, and Branch Prediction Memory System Hierarchy: Semiconductor RAM/ROM, Memory Interleaving, Cache Principles, Mapping Schemes, and Cache Coherence Virtual Memory: Paging, Page Tables, Translation Lookaside Buffer (TLB), and Memory Management Unit (MMU) Input/Output Organization: Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA), and Bus Arbitration Schemes Advanced Architectures: RISC vs CISC Concepts, Superscalar Processors, Vector Processing, and Multi-Core Architectures

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم	
Strategies	The module will employ a combination of strategies including: Lectures: To provide theoretical foundations on CPU structures, ISA design, memory hierarchy, and pipelining. Group Discussions: To evaluate architectural trade-offs (RISC vs CISC, hardwired vs microprogrammed control). Problem-Solving Sessions: To work through computer arithmetic problems, address mapping calculations, and pipeline hazard analysis. Practical Lab Sessions: Hands-on experience developing assembly language programs and building digital logic datapaths using simulators (MARS/Logisim). Tutorials: To practice complex ISA instructions, cache hit/miss calculations, and virtual memory page table translations. Mid-term and final exams: To assess comprehensive theoretical knowledge and hardware-software hardware design capabilities. Preparatory Week: To give students time to revise and clarify any difficulties before the final exam.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	5% (5)	5, 10	LO #1, 3 and 5
	Assignments	2	5% (5)	2, 12	LO #2, 4 and 6
	Projects / Lab.	1	10% (10)	Continuous	LO #7
	Report	1	10% (10)	13	LO #3, 5 and 7
Summative assessment	**Midterm Exam**	2 hr	20% (20)	8	LO #1 - 4
	Final Exam	2hr	50% (50)	16	All

Total assessment	100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Computer Architecture: History, Von Neumann Model, System Buses, and Performance Evaluation Metrics (CPI, Execution Time).
Week 2	Instruction Set Architecture (ISA) I: Register Structure, Instruction Types, Instruction Execution Cycles, Assembly Language Principles.
Week 3	Instruction Set Architecture (ISA) II: Addressing Modes (Immediate, Direct, Indirect, Register, Displacement) and Instruction Formats.
Week 4	Computer Arithmetic I: Signed and Unsigned Integer Representation, Addition, Subtraction, and Arithmetic Logic Unit (ALU) Structure.
Week 5	Computer Arithmetic II: Multiplication Algorithms (Booth's Algorithm), Division Algorithms, and IEEE 754 Floating-Point Standard.
Week 6	Datapath and Control Unit I: Single-Cycle vs Multi-Cycle Datapath Design and Instruction Execution Control.
Week 7	Datapath and Control Unit II: Control Unit Implementation: Hardwired Control Unit vs Microprogrammed Control Unit.
Week 8	Mid-term Exam (الفصل منتصف اختبار) & Instruction Pipelining I: Pipelining Fundamentals, Pipeline Stages (IF, ID, EX, MEM, WB).
Week 9	Instruction Pipelining II: Pipeline Hazards (Structural, Data, Control Hazards), Data Forwarding, Stalling, Branch Prediction.
Week 10	Memory System Hierarchy I: Memory System Characteristics, Semiconductor RAM/ROM, Memory Interleaving, and DRAM Organization.
Week 11	Memory System Hierarchy II: Cache Memory Fundamentals, Cache Mapping Schemes (Direct, Fully Associative, Set-Associative Mapping).
Week 12	Memory System Hierarchy III: Cache Replacement Algorithms (LRU, FIFO), Write Policies (Write-Through, Write-Back), Cache Coherence.
Week 13	Virtual Memory Systems: Concept, Paging, Page Tables, Translation Lookaside Buffer (TLB), Address Translation, MMU.

Week 14	Input/Output Organization: Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA), and Bus Arbitration.
Week 15	Advanced Architectures: RISC vs CISC Trade-offs, Superscalar Execution, Vector Processing, Multi-Core Systems.
Week 16	**Course Review, Future Trends in Computer Architecture & Final Examination (النهائي الاختبار).**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Introduction to Assembly Language & MIPS/x86 Architecture Simulators Setup (MARS / SPIM / Logisim).
Week 2	Lab 2: Writing Basic Assembly Programs: Arithmetic, Logic Operations, and Data Transfer Instructions.
Week 3	Lab 3: Implementing Control Flow in Assembly: Conditional Branches, Jump Statements, and Loops.
Week 4	Lab 4: Procedures, Subroutines, Stack Management, and Parameter Passing in Assembly Language.
Week 5	Lab 5: Logisim CPU Datapath Simulation: Building a 4-bit/8-bit ALU, Registers, and Control Logic.
Week 6	Lab 6: Cache Memory Simulator Lab: Analyzing Cache Hit/Miss Rates, Block Sizes, and Mapping Strategies.
Week 7	Lab 7: Revision, CPU Architecture / Assembly Project Presentation, Practical Exam, and Assessment.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Stallings, W. (2018). "Computer Organization and Architecture: Designing for Performance" (11th ed.). Pearson.	Yes
Recommended Texts	Patterson, D. A., & Hennessy, J. L. (2020). "Computer Organization and Design: The Hardware/Software Interface" (6th ed. MIPS/RISC-V Edition). Morgan Kaufmann.	Yes
Websites	Yes	

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	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.