Unit 19: Computer Systems Architecture

Unit code: J/601/2003

QCF Level 3: BTEC National

Credit value: 10

Guided learning hours: 60

Aim and purpose

To enable learners to understand the underlying architecture and components behind the functioning of computer systems.

Unit introduction

All computer systems share the same underlying computer architecture principles. This unit examines these principles and explores the fundamentals of how computer systems work. Learners will focus on the technical detail, including how the components function at an electronic level.

Learners will explore how various types of data can be represented and then stored within computer systems. This is followed by a study of the low-level system components. It includes the processor, buses and memory incorporating an analysis of how these components interact to manipulate data using the fetch-execute cycle. Low-level program instructions make up the fetch-execute cycle, and simple assembly code instructions are investigated along with their interaction with the various registers that make up the Central Processing Unit (CPU). Learners will have the opportunity to develop simple programs in a low-level language.

Learning outcomes

On completion of this unit a learner should:

1 Understand how data can be represented within computer systems

2 Understand the functions of computer system components

3 Understand the principles of processor operations.

Unit content

1 Understand how data can be represented within computer systems

Numeric data: conversions between different representations of data; representing integer numbers in different number bases; converting between number bases using integer numbers eg denary to binary, denary to hexadecimal, binary to hexadecimal; performing arithmetic operations in different number bases; representing fixed-point numbers in different number bases; representing floating-point numbers in binary

Boolean logic: logic gates; truth tables; use of logic gates in integrated circuits; logical operations eg AND, OR, NOT, NAND, NOR, XOR

Coding of data: sign and magnitude; two's complement; floating point; binary coded decimal; coding of character data eg ASCII (American Standard Code for Information Interchange)

Types of data: representing bit patterns for different types of data eg graphics, video, audio and other data; graphics eg bitmap (resolution, colour depth, file calculations), vector (objects, properties); sound (compression, sampling resolution, sampling rate, streaming audio, quality); video (compression, encoding, streaming, quality); analogue data; digital data; analogue signals; digital signals; data conversion eg analogue to digital; file formats eg mp3, mp4, wav, avi

2 Understand the functions of computer system components

Key components: Central Processing Unit (CPU); memory; interfaces; clock; buses, diagrammatic representation; Von Neuman architectures

Central Processing Unit: control unit; ALU (Arithmetic Logic Unit); general purpose registers; special purpose registers eg instruction pointer, accumulator; core eg single, multiple; features eg pipelining, multiprocessing, parallel processing; polling; interrupts

Memory: I/O maps; Direct Memory Access (DMA); ROM (Read Only Memory); cache; RAM (Random Access Memory) eg static, dynamic, flash

Buses: system bus; address bus; control bus; physical connections to components eg Central Processing Unit, memory, input/output (I/O) devices, system buses

Peripherals: types eg hard disc, printer, scanner, network card

3 Understand the principles of processor operations

CPU instruction sets: Reduced Instruction Set Computer (RISC); Complex Instruction Set Computer (CISC); clock rate; performance levels

Addressing: modes eg immediate; relative; address bus; addressing in the fetch-execute cycle

Machine operations: how they are organised and represented; role of the instruction decoder; low-level programs; assembly code instructions eg fetch, load, add; decision making and branching; using registers, transferring data between registers, fetch-execute cycle; program storage; data storage; addressing

Assessment and grading criteria

In order to pass this unit, the evidence that the learner presents for assessment needs to demonstrate that they can meet all the learning outcomes for the unit. The assessment criteria for a pass grade describe the level of achievement required to pass this unit.

Assessment and grading criteria		
To achieve a pass grade the evidence must show that the learner is able to:	**To achieve a merit grade the evidence must show that, in addition to the pass criteria, the learner is able to:**	**To achieve a distinction grade the evidence must show that, in addition to the pass and merit criteria, the learner is able to:**
P1 explain, using examples, how numeric and alphanumeric data can be coded within a computer system [RL6]		
P2 explain, using examples, how different types of data can be converted and stored in computer systems		
P3 convert numeric data between different number systems including floating point	**M1** explain, using examples, how floating point numbers can be represented in binary	
P4 carry out Boolean logic operations		
P5 illustrate the key computer system components and how they interact		
P6 explain the different types of memory that can be attached to a processor	**M2** compare the roles played by different types of memory	**D1** explain how the processor is physically connected to memory and input/output (I/O) devices using the system buses
P7 explain how polling and interrupts are used to allow communication between processor and peripherals		
P8 compare Reduced Instruction Set Computer (RISC) chips and Complex Instruction Set Computer (CISC) chips		

Assessment and grading criteria					
P9	illustrate the use of the different processor registers in the fetch-execute cycle.	**M3**	create a low-level program which includes decision making and branching.	**D2**	explain how the width of the data bus and address bus affect processor performance and complexity.

PLTS: This summary references where applicable, in the square brackets, the elements of the personal, learning and thinking skills applicable in the pass criteria. It identifies opportunities for learners to demonstrate effective application of the referenced elements of the skills.

Key	IE – independent enquirers	RL – reflective learners	SM – self-managers
	CT – creative thinkers	TW – team workers	EP – effective participators

Essential guidance for tutors

Delivery

It is recommended that this unit is delivered either after or in conjunction with *Unit 2*: *Computer Systems*.

Simulation software could be used extensively in this unit, for example to develop learners' understanding of logic gates and low-level programming. Other software that simulates the internal operation of the processor would also be valuable to confirm learners' understanding.

Alternatively, a software application that simulates a simple CPU and an associated assembly language could be used. This type of software usually covers the basic elements of assembly language programming. Learners are able to put the theory into practice using the simulator to run programs in a controlled way, seeing all CPU activity step by step.

LO1 should be delivered using a series of worksheets to deliver the maths theory and Boolean logic operations. It's important to link the number theory to computer activity, noting that at the lowest level binary represents the off/on nature of electricity. For example, hexadecimal is a user-friendly way of representing binary and is used in electronics and computing. Use of binary and hexadecimal in IP addressing (versions 4 and 6) could be demonstrated.

For LO2 and LO3, the CPU can be introduced as the part of the computer architecture that runs computer programs. This is facilitated by the fetch-execute cycle and this could be explained in conjunction with a simple assembly language program that, for example, adds two numbers together. This could then be used to introduce learners to the different registers that make up the CPU. The functions of the various buses and the different types of memory could then be covered

Outline learning plan

The outline learning plan has been included in this unit as guidance and can be used in conjunction with the programme of suggested assignments.

The outline learning plan demonstrates one way in planning the delivery and assessment of this unit.

Topic and suggested assignments/activities and/assessment
Introduction to the unit
How data is represented within a computer system: • whole-class exercise – tutor presentation on numeric data, followed by practical exercise • whole-class exercise – tutor presentation on coding of data, followed by practical exercise • whole-class exercise – tutor presentation on representing analogue data, followed by practical exercise • a mixture of practical exploration of the technologies, learner exercises, case studies and detailed investigation.
Assignment 1 – The Devil is in the Data

Topic and suggested assignments/activities and/assessment
Low-level components of computer systems: • whole-class exercise – tutor presentation on architecture components, followed by practical exercise • whole-class exercise – tutor presentation on processors, followed by practical exercise • whole-class exercise – tutor presentation on buses, followed by practical exercise • whole-class exercise – tutor presentation on memory, followed by practical exercise • a mixture of practical exploration of the technologies, learner exercises, case studies and detailed investigation.
Processor operations: • whole-class exercise – tutor presentation on the operation and use of logic gates, followed by practical exercise • whole-class exercise – tutor presentation on representation of gates and logical circuits, followed by individual exercise • whole-class exercise – tutor presentation on gates and logic circuits, followed by practical exercise • mixture of practical exploration of the technologies, learner exercises, case studies and detailed investigation.
Assignment 2 – The Data Driver
Low-level programming: • whole-class exercise – tutor presentation on low-level programs, followed by practical exercise • whole-class exercise – tutor presentation on addressing modes, followed by practical exercise • individual exercise – research into uses of low-level programs
Assignment 3 – Low-level Programmer

Assessment

To achieve a pass grade, learners must achieve the nine pass criteria listed in the grading grid.

To achieve a merit grade, learners must achieve all of the pass grade criteria and the three merit grade criteria.

To achieve a distinction grade, learners must achieve all of the pass and merit grade criteria and the two distinction grade criteria.

Suggested Assignment 1 – The Devil is in the Data

For P1, appropriately designed short test questions could show understanding; alternatively some worked examples in a controlled open book environment would be appropriate. Whichever method is used, learners must explain in their own words how data can be coded, and they must use examples within their explanation.

For P2, learners could use combinations of diagrams, and descriptions that show coverage of the different types of data in the unit content. Alternatively, a presentation and demonstration could be appropriate.

P3 could be assessed through conventional testing, however learners do not need to be 100 per cent correct in all conversions in order to meet this criterion but they must eventually show competence. Evidence of learners' workings should be provided.

For P4, learners must carry out Boolean logic operations as specified in the content. This could be assessed through conventional testing as long as the criterion is covered. Evidence of learners' workings must be provided.

For M1, learners could investigate, for example, single and double precision floating point and show how numbers are stored in 16, 32, 64 bit etc registers.

Suggested Assignment 2 – The Data Driver

For P5, a complex and annotated diagram or series of related diagrams could be appropriate, perhaps stored in a web page with appropriate detail stored in hot spots. Confirmation of learners' understanding could be gained using a discussion which, if suitably documented, could also be part of the evidence.

For P6, a written report, presentation or web document could be used to explain the different types of memory attached to a processor.

For M2, learners could look at the role of processor cache memory, main memory and DMA.

For D1, learners could produce a diagram showing schematically the connectivity between processor and I/O devices with an explanation of, for example, how a document is printed or how a file is retrieved from disk.

Suggested Assignment 3 – Low-level Programmer

For P7, written responses to a series of scenarios could be used to generate evidence.

For P8, learners could research the different types of chips and look at the performance advantages and disadvantages of each.

For P9, learners could show diagrammatically the different registers involved in running a simple programme that, for example, adds two numbers together incorporating the fetch-execute cycle.

For M3, learners should produce listings of the programs, together with a statement from the tutor confirming that the program is valid.

For D2, learners could research bit sizes of current address and data buses and explain the effect these have on performance.

Programme of suggested assignments

The table below shows a programme of suggested assignments that cover the pass, merit and distinction criteria in the assessment and grading grid. This is for guidance and it is recommended that centres either write their own assignments or adapt any Pearson assignments to meet local needs and resources.

Criteria covered	Assignment title	Scenario	Assessment method
P1-P4, M1	The Devil is in the Data	A series of mathematical exercises based on number conversions using different representations of data. A series of Boolean algebra exercises demonstrating use of logic diagrams.	Website/portfolio Presentation
P5, P6, M2, D1	The Data Driver	A company requests a report illustrating how key computer system components interact. A further extension to explain different types of memory, and the role they play in computer systems.	Poster Presentation
P7-P9, M2, M3, D2	Low-level Programmer	A company requests further information about the relationship with the processor and peripherals. The company wants a comparison of RISC and CISC to help their understanding of computer systems architecture. A company asks you to demonstrate assembly programming by creating a simple low-level program for a specific need.	Documentation Working program

Links to other BTEC units

This unit forms part of the BTEC in IT sector suite. This unit has particular links with the following unit titles in the IT suite:

Level 1	Level 2	Level 3
		Unit 2: Computer Systems

Essential resources

Learners must have access to practical resources and suitable technology. They can also use simulators or multimedia tools to gain experience before handling 'live resources'.

Delivery of personal, learning and thinking skills

The table below identifies the opportunities for personal, learning and thinking skills (PLTS) that have been included within the pass assessment criteria of this unit.

Skill	When learners are ...
Reflective learners	communicate their learning in relevant ways for different audiences when explaining how data can be coded within a computer system.

Functional Skills – Level 2

Skill	When learners are ...
ICT – Using ICT	
Select, interact with and use ICT systems safely and securely for a complex task in non-routine and unfamiliar contexts	carrying out Boolean operations and converting between number systems producing a working program
ICT – Finding and selecting information	
Use appropriate search techniques to locate and select relevant information	preparing a report or presentation on computer system components
Select information from a variety of sources to meet requirements of a complex task	preparing a report or presentation on computer system components
ICT – Developing, presenting and communicating information	
Combine and present information in ways that are fit for purpose and audience	presenting a report on computer system components
Mathematics – Representing	
Understand routine and non-routine problems in familiar and unfamiliar contexts and situations	converting numeric data between different number systems, including floating point
Identify the situation or problems and identify the mathematical methods needed to solve them	converting numeric data between different number systems, including floating point.