

- 1(a) The stored program concept uses the Fetch-Decode-Execute cycle to get the next instruction from memory and then execute it.

Describe what happens during the **fetch** stage of the Fetch-Decode-Execute cycle.

You should state the different registers and buses that are used in your answer.

[4]

- (b) One of the instructions that may be fetched and executed as part of this cycle is a branch instruction.

State the name of the register that would be altered in the **execute** phase during a branch instruction.

[1]

- Explain how pipelining improves the performance of a CPU.

[3]

- Give **two** features of a replacement processor that would increase the typical performance of the computer.

1 _____

2 _____

[2]

(b) The processor contains registers including the accumulator and the program counter. The contents of these registers are modified during the Fetch-Decode-Execute cycle.

i. Describe how the accumulator is used during the Fetch-Decode-Execute cycle.

[2]

ii. Describe how the program counter is used during the Fetch-Decode-Execute cycle.

[2]

iii. State the name of **three** other registers that are used during the Fetch-Decode-Execute cycle.

1 -----

2 -----

3 -----

[3]

- (c) A charity uses a desktop computer to record financial donations that it receives. The computer contains a single core, 2.4GHz processor with 2MB cache.

A processor uses the Von Neumann architecture.

- i. Describe what is meant by the term ‘Von Neumann architecture’.

----- [2]

- ii. Give **one** way that the Harvard architecture differs from the Von Neumann architecture.

----- [1]

3 A program written using the Little Man Computer instruction set is shown in Fig. 1.

```

      INP
      STA numone
      INP
      STA numtwo
main   LDA numone
      SUB numtwo
      BRP pos
notpos LDA count
      OUT
      LDA numone
      OUT
      HLT
pos    STA numone
      LDA count
      ADD one
      STA count
      BRA main
numone DAT
numtwo DAT
one    DAT 1
count  DAT 0
```

Fig. 1

In Von Neumann Architecture, various registers are used when a program is executed.

i. State what is meant by the term 'register'.

[2]

ii. Explain how the accumulator is used when the line BRP pos is executed.

[2]

4(a) The CPU uses pipelining to improve efficiency.

Explain what is meant by the term 'pipelining'.

[3]

(b) OCR Insurance uses a computer system to calculate the price that customers pay for car insurance.

The computer system contains a CPU, GPU, RAM and ROM.

i. State **two** factors that affect the performance of a CPU.

1

2

[2]

ii. Explain the difference between RAM and ROM, including how these are used by the computer system.

[4]

iii. Describe **one** non-graphical use for a GPU.

[2]

5(a) The table below shows the Little Man Computer instruction set.

Mnemonic	Instruction
ADD	Add
SUB	Subtract
STA	
LDA	Load
	Branch always
BRZ	
BRP	
INP	Input
OUT	Output
	End program

Complete the table above to show the missing mnemonics and instructions. [5]

- Blank lined paper for writing.

[6]

6(a) A program written in the Little Man Computer instruction set is given below.

```

      INP
      STA    num
loop  LDA    total
      ADD    num
      STA    total
      LDA    count
      ADD    one
      STA    count
      SUB    num
      BRZ    end
      BRA    loop
end   LDA    total
      OUT
      HLT
one   DAT    1
num   DAT    0
count DAT    0
total DAT    0
```

State what the purpose of the program is.

----- [1]

(b) Explain which registers are used and their values when the line STA count is **executed** and the accumulator is holding the value 9. The label count refers to memory location 16.

----- [2]

7(a) A digital coffee making machine has a CPU that uses the Little Man Computer Instruction Set.

Little Man Computer operates on a computer system based on the Von Neumann Architecture.

i. State **two** features of the Von Neumann architecture.

1

2

[2]

ii. Describe **one** feature, **not** part of the standard Von Neumann Architecture, which contemporary CPUs may have in order to improve performance.

[2]

- (b) Part of the coffee making machine’s code asks the user to press a button to select strength. The code outputs 1 which will switch on a green light to indicate a valid selection or outputs 0 to indicate an invalid selection.

The code is shown below:

```

                                INP
                                STA    entry
                                LDA    max
                                SUB    entry
                                BRP     accept
                                LDA    redLight
                                BRA    printAndEnd
accept                          LDA    greenLight
printAndEnd                    OUT
                                HLT

greenLight                     DAT    1
redLight                       DAT    0
max                             DAT    5
entry                          DAT

```

Fig. 1

- i. Tick the appropriate boxes below to indicate which inputs will result in a green light (i.e. code outputs 1) and which with a red light.

Input	Green Light	Red Light
1		
2		
3		
4		
5		
6		
7		
8		
9		

[2]

- ii. Explain which registers and buses are used, and the values they store/carry, when the line `LDA redLight` is executed (after it has been fetched and decoded). You should assume the address `redLight` refers to memory location 11.

[6]

iii. Write code in a high-level language or pseudocode that has the same functionality as the code in Fig. 1.

- iv. * Discuss the differences between assembly code and high-level languages. You should refer to:
- the advantages and disadvantages of writing programs in assembly code rather than a high-level language
 - when each approach might be used
 - why the coffee machine was programmed in assembly code.

----- [9]

8 An architect firm specialises in designing skyscrapers.

The firm uses high end computers with high performance CPUs, GPUs and large amounts of RAM.

i. Give one use the firm might have for GPUs.

----- [1]

ii. Describe what is meant by the term 'RAM'.

----- [2]

iii. State **one** characteristic a high performance CPU might have.

----- [1]

The complete program written using the Little Man Computer instruction set is shown below:

	INP	
main	STA	entry
	BRA	test
fail	LDA	entry
	ADD	one
	BRA	main
test	SUB	ten
	BRZ	pass
	BRP	test
	BRA	fail
pass	LDA	entry
	OUT	
	HLT	
entry	DAT	
ten	DAT	10
one	DAT	1

- i. Give **one** instruction in the program that when executed, changes the value in the Accumulator.

----- [1]

- ii. Give **one** instruction in the program that when executed, changes the value in the Program Counter.

----- [1]

iii. State the value the code outputs for the input 18.

----- [1]

iv. State the value the code outputs for the input 37.

----- [1]

v. Describe the purpose of the program.

----- [2]

END OF QUESTION PAPER

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
1	a	1 mark per bullet to max 4: - The contents of the Program Counter/PC are copied/sent to the Memory Address Register/ MAR - The address is sent/transferred along the <u>address bus</u> - The <u>control unit</u> sends/transfers a (read) signal along the <u>control bus</u> - Contents stored in the memory address are sent/transferred along the <u>data bus</u> - Contents (from memory) are stored in the Memory Data Register/MDR ...and sent/copied to the Current Instruction Register/ CIR - The Program Counter/PC is incremented	4	Accept IR/Instruction Register for CIR/Current instruction Register Accept MBR/Memory Buffer Register for MDR/Memory Data Register <u>Examiner's Comments</u> Candidates who had good knowledge of the FDE cycle were able to gain 3 or 4 marks on this question. Some candidates did not have a good understanding of what a register or a bus is and tended to say they did something other than transport or temporarily store. Many however gave clear and correct responses.  Misconception Some candidates gave responses which mentioned a register fetching the data or passing data. A register is a temporary store for data/instructions/addresses and as such doesn't fetch or pass anything. Exemplar 1 <i>The PC holds the next instruction's address which is copied to the MAR, memory address register. The Program Counter is incremented. The address from the MAR is sent on the address bus to the memory. Simultaneously a fetch/read signal is sent on the control bus by the control unit. The data or instruction goes on the data bus and is copied into the Memory Data Register. It is then copied into the CIR.</i> [4] This candidate has shown a clear understanding of the role of the buses and registers in the fetch part of the FDE cycle.
	b	Program Counter // PC	1	<u>Examiner's Comments</u> Candidates who understood the FDE cycle gained the mark. Those that did not tended to mention a register used in the fetch or decode stages.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
	c		1 mark per bullet: - Allows the next <u>instruction</u> to be fetched whilst the previous one is being decoded/executed //allows the overlapping of different parts of the FDE - It increases throughput // increases the number of instructions processed in a set period of time - It prevents the CPU having to wait // prevents idle components	3	DNA responses that talk about the FDE running faster. Allow a diagram demonstrating pipelining for MP1. Allow 'it will take less time to do the same amount of instructions' for MP2 BOD 'processes' for 'instructions' for MP2. DNA 'more efficient' on its own for MP2 DNA points if clearly discussing multiple cores. <u>Examiner's Comments</u> Many candidates were able to gain a mark for an instruction being fetched at the same time as another is decoded. Some candidates discussed multiple cores which were mentioned in the question stem but did not apply to this question. Candidates should have access to previous mark schemes.
			Total	8	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
2	a		• Higher/faster clock speed • More cores//dual/quad/etc core • More cache memory.	2	Answers must refer to an improvement (more/higher/faster) not just “change the clock speed” Allow discussions of level 1/level 2 cache sizes for one mark. Accept valid features of CPUs that would improve performance e.g. Use of: Pipelining Simultaneous Multithreading Do not accept RISC/CISC. Examiner's Comments Most candidates were able to gain full marks on this question. Less successful responses often mentioned clock speed, cache or cores without referring to an improvement, e.g. higher or faster.
	b	i	• Holds all input/output • Holds results of calculations (from the ALU) • Checked for conditional branching (e.g. BRZ) • Stores data which has come from the MDR/RAM	2	Examiner's Comments Most candidates were able to access 1 mark for the result of ALU calculations, but few were able to give two uses. Some confused the accumulator with the program counter and the ALU.
		ii	• Holds the <u>address/location</u> of the <u>next</u> instruction (to be executed/fetched) • Contents copied to the MAR at start of FDE • Incremented (by one) on every cycle • Can be changed by branch/jump instructions	2	Examiner's Comments This question was generally well answered by candidates who gave clear responses. Misconception  Some candidates thought that the program counter kept track of a count of the number of instructions that had been fetched.
		iii	• Memory Address Register // MAR • Memory Data Register // MDR • Current Instruction Register // CIR • Index Register // IR	3	Allow Memory Buffer Register for MDR Examiner's Comments Most candidates gained full marks on this question and were able to correctly identify three other registers. Some lost marks for saying the ALU or control unit were registers.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
	c	i	• <u>Both data and instructions</u> share the same memory • <u>Instructions and Data</u> stored in same format • A single set of buses / same <u>bus for instructions & data</u> (to connect CPU to Memory and I/O) • Has a (single) control unit • Has an ALU. • Has ways to input and output. • Has access to storage, • Works sequentially through instructions // follows Fetch-execute cycle • (Special) registers within CPU • Based on stored program concept	2	<u>Examiner's Comments</u> Many candidates were able to access full marks on this question. This question has been asked in previous papers and candidates should be encouraged to use these to make sure they are clear in their responses. There were many possible responses in the mark scheme to help candidates to gain full marks. Most candidates gained at least 1 mark.
		ii	• Separate <u>memory for data and instructions</u> / Multiple memory units • Different (sets of) buses one for <u>instructions</u> & one for <u>data/ instructions and data</u> can be accessed concurrently.	1	<u>Examiner's Comments</u> This question was generally answered well by candidates and the majority gave separate areas of memory for data and instructions. Where candidates were not given marks, it was generally because their answer was unclear, e.g. just saying 'separate memory'.
			Total	12	
3		i	• Temporary storage/memory location... • ...inside the CPU • Used for a single specific purpose • Faster access speed than RAM / secondary storage	2 AO1.1	
		ii	• Accumulator checked to see if value held is positive or <u>zero</u> • If so, BRANCH carried out / jumps to specified location.	2 AO1.2	
			Total	4	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
4	a		- Concurrent processing of multiple instructions - One instruction can be fetched while previous is being decoded... - And the one before is being executed. - In case of a branch pipeline is flushed. - Increases speed of execution	3 AO1.1	
	b	i	- Clock speed - Number of cores - Cache	2 AO1.1	Accept Use of pipelining/size of pipeline Use of out of order execution Use of SIMD instructions Integrated graphics processing on CPU
		ii	- RAM is <u>volatile</u> - Used for storing programs/data/parts of OS <u>currently in use</u> - ROM is <u>non volatile</u> - Used for storing (e.g.) BIOS / bootstrap	4 AO1.2	
		iii	- Performing complex numerical calculations - Calculations on matrices / vectors / multiple data at the same time ...e.g. insurance pricing, modelling risk, calculating bills	2 AO2.2	Example has to relate to insurance company
			Total	11	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance																						
5	a		• Store value in accumulator at address given • BRA // BR • Branch if zero • Branch if <u>zero or positive</u> • HLT // COB // END	5 AO1.1	<table><tr><th>Mnemonic</th><th>Instruction</th></tr><tr><td>ADD</td><td>Add</td></tr><tr><td>SUB</td><td>Subtract</td></tr><tr><td>STA</td><td>Store value in accumulator at address given</td></tr><tr><td>LDA</td><td>Load (to accumulator)</td></tr><tr><td>BRA</td><td>Branch always</td></tr><tr><td>BRZ</td><td>Branch if zero</td></tr><tr><td>BRP</td><td>Branch if zero or positive</td></tr><tr><td>INP</td><td>Input</td></tr><tr><td>OUT</td><td>Output</td></tr><tr><td>HLT</td><td>End program</td></tr></table>	Mnemonic	Instruction	ADD	Add	SUB	Subtract	STA	Store value in accumulator at address given	LDA	Load (to accumulator)	BRA	Branch always	BRZ	Branch if zero	BRP	Branch if zero or positive	INP	Input	OUT	Output	HLT	End program
Mnemonic	Instruction																										
ADD	Add																										
SUB	Subtract																										
STA	Store value in accumulator at address given																										
LDA	Load (to accumulator)																										
BRA	Branch always																										
BRZ	Branch if zero																										
BRP	Branch if zero or positive																										
INP	Input																										
OUT	Output																										
HLT	End program																										
	b		• Inputs two numbers • ..stores at least one of them • Comparison / subtraction to decide which is larger • Jump / output if num1 larger • Jump / output if num2 larger <u>or nums equal</u> • Loops back to start after either output	6 AO3.2	<u>Example answer</u> start INP STA x INP STA y SUB x BRP first LDA x OUT BRA start first LDA y OUT BRA start x DAT y DAT																						
			Total	11																							

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
6	a		Squares a number / multiplies a number by itself	1 AO3.3	<u>Examiner's Comments</u> Many candidates achieved full marks on part (a) and went on to achieve the mark on part (b). Those who did not score well on part (a) could not determine the purpose of the program and therefore did not achieve the part (b) mark.
	b		– The value <u>16</u> is copied to the MAR – The contents of the of the ACC (i.e. 9) are copied to the MDR – The value <u>9</u> is copied to location <u>16/count</u> 1 per -, max 2	2 AO2.2	<u>Examiner's Comments</u> It is encouraging that many more candidates are appropriately applying their response to the scenario, in this type of question, than in previous series. However, there are still some candidates giving a description of the order of data movement between registers without referring to the data or address values / labels given in the question. Candidates should be reminded to refer to the values given in this type of question in their response.
			Total	3	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
7	a	i	- (Single) Control Unit - (Single) Arithmetic Logic Unit - (Special) registers within CPU - Instructions and Data stored in same area of memory - Instructions and Data stored in same format - A single set of buses / same bus for instructions & data (to connect CPU to Memory and I/O) (1 Mark per -, Max 2)	2 (AO1.1)	Accept acronyms ALU, CU <u>Examiner's Comments</u> The majority of candidates answered this question well although some candidates stated that 'instructions and data are stored in the same memory location' more attention to detail is required at this level of study.
		ii	Two separate areas of memory... ...one for instructions & one for data./instructions and data can be accessed concurrently. Different (sets of) buses... ... one for instructions & one for data./ instructions and data can be accessed concurrently. Pipelining... ...whilst an instruction is being executed the next can be decoded and the subsequent one fetched. Use of Cache... ...A small amount of high performance memory is (next to the CPU) / which stores frequently used data/instructions Virtual cores/Hyper-threading™ Treating a physical core as two virtual cores. Multiple Cores... ...Each core acts as a separate processing unit. Onboard Graphics... ...Built in circuitry for graphics processing. (1 Mark for identifying feature, 1 mark for description)	2 (AO1.2)	Accept any reasonable description. Do not accept "64-bit" e.g. Performance boosting mode... ...Clock speed can be temporarily increased for performance boost. Out of Order Execution... ...Instructions can be executed before earlier ones if they are ready. Super Scalar... ...Multiple instructions can be executed simultaneously. <u>Examiner's Comments</u> A whole range of features were accepted for this question. Most candidates stated an appropriate feature but some then did not go on to describe how the feature improved performance.

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance																														
	b	i	<table><thead><tr><th>Input</th><th>Green Light</th><th>Red Light</th></tr></thead><tbody><tr><td>1</td><td>✓</td><td></td></tr><tr><td>2</td><td>✓</td><td></td></tr><tr><td>3</td><td>✓</td><td></td></tr><tr><td>4</td><td>✓</td><td></td></tr><tr><td>5</td><td>✓</td><td></td></tr><tr><td>6</td><td></td><td>✓</td></tr><tr><td>7</td><td></td><td>✓</td></tr><tr><td>8</td><td></td><td>✓</td></tr><tr><td>9</td><td></td><td>✓</td></tr></tbody></table> Rows 1-4 correct 1 Mark Rows 5-9 correct 1 Mark	Input	Green Light	Red Light	1	✓		2	✓		3	✓		4	✓		5	✓		6		✓	7		✓	8		✓	9		✓	2 (AO3.3)	Accept T for a tick. Penalise if blank table elements have content. <u>Examiner's Comments</u> Most candidates achieved the first mark on this question. The second mark was lost by those who ticked 'red light' for an input of 5.
Input	Green Light	Red Light																																	
1	✓																																		
2	✓																																		
3	✓																																		
4	✓																																		
5	✓																																		
6		✓																																	
7		✓																																	
8		✓																																	
9		✓																																	
		ii	- The value 11 is stored in the MAR. - 11 is sent down the address bus. - A read signal is sent down the control bus. - <u>Q</u> is sent (back from memory) down the data bus. - 0 is stored in the MDR... - ...and then copied to the ACC (1 Mark per -, max 6)	6 (AO1.2)	<u>Examiner's Comments</u> Many candidates gave a comprehensive description of the fetch execute cycle but did not apply their response to the assembly code instruction given in the question therefore losing marks.																														
		iii	- Takes in a value from user. - If value is 5 or less it shows green - Otherwise it shows Red (1 Mark per -, max 3)	3 (AO 3.2)	Do not credit structured English Example value = input("Enter a Value") if value <=5 then print("GREEN") else print("RED") endif Accept equivalents to <=5 (e.g. < 6) For Green/Red (or 1/0) accept any pseudocode equivalent (GreenLightOn(), Output 1, print(1) Output Green etc.) as long as the logic is correct. <u>Examiner's Comments</u> Most candidates gained credit for writing code to read the value from the user with some then losing marks for either specifying the incorrect condition e.g. IF value < 5 and/or not outputting the result.																														

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
		iv	Mark Band 3-High Level (7-9 marks)¹ The candidate demonstrates a thorough knowledge and understanding of assembly code and high level languages. The material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well-considered. There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated. Mark Band 2-Mid Level (4-6 marks) The candidate demonstrates reasonable knowledge and understanding assembly code and high level languages; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a sound discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence. Mark Band 1-Low Level (1-3 marks) The candidate demonstrates a basic knowledge assembly code and high level languages; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided.	9 AO1.1 (2) AO1.2 (2) AO2.1 (2) AO3.3 (3)	AO1 Assembly code uses mnemonics to represent machine code instructions/opcodes. High level languages use more natural/mathematical notation. Assembly code consists of simple instructions As such many more lines of assembly code are required to perform the same task as a few lines of a high level language. Assembly code is specific to the instruction set of a given processor. High Level languages are not architecture specific. AO2 Assembly code allows the programmer to choose the exact instructions so they can write code that is highly efficient. It also allows them to have direct control of how memory is used via addressing modes. Direct control of hardware. High level language compilers have optimisers that can also try and do this (and in some cases may outperform a human writing in assembly code). As high level code is more intuitive and easier to read it is easier to follow, debug and build as part of a team. It can also be written in a much shorter time frame. The high level code can be recompiled for different architectures. High level languages come in a variety of paradigms so programmers can choose according to the problem/their preference. AO3 Assembly language is best suited to situations such as: -compilers or interpreters don't exist for the target CPU i.e. embedded systems -highest possible performance is critical -memory is very limited. For larger projects which don't fall under the constraints above high level languages are likely to be preferable. <u>Examiner's Comments</u> Candidates were assessed on the quality

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
			The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear. 0 marks No attempt to answer the question or response is not worthy of credit.		of their extended response in this question. Most candidates could describe the basic differences between assembly code and high level languages, with many giving examples of where each would be best used. Some candidates gave clear and appropriate justification for the coffee machine being programmed in assembly code. In general, most candidates scored reasonably well on this question.
			Total	24	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
8		i	To render models of proposed buildings. (1) Run CAD software. (1) Run modelling calculations. (1) Any example sensible to scenario. (1) (Max 1)	1 (AO2.1)	Examiner's Comments Some candidates did not apply the use of GPU to the scenario. Those who did, generally gained full marks for this question.
		ii	Random Access Memory (1) A form of primary memory (1) Used to hold data and / or programs <u>in use</u> (1) Volatile / Loses its contents when power is lost. (1) (Max 2)	2 (AO1.1)	
		iii	Multiple Cores (1) High / Fast Clock Speed (1) Ability to use pipelining (1) Large Cache (1) (Max 1)	1 (AO1.1)	Accept concurrency / parallel processing for pipelining
			Total	4	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
9		i	LDA (1) SUB (1) ADD (1) INP (1) (Max 1)	1 (AO1.2)	Examiner's Comments Most candidates identified correctly, an instruction which changed the value in the Accumulator but fewer correctly identified an instruction which changed the value in the Program Counter.
		ii	BRA (1) BRP (1) BRZ (1) (Max 1)	1 (AO1.2)	Examiner's Comments Most candidates identified correctly, an instruction which changed the value in the Accumulator but fewer correctly identified an instruction which changed the value in the Program Counter.
		iii	20	1 (AO2.1)	Examiner's Comments Candidates invariably gave both correct output values.
		iv	40	1 (AO2.1)	Examiner's Comments Candidates invariably gave both correct output values.
		v	Rounds up (the number input)... (1) ...To the nearest multiple of ten (and outputs it) (1)	2 (AO2.2)	Rounds to multiple of ten gets one mark.
			Total	6	