1.1 PROCESSORS, INPUT/OUTPUT AND STORAGE · 1.1.1

CPU architecture & the fetch-decode-execute cycle — Mark scheme

39 marks · spec 1.1.1(a)(b)(e)

AO key: AO1 = knowledge & understanding · AO2 = application to context · AO3 = make reasoned judgements. Accept any valid alternative wording. Do not award the same point twice.

Q	ANSWER	AO	MARKS
1(a)	A small, very fast storage location/cell inside the CPU/processor (that temporarily holds data, an instruction or an address). (1)	AO1	1
1(b)	Program Counter (PC). (1) Do not accept CIR.	AO1	1
1(c)	Address bus. (1)	AO1	1
1(d)	1 mark per valid operation, max 2: add / subtract / multiply / divide / AND / OR / NOT / XOR / shift / compare. (2)	AO1	2

Q	ANSWER	AO	MARKS
2(a)	1 mark per point, max 4: • the address of the next instruction is copied from the PC to the MAR (1) • the instruction at that address is copied into the MDR (along the data bus) (1) • the PC is incremented (1) • the contents of the MDR are copied into the CIR (1)	AO1	4
2(b)	Current Instruction Register (CIR). (1)	AO1	1
2(c)	• so that it holds/points to the address of the next instruction (1) • so the correct instruction is fetched on the next cycle / instructions are executed in sequence (1)	AO2	2

Q	ANSWER	AO	MARKS
3(a)	1 mark per point, max 4: • the address (45) is copied into the MAR (1) • the address is sent to memory (along the address bus) (1) • the data held at address 45 is copied into the MDR (1) • (via the data bus) the data is then passed to the ALU/accumulator to be added (1)	AO2	4
3(b)	Accumulator (ACC). (1)	AO1	1
3(c)	1 mark per point, max 3: • the address bus carries the address (45) from the CPU to memory (1) • the address bus is one-directional / out of the CPU only (1) • the data bus carries the data from memory back to the CPU (1) Allow the reverse mark for stating the data bus is bidirectional.	AO2	3

Q	ANSWER	AO	MARKS
4(a)	• holds the address (1) • of the memory location to be read from or written to (1)	AO1	2
4(b)	1 mark per point, max 3: • the address bus has a fixed number of lines/bits (1) • n lines can represent 2^n different addresses (1) • so a wider bus can address more memory / e.g. 32 lines → 2^{32} locations (1)	AO1 / AO2	3

Q	ANSWER	AO	MARKS
5(a)	1 mark per point, max 2: • a single/shared memory holds both data and instructions (1) • a single shared bus system is used for both / the stored-program concept with instructions executed serially (1)	AO1	2
5(b)	It uses physically separate memories (and buses) for instructions and data. (1)	AO1	1
5(c)	1 mark for identifying a feature + 1 mark for description, max 2. E.g. • split cache: separate instruction and data caches (1) so the core can fetch instructions and data in parallel (Harvard-style) at the cache (1) Allow pipelining, multiple cores, or multiple internal data buses if correctly described.	AO1	2

<table>
<tr><th>Q</th><th>LEVELS-OF-RESPONSE MARK SCHEME</th><th>AO</th><th>MARKS</th></tr>
<tr><td>6</td><td>Mark using the levels descriptors below. AO1 (knowledge of the two architectures), AO2 (application to the embedded DSP context) and AO3 (justified conclusion).
<table><tr><th>LEVEL</th><th>DESCRIPTOR</th></tr><tr><td>Level 3 (7–9)</td><td>Thorough, balanced discussion of both architectures with accurate advantages and disadvantages, clearly applied to the embedded DSP context, leading to a justified conclusion. Accurate terminology; well structured.</td></tr><tr><td>Level 2 (4–6)</td><td>Discusses both architectures with some accurate points and some application to the context. A conclusion is reached but justification may be partial or one-sided.</td></tr><tr><td>Level 1 (1–3)</td><td>Basic points about one or both architectures with little or no application to the context and little or no justified conclusion.</td></tr><tr><td>0</td><td>Nothing creditworthy.</td></tr></table>
Indicative content (not exhaustive; credit any valid point):
• Von Neumann: single shared memory/bus for instructions and data; simpler and cheaper; suffers the Von Neumann bottleneck as data and instructions cannot move at the same time.
• Harvard: separate instruction and data memories and buses; instructions and data can be fetched simultaneously, so faster; instructions can sit in ROM and data in RAM, which suits an embedded device.
• Application: the controller fetches fixed program instructions while continuously processing a high-speed audio stream, so simultaneous instruction and data access is valuable; the bottleneck would limit a Von Neumann design.
• Justified conclusion: Harvard is more suitable here; reference to contemporary processors using a split cache to gain similar benefits is creditworthy.</td><td>AO1 ×3
AO2 ×3
AO3 ×3</td><td>9</td></tr>
</table>

Total for paper: 39 marks