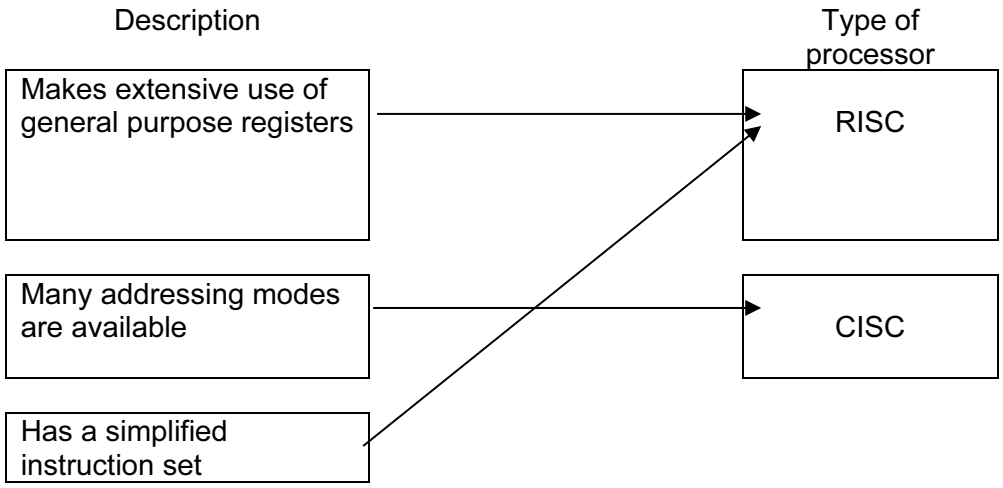
QUESTION 1.

4 (a)



I mark for correct arrow from each description

[3]

(b) (i)

Time Interval

stage	1	2	3	4	5	6	7	8	9
Fetch instruction	A	B	C						
Decode instruction		A	B	C					
Execute instruction			A	B	C				
Access operand in memory				A	B	C			
Write result to register					A	B	C		

Completing the As (1 Mark)

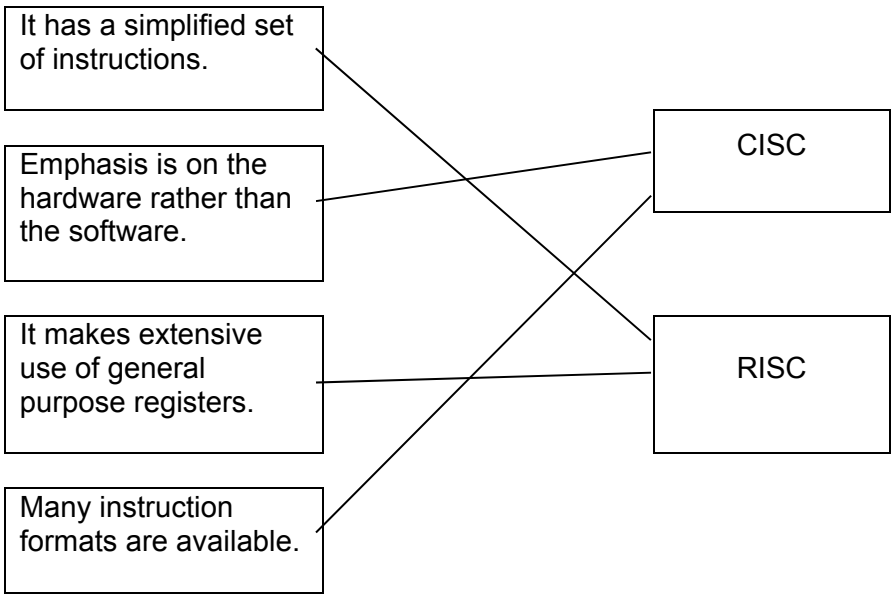
B in column 2, Row 1 (1 Mark)

Remainder completed (1 Mark)

[3]

(ii) With pipelining no of cycles = 7 [1]
Without pipelining no of cycles = 3 * 5 = 15 [1]
No of cycles saved = 8 [1]

QUESTION 2.

Question	Answer	
2(a)	**Description** / **Type of processor**  1 mark for each correct line	
2(b)(i)	**One mark** per point – **max 2** • Pipelining is instruction level parallelism • Execution (A: processing) of an instruction is split into a number of stages • When first stage for an instruction is completed the first stage of the next instruction can start executing • Another instruction can start executing before the previous one is finished • Processing of a number of instructions can be concurrent / simultaneous	**2**
2(b)(ii)	**Time Interval** **Stage** / **1** / **2** / **3** / **4** / **5** / **6** / **7** / **8** Fetch instruction: D **E** Read registers and decode instruction: D E Execute instruction: D E Access operand in memory: D E Write result to register: D E D at time interval 1 (1) D and E in second row (in that order) (1) Remainder completed correctly (1)	**3**

Question	Answer	
2(c)(i)	**Two** from: • The result of the first addition is not stored in (register) r3 (1) • Before the next instruction needs to load value from r3 (1) • There is a data dependency issue (1) • r3 is being fetched and stored on the same clock pulse (1)	
2(c)(ii)	The third instruction is not dependent on the first two, therefore, instruction 2 and 3 need to be swapped	**1**

QUESTION 3.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>1 mark per bullet point to max 4:
• RISC has fewer instructions // CISC has more instructions
• RISC has many registers // CISC has few registers
• RISCs instructions are simpler // CISC's instructions are more complex
• RISC has a few instruction formats // CISC has many instruction formats
• RISC usually uses single-cycle instructions // CISC uses multi-cycle instructions
• RISC uses fixed-length instructions // CISC uses variable-length instructions
• RISC has better pipelineability // CISC has poorer pipelineability
• RISC requires less complex circuits // CISC requires more complex circuits
• RISC has fewer addressing modes // CISC has more addressing modes
• RISC makes more use of RAM // CISC makes more use of cache/less use of RAM
• RISC has a hard-wired control unit // CISC has a programmable control unit
• RISC only uses load and store instructions to address memory
// CISC has many types of instructions to address memory</td><td>4</td></tr>
<tr><td>5(b)(i)</td><td>1 mark per bullet point:
• Completing the As correctly
• B in column 2, row 1 no other Bs in row 1
• Remainder correctly completed

<table>
<tr><th rowspan="2">Stage</th><th colspan="9">Time interval</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th></tr>
<tr><td>Fetch instruction</td><td>A</td><td>B</td><td>C</td><td>D</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Decode instruction</td><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Execute instruction</td><td></td><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td></td><td></td><td></td></tr>
<tr><td>Access operand in memory</td><td></td><td></td><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td></td><td></td></tr>
<tr><td>Write result to register</td><td></td><td></td><td></td><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td></td></tr>
</table></td><td>3</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(b)(ii)</td><td>1 mark per bullet point:
• Correct number of cycles for pipelining 8
• Correct number of cycles without pipelining 4 × 5 = 20
• No of cycles saved 20 – 8 = 12</td><td>3</td></tr>
<tr><td>5(c)</td><td>1 mark for each row
<table><tr><th rowspan="2">Statement</th><th colspan="3">Architecture</th></tr><tr><th>SIMD</th><th>MIMD</th><th>SISD</th></tr><tr><td>Each processor executes a different instruction</td><td></td><td>✓</td><td></td></tr><tr><td>There is only one processor</td><td></td><td></td><td>✓</td></tr><tr><td>Each processor executes the same instruction input using data available in the dedicated memory</td><td>✓</td><td></td><td></td></tr><tr><td>Each processor typically has its own partition within a shared memory</td><td></td><td>✓</td><td></td></tr></table></td><td>4</td></tr>
</table>

QUESTION 4.

<table>
<tr><th>Question</th><th>Answer</th><th></th></tr>
<tr><td>7(a)</td><td>1 mark for 2/3 rows correct
2 marks for 4/5 rows correct
3 marks for 6 correct rows

<table><tr><th>Statement</th><th>RISC</th><th>CISC</th></tr><tr><td>Larger instruction set</td><td></td><td>✓</td></tr><tr><td>Variable length instructions</td><td></td><td>✓</td></tr><tr><td>Smaller number of instruction formats</td><td>✓</td><td></td></tr><tr><td>Pipelining is easier</td><td>✓</td><td></td></tr><tr><td>Microprogrammed control unit</td><td></td><td>✓</td></tr><tr><td>Multi-cycle instructions</td><td></td><td>✓</td></tr></table></td><td></td></tr>
<tr><td>7(b)(i)</td><td>1 mark per bullet point

• SISD // Single instruction single data
• SIMD // Single instruction multiple data
• MISD // Multiple instruction single data
• MIMD // Multiple instruction multiple data</td><td>4</td></tr>
<tr><td>7(b)(ii)</td><td>1 mark per bullet point (max 3)

• Large number of processors
• … working collaboratively on the same program
• … working together simultaneously on the same program
• … communicating via a messaging interface</td><td>3</td></tr>
</table>