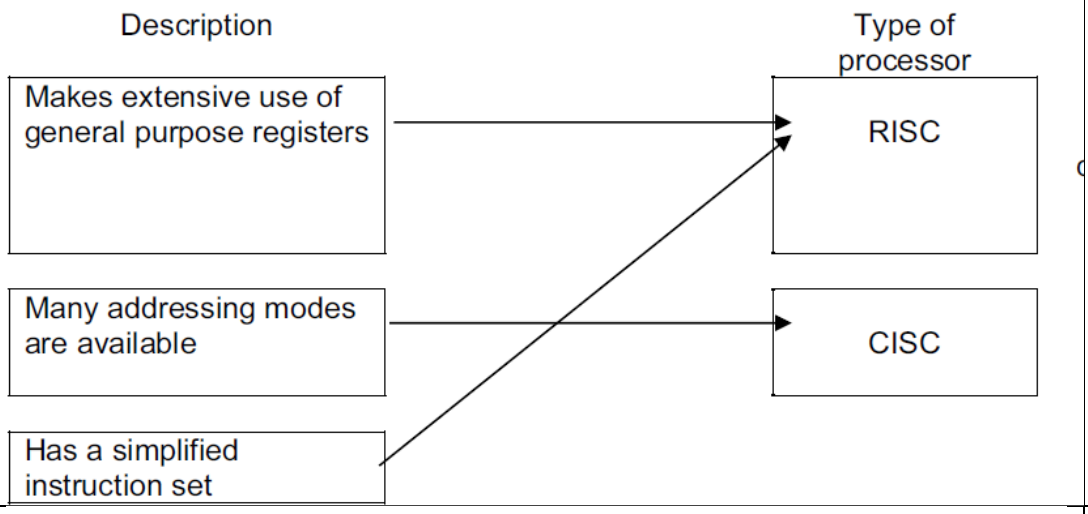
Question No. 1

a) — **3**



b) i) — **3**

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)

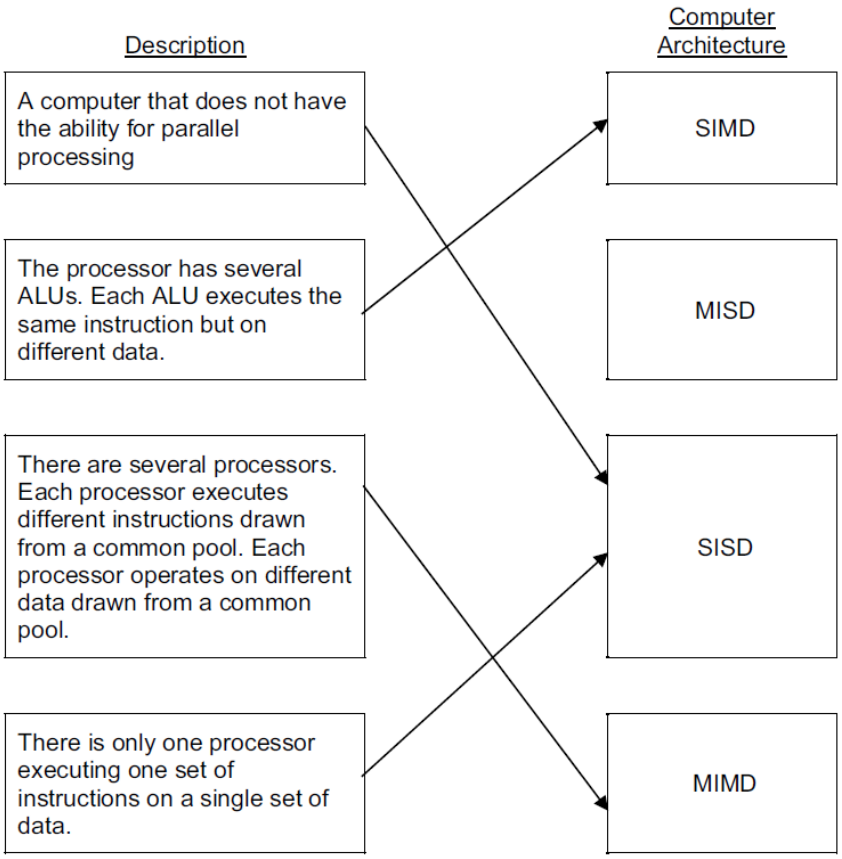
B in column 2, Row 1 (1 Mark)

Remainder completed (1 Mark)

b) ii) — **3**

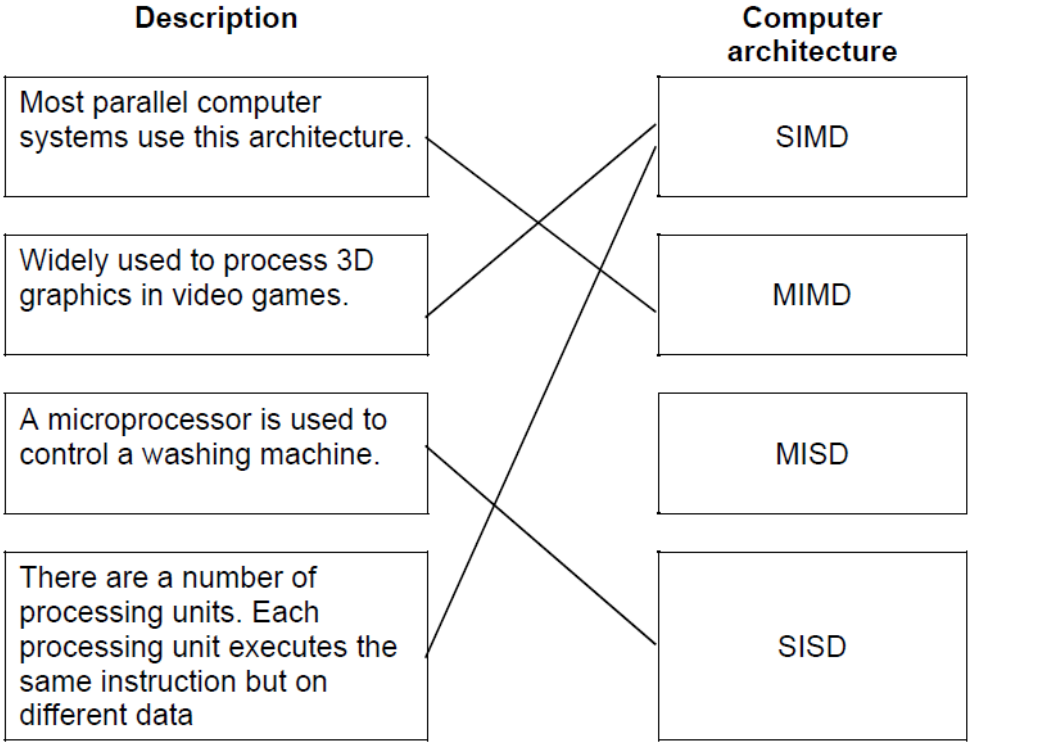
With pipelining no of cycles = 7
Without pipelining no of cycles = 3 * 5 = 15
No of cycles saved = 8

Question No. 2

<table>
<tr><td>a)</td><td>1 mark for correct arrow from each description
 
</td><td>4</td></tr>
<tr><td>b) i)</td><td>Massive: many/large number of processors // hundreds/thousands of processors</td><td>1</td></tr>
<tr><td>b) ii)</td><td>Parallel: to perform a set of coordinated computations in parallel/simultaneously</td><td>1</td></tr>
<tr><td>(c)</td><td>processors need to be able to communicate …
so that processed data can be transferred from one processor to another

suitable algorithm/program/software/design // appropriate programming language
which allows data to be processed by multiple processors simultaneously</td><td>4</td></tr>
</table>

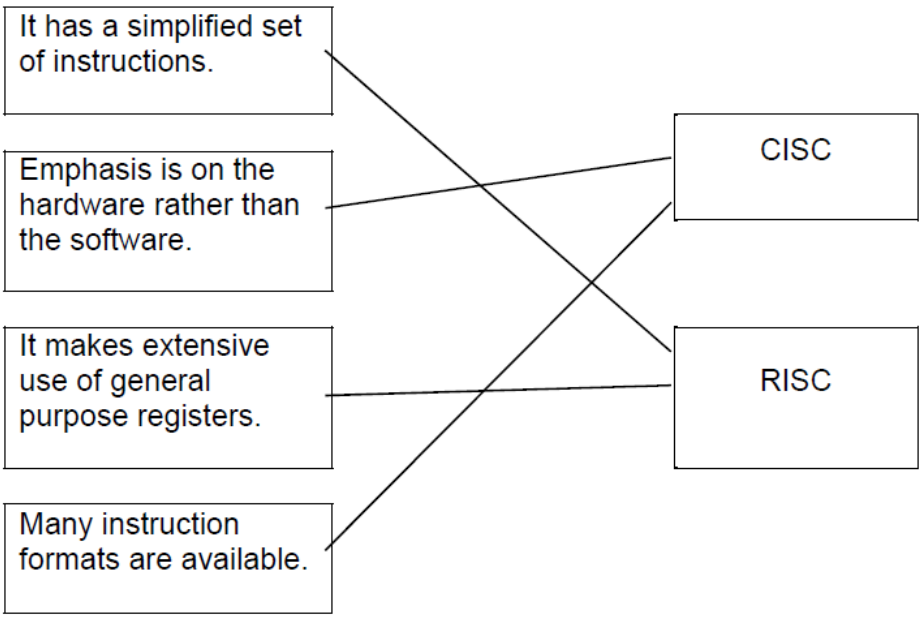
Question No. 3

<table>
<tr><td>a)</td><td> 

1 mark for each correct line</td><td>4</td></tr>
<tr><td>b)</td><td>Only one (separate) processor / not many separate processors (is not massively parallel) 1
Quad core computer system // processing units share the same bus 1
1 mark for each point, max 2</td><td>2</td></tr>
<tr><td>c)</td><td>Split into blocks of code ….
… that can be processed simultaneously …
… instead of sequentially
Each block is processed by a different processor
which allows each of the many processors to simultaneously process the different blocks of code independently
Requires both parallelism and co-ordination
1 mark for each point, max 2</td><td>2</td></tr>
<tr><td>d)</td><td>1 mark for identification of hardware issue, for example:
Communication between the different processors is the issue
1 mark for further explanation from:
Each processor needs a link to every other processor
Many processors require many of these links
Challenging topology</td><td>2</td></tr>
</table>

Question No. 4

a) — **3**



1 mark for each correct line

b) i) — **3**

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

b) ii) — **3**

	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)

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)	**3**
c) ii)	The third instruction is not dependent on the first two, therefore, instruction 2 and 3 need to be swapped	**3**

Question No. 5

a) — **4**

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

b) i) — **3**

1 mark per bullet point:

- Completing the As correctly
- B in column 2, row 1 no other Bs in row 1
- Remainder correctly completed

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

b) ii) — **3**

1 mark per bullet point:

- Correct number of cycles for pipelining 8
- Correct number of cycles without pipelining $4 \times 5 = 20$
- No of cycles saved $20 - 8 = 12$

c) — **4**

1 mark for each row

Statement	Architecture		
	SIMD	MIMD	SISD
Each processor executes a different instruction		✓	
There is only one processor			✓
Each processor executes the same instruction input using data available in the dedicated memory	✓		
Each processor typically has its own partition within a shared memory		✓	

Question No. 6

<table>
<tr><td>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>3</td></tr>
<tr><td>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>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>

Question No. 7

<table>
<tr>
<td>a)</td>
<td>1 mark for each correct term
<table>
<tr><th>Description</th><th>Term</th></tr>
<tr><td>• There are several processors.
• Each processor executes different sets of instructions on one set of data at the same time.</td><td>MISD</td></tr>
<tr><td>• The processor has several ALUs.
• Each ALU executes the same set of instructions on different sets of data at the same time.</td><td>SIMD</td></tr>
<tr><td>• There is only one processor.
• The processor executes one set of instructions on one set of data.</td><td>SISD</td></tr>
<tr><td>• There are several processors.
• Each processor executes a different set of instructions.
• Each processor operates on different sets of data.</td><td>MIMD</td></tr>
</table>
</td>
<td>4</td>
</tr>
<tr>
<td>b)</td>
<td>1 mark per bullet point to max 3

• A large number of processors
• Collaborative processing // coordinated simultaneous processing
• Network infrastructure
• Communicate using a message interface / by sending messages</td>
<td>3</td>
</tr>
</table>

Question No. 8

-	• RISC / reduced instruction set computer • CISC / complex instruction set computer • Pipelining	**3**

Question No. 9

a)	Max **three**, **one** mark for role, **one** mark for expansion OS1, OS2 and OS3 are guest operating systems … secondary to the one installed on the hardware OS4 is the host operating system … interacts directly with the machine hardware MyApp needs to run on all three guest operating systems … … with identical results	**3**
b)	Any **three** from Create/delete/manage virtual machine Translate instructions used by guest operating system to that required by host operating system Hardware emulation Protecting each virtual machine … so instances of MyApp can be tested together	**3**
c)	**One** mark for benefit and **one** mark for relevant explanation **One** mark for drawback **one** mark for relevant explanation For example: Benefit: multiple operating systems can exist simultaneously … allowing for testing using the same hardware only one set of hardware required ... reduces cost of producing the app // no need to set up more than one computer Drawback: execution of extra code … so performance is degraded // more time taken to execute the app // cannot make judgements about response time etc	**4**

Question No. 10

-	Software / a program Physical / different Guest Host	4

Question No. 11

a) i)	Examples: Create / delete virtual machine Existing hardware made available to guest OS // hardware emulation Ensures each virtual machine is protected from actions of another virtual machine	**2**
a) ii)	Guest operating system: An operating system running in a virtual machine // Controls virtual hardware // OS is being emulated Host operating system: The operating system that is actually controlling the physical hardware // the operating system for the physical machine// the OS running the VM software Guest OS is running under the Host OS software	**2**
b) i)	Examples: Trial/use alternative replacement operating system(s) … Test to identify possible problems Much easier to create VM with a new OS than create new computer system Trial/use alternative replacement web server software … Test to identify possible problems Easier to try alternative new software and new OS combinations To provide some additional service(s) Trial/test its use - description e.g. a print server General description point – to provide a safe environment during testing (which does not disrupt the web server service)	**4**
b) ii)	Examples: Using virtual machine means execution of extra code // emulation of some hardware … Non-VM installation may not perform in the same way Execution speed slower than non-VM system Problems in judging actual response times at time of maximum traffic needs fastest possible speed Particular hardware may be difficult to emulate	**2**

Question No. 12

a) i)	**A**: Guest (operating system) (1) **B**: Host (operating system) (1)	**2**
a) ii)	**One mark** for each valid point, **max 3** • Guest OS (A) handles request as if it were running on its own physical machine // guest OS (A) is not aware it is running on a virtual platform • Guest OS (A) handles the request as usual • I/O requests are translated by the virtual machine software • Into instructions executed by host OS (B) • Host OS (**B**) retrieves the data from the file • Host OS (**B**) passes the data to the virtual machine software • The virtual machine software passes the data to the guest OS (**A**) • Guest OS passes the data to the application	**3**
b) i)	**One mark** from: • Because software can be tried on different OS using same hardware • Because no need to purchase / request all sorts of different hardware • Easier to recover if software causes system crash • VM provides protection to other software / host OS from malfunctioning software	**1**
b) ii)	**Max 2 marks** per limitation, **max 2** limitations – **max 4 marks** Virtual machine may not be able to emulate some hardware … So that hardware cannot be tested using a virtual machine … By relevant example, e.g. developing hardware drivers Using virtual machine means execution of extra code // processing time increased … so cannot accurately test speed of real performance A virtual machine might not be as efficient … By relevant example, e.g. might not be able to access sufficient memory	**4**

Question No. 13

	One mark for each benefit to **max 2** from • Because software can be tried on different OS (using same hardware) • Because no need to purchase / request all sorts of different hardware • Easier to recover if software causes system crash • VM provides protection to other software / host OS from malfunctioning software **One mark** for each limitation to **max 2** from • Virtual machines may not be able to emulate some hardware • Virtual machines cannot directly access some hardware • Using virtual machine means execution of extra code // processing time increased • A virtual machine might not be as efficient // performance degrades • Use of a virtual machine increases the maintenance overheads	**4**