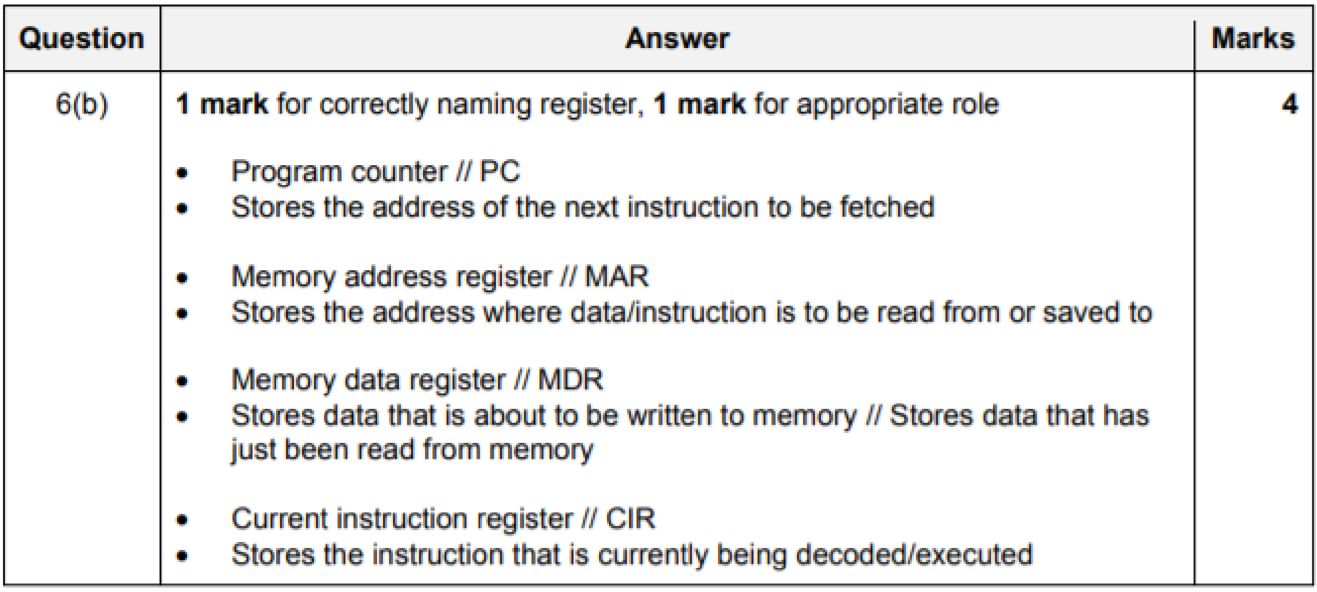
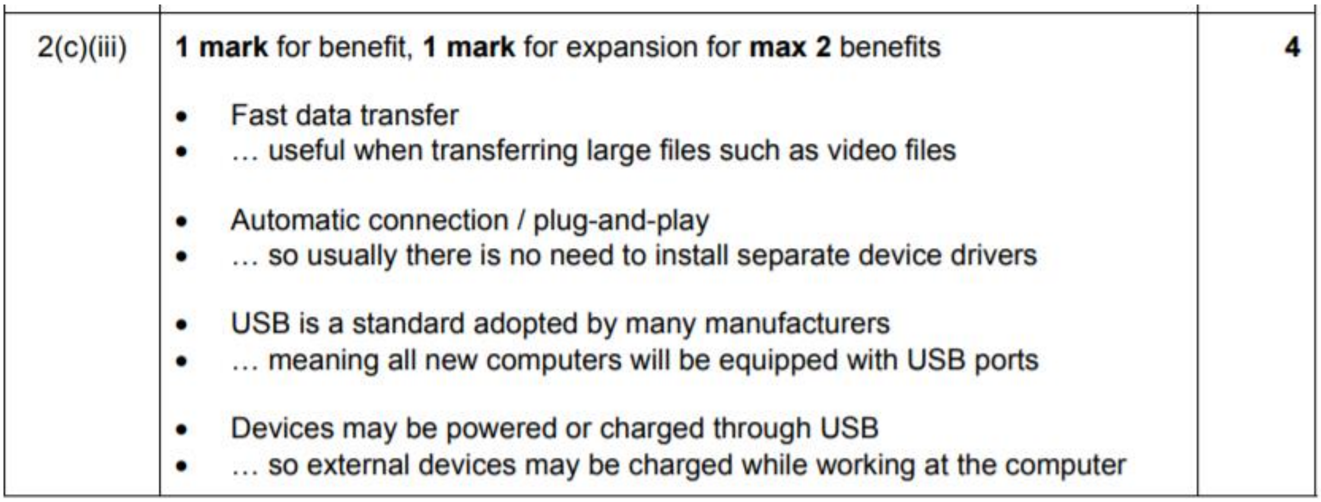
ANSWERS PROCESSOR FUNDAMENTALS

Answer 1



Question	Answer	Marks
6(b)	**1 mark** for correctly naming register, **1 mark** for appropriate role • Program counter // PC • Stores the address of the next instruction to be fetched • Memory address register // MAR • Stores the address where data/instruction is to be read from or saved to • Memory data register // MDR • Stores data that is about to be written to memory // Stores data that has just been read from memory • Current instruction register // CIR • Stores the instruction that is currently being decoded/executed	**4**

Answer 2



2(c)(iii)	**1 mark** for benefit, **1 mark** for expansion for **max 2** benefits • Fast data transfer • … useful when transferring large files such as video files • Automatic connection / plug-and-play • … so usually there is no need to install separate device drivers • USB is a standard adopted by many manufacturers • … meaning all new computers will be equipped with USB ports • Devices may be powered or charged through USB • … so external devices may be charged while working at the computer	**4**

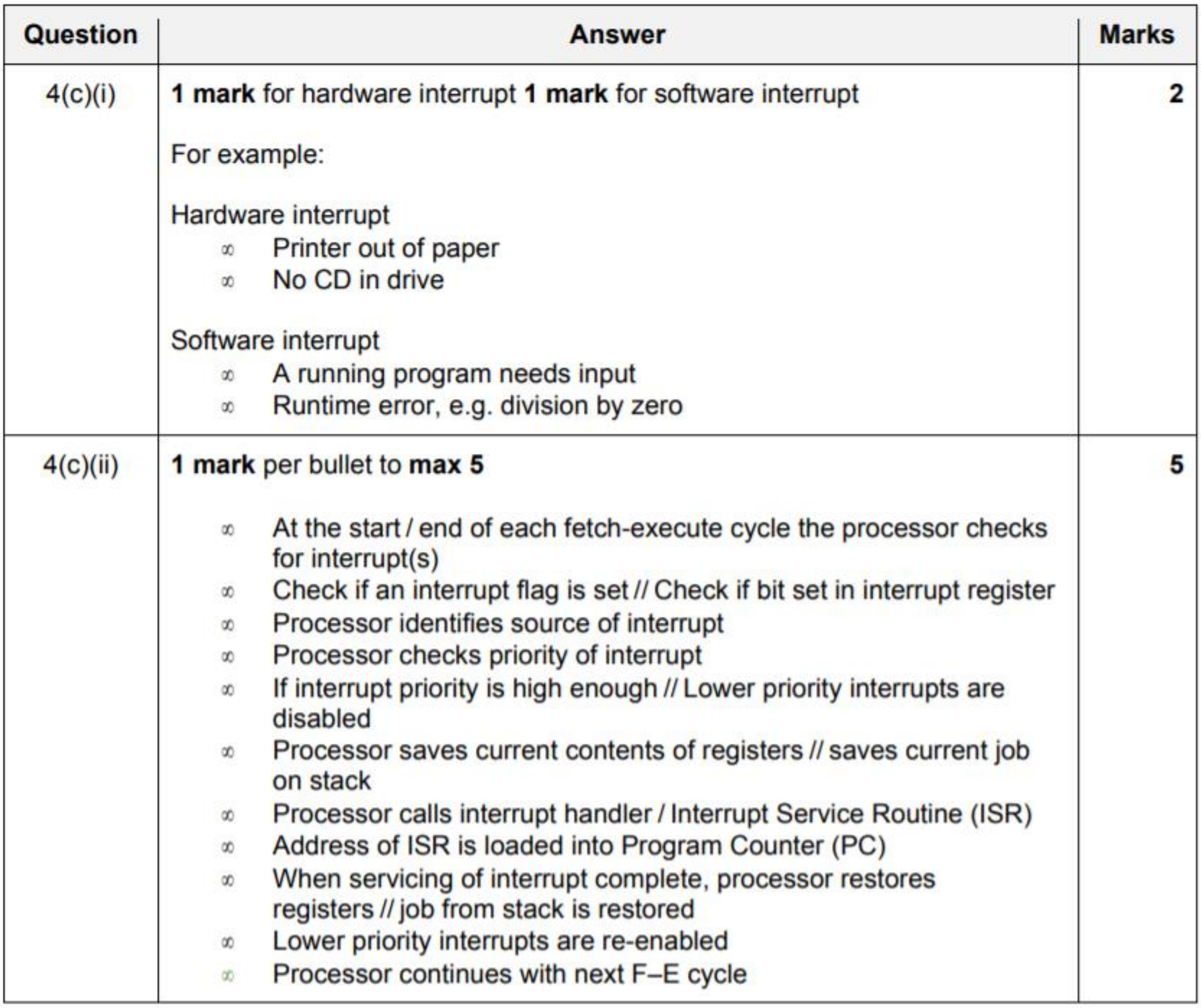
Answer 3

2(c)(i)	**1 mark** • To provide the connection to peripheral devices • To provide an interface between the computer and other devices	**1**
2(c)(ii)	**1 mark** • USB • HDMI • SCSI • Firewire • Ethernet • Any other acceptable port	**1**

Answer 4

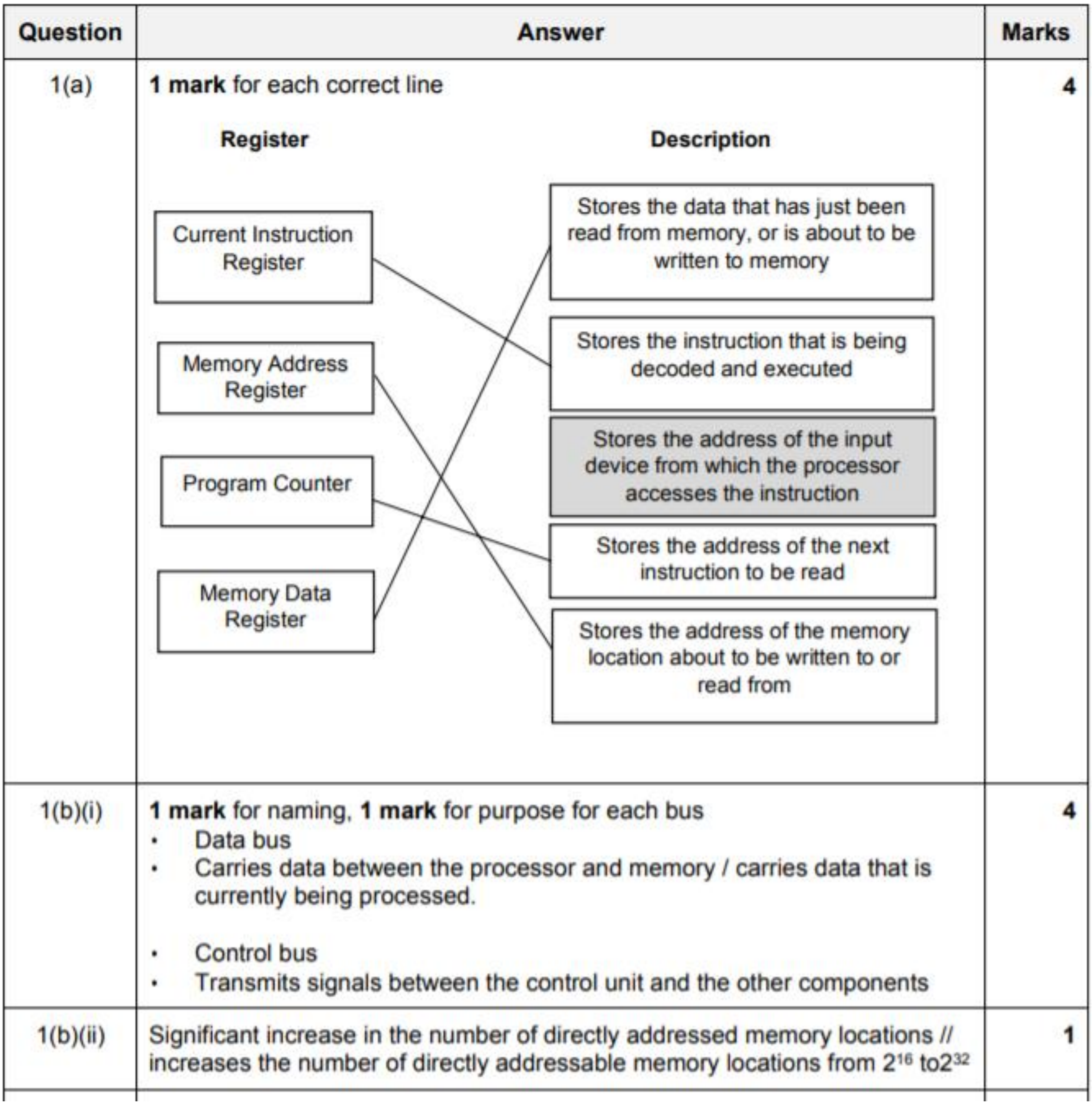
Question	Answer	Marks
3(a)	**1 mark** for each error **and** correction • Line 02 should be +1 not −1 // `PC ← [PC] + 1` • Line 03 should be double brackets around `MAR` // `MDR ← [[MAR]]` • Line 04 should be `MDR` not `MAR` // `CIR ← [MDR]`	**3**
3(b)	**1 mark** for each group to **max. 2** • Data movement • Arithmetic operations • (Unconditional and conditional) jump instructions • Compare instructions • Modes of addressing	**2**
3(c)	**1 mark** per bullet • Storing 0 in 401 (line 51) • Loading memory location 300, value 2 to ACC (line 52) • Adding 64 to ACC to give 66 (line 55) • Outputting B (line 56) • Load 0 (line 57), increment ACC (line 58) and store 1 in 401 (line 59) • Incrementing IX (line 60) • Loading 5 (line 52), adding 64 (line 55), outputting E (line 56) loading 1 (line 57), incrementing ACC (line 58), storing 2 in 401 (line 59) and incrementing IX (line 60) • Load 0 (line 52) <u>and end</u>	**8**

Answer 5



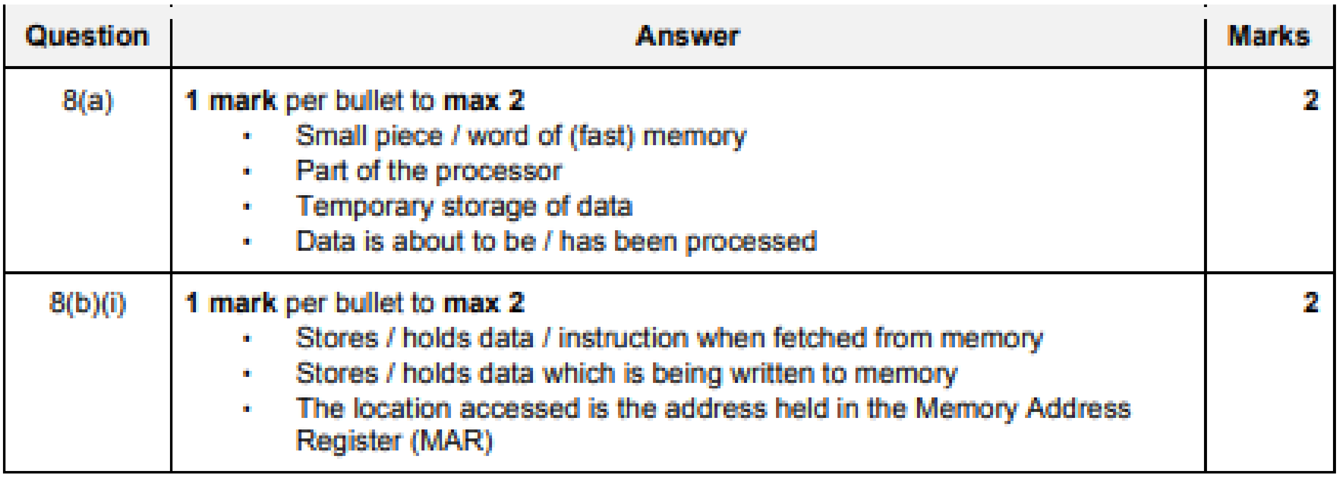
Question	Answer	Marks
4(c)(i)	**1 mark** for hardware interrupt **1 mark** for software interrupt For example: Hardware interrupt ∞ Printer out of paper ∞ No CD in drive Software interrupt ∞ A running program needs input ∞ Runtime error, e.g. division by zero	**2**
4(c)(ii)	**1 mark** per bullet to **max 5** ∞ At the start / end of each fetch-execute cycle the processor checks for interrupt(s) ∞ Check if an interrupt flag is set // Check if bit set in interrupt register ∞ Processor identifies source of interrupt ∞ Processor checks priority of interrupt ∞ If interrupt priority is high enough // Lower priority interrupts are disabled ∞ Processor saves current contents of registers // saves current job on stack ∞ Processor calls interrupt handler / Interrupt Service Routine (ISR) ∞ Address of ISR is loaded into Program Counter (PC) ∞ When servicing of interrupt complete, processor restores registers // job from stack is restored ∞ Lower priority interrupts are re-enabled ∞ Processor continues with next F–E cycle	**5**

Answer 6

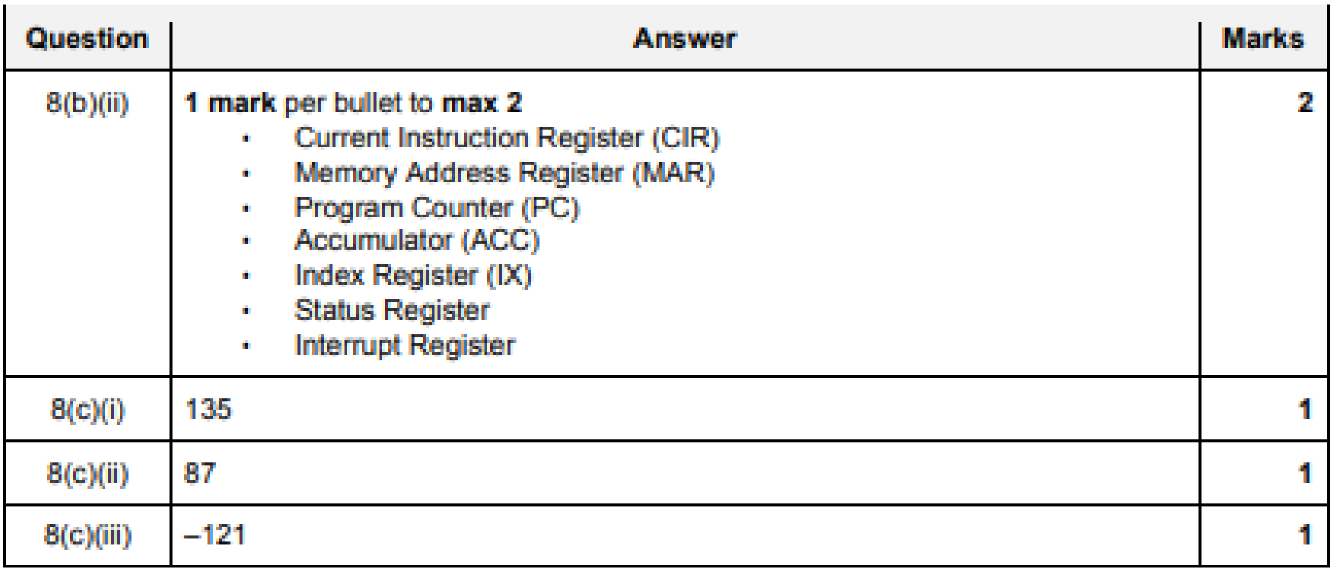


Question	Answer	Marks
1(a)	**1 mark** for each correct line **Register** Current Instruction Register Memory Address Register Program Counter Memory Data Register **Description** Stores the data that has just been read from memory, or is about to be written to memory Stores the instruction that is being decoded and executed Stores the address of the input device from which the processor accesses the instruction Stores the address of the next instruction to be read Stores the address of the memory location about to be written to or read from	**4**
1(b)(i)	**1 mark** for naming, **1 mark** for purpose for each bus • Data bus • Carries data between the processor and memory / carries data that is currently being processed. • Control bus • Transmits signals between the control unit and the other components	**4**
1(b)(ii)	Significant increase in the number of directly addressed memory locations // increases the number of directly addressable memory locations from 2^{16} to2^{32}	**1**

Answer 7

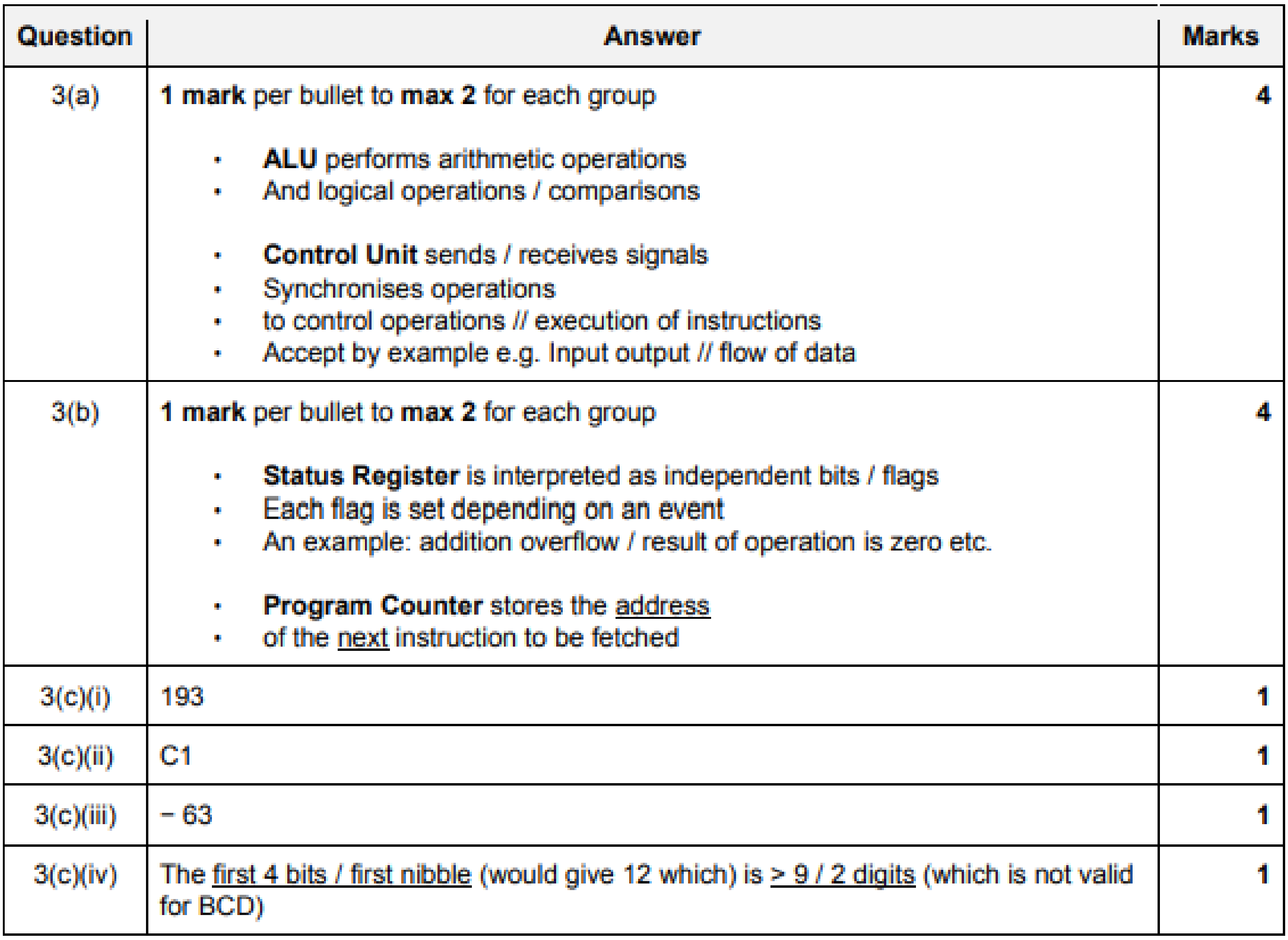


Question	Answer	Marks
8(a)	**1 mark** per bullet to **max 2** • Small piece / word of (fast) memory • Part of the processor • Temporary storage of data • Data is about to be / has been processed	**2**
8(b)(i)	**1 mark** per bullet to **max 2** • Stores / holds data / instruction when fetched from memory • Stores / holds data which is being written to memory • The location accessed is the address held in the Memory Address Register (MAR)	**2**



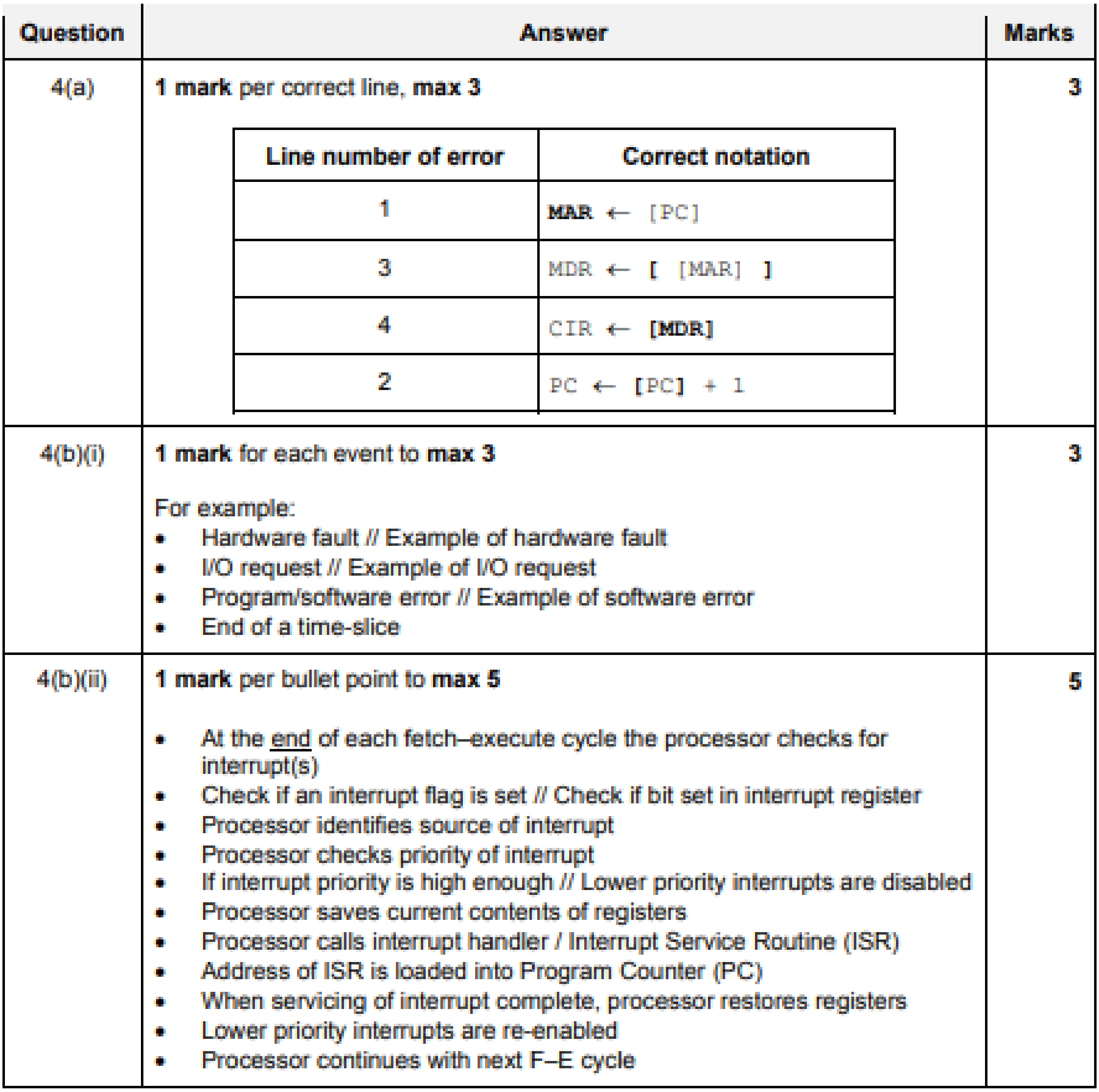
Question	Answer	Marks
8(b)(ii)	**1 mark** per bullet to **max 2** • Current Instruction Register (CIR) • Memory Address Register (MAR) • Program Counter (PC) • Accumulator (ACC) • Index Register (IX) • Status Register • Interrupt Register	**2**
8(c)(i)	135	**1**
8(c)(ii)	87	**1**
8(c)(iii)	–121	**1**

Answer 8



Question	Answer	Marks
3(a)	**1 mark** per bullet to **max 2** for each group • **ALU** performs arithmetic operations • And logical operations / comparisons • **Control Unit** sends / receives signals • Synchronises operations • to control operations // execution of instructions • Accept by example e.g. Input output // flow of data	**4**
3(b)	**1 mark** per bullet to **max 2** for each group • **Status Register** is interpreted as independent bits / flags • Each flag is set depending on an event • An example: addition overflow / result of operation is zero etc. • **Program Counter** stores the address • of the next instruction to be fetched	**4**
3(c)(i)	193	**1**
3(c)(ii)	C1	**1**
3(c)(iii)	– 63	**1**
3(c)(iv)	The first 4 bits / first nibble (would give 12 which) is > 9 / 2 digits (which is not valid for BCD)	**1**

Answer 9



<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4(a)</td><td>

1 mark per correct line, **max 3**

<table>
<tr><th>Line number of error</th><th>Correct notation</th></tr>
<tr><td>1</td><td>MAR ← [PC]</td></tr>
<tr><td>3</td><td>MDR ← [[MAR]]</td></tr>
<tr><td>4</td><td>CIR ← [MDR]</td></tr>
<tr><td>2</td><td>PC ← [PC] + 1</td></tr>
</table>

</td><td>3</td></tr>
<tr><td>4(b)(i)</td><td>

1 mark for each event to **max 3**

For example:

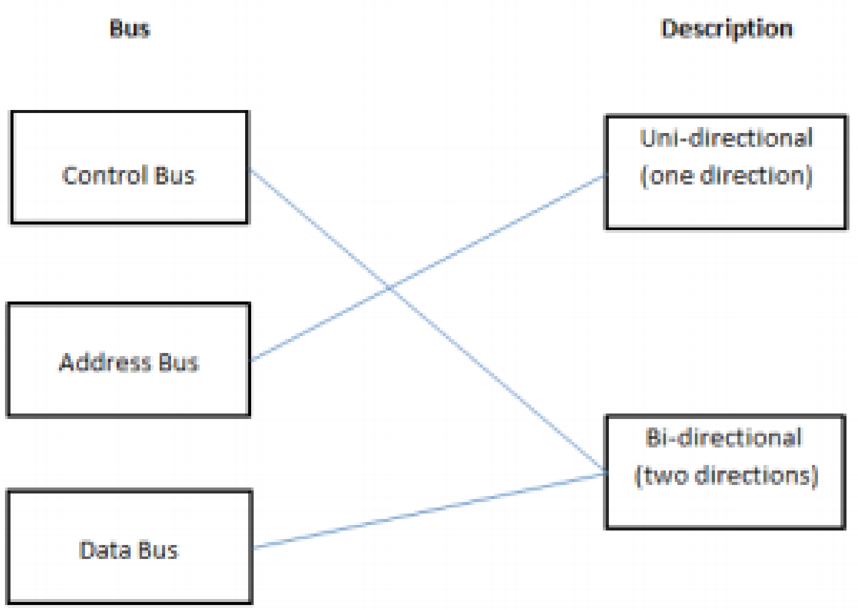
- Hardware fault // Example of hardware fault
- I/O request // Example of I/O request
- Program/software error // Example of software error
- End of a time-slice

</td><td>3</td></tr>
<tr><td>4(b)(ii)</td><td>

1 mark per bullet point to **max 5**

- At the end of each fetch–execute cycle the processor checks for interrupt(s)
- Check if an interrupt flag is set // Check if bit set in interrupt register
- Processor identifies source of interrupt
- Processor checks priority of interrupt
- If interrupt priority is high enough // Lower priority interrupts are disabled
- Processor saves current contents of registers
- Processor calls interrupt handler / Interrupt Service Routine (ISR)
- Address of ISR is loaded into Program Counter (PC)
- When servicing of interrupt complete, processor restores registers
- Lower priority interrupts are re-enabled
- Processor continues with next F–E cycle

</td><td>5</td></tr>
</table>

Question	Answer	Marks
4(c)	**1 mark** for **1 correct** connection **2 marks** for all **3 correct** connections  Bus Description Control Bus Address Bus Data Bus Uni-directional (one direction) Bi-directional (two directions)	**2**

Answer 10

Question	Answer	Marks

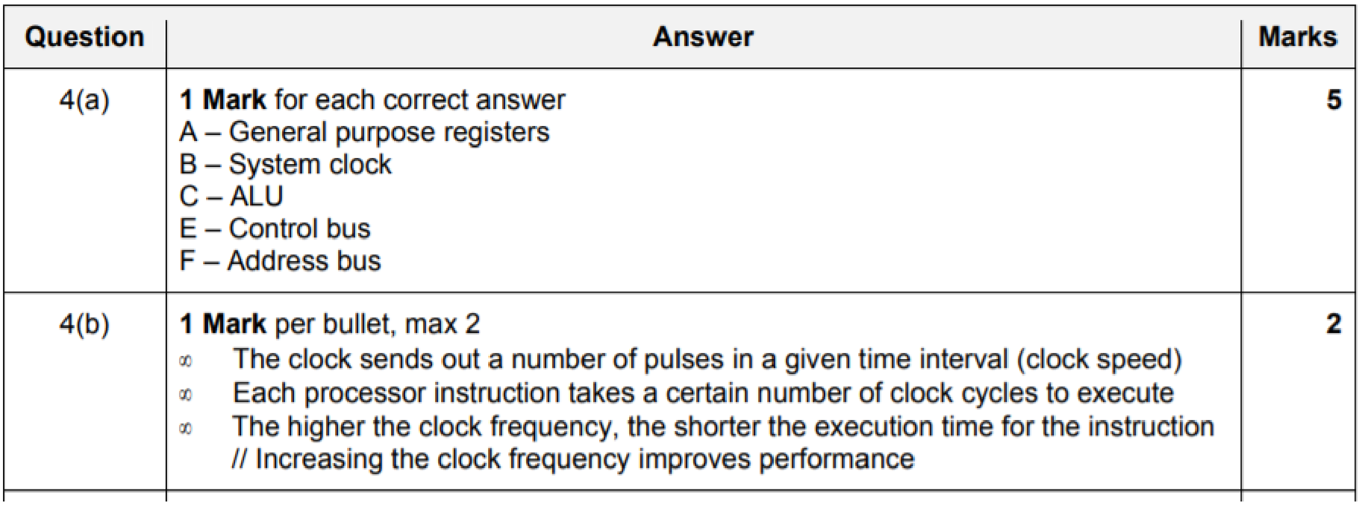
6(a)(i)	MAR ← [**PC**]
	PC ← PC + 1
	MDR ← [[MAR]]
	CIR ← [MDR]
		3

6(a)(ii)	**1 mark** from: • The contents of the MAR is an address, it is the contents of that address which is transferred to MDR • The contents of the address pointed to by the MAR is transferred to the MDR	**1**
6(b)	**1 mark** per bullet point to **max 2** • A signal from a source / device • Telling the processor its attention is needed	**2**

Answer 11

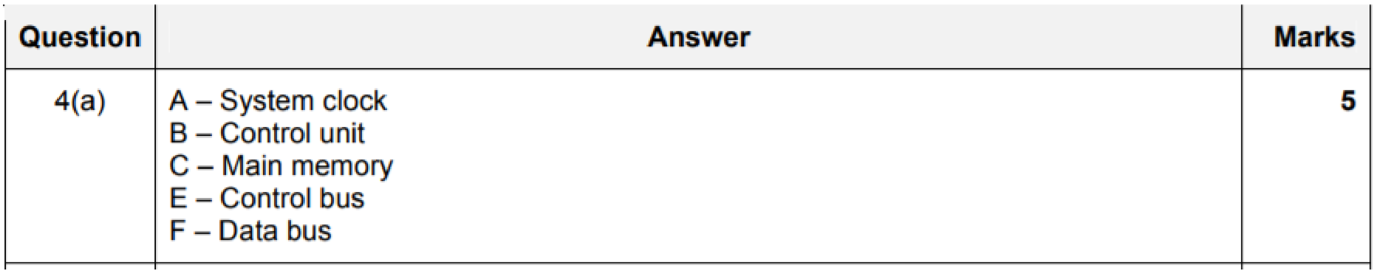
Question	Answer	Marks
4(a)	A – System clock B – Control unit C – Main memory E – Control bus F – Data bus	**5**

Answer 12



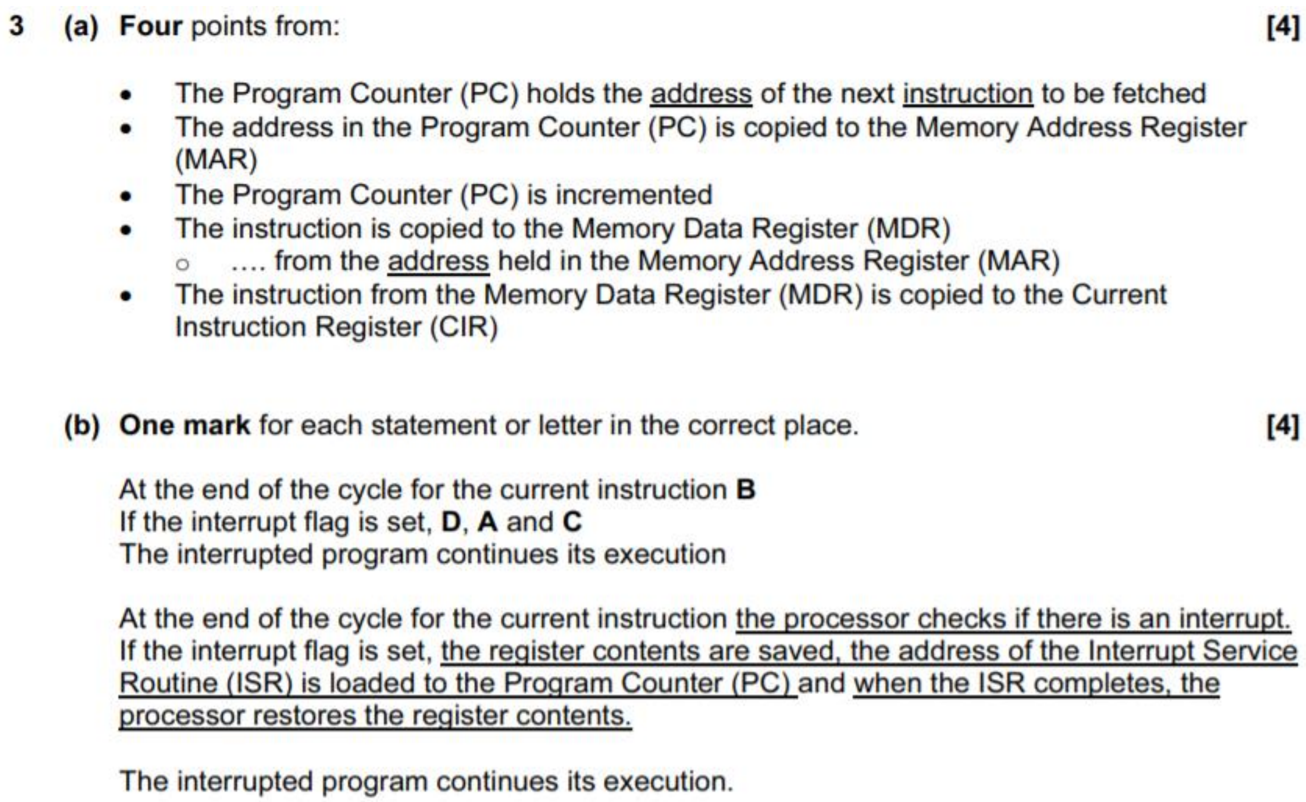
Question	Answer	Marks
4(a)	**1 Mark** for each correct answer A – General purpose registers B – System clock C – ALU E – Control bus F – Address bus	**5**
4(b)	**1 Mark** per bullet, max 2 ∞ The clock sends out a number of pulses in a given time interval (clock speed) ∞ Each processor instruction takes a certain number of clock cycles to execute ∞ The higher the clock frequency, the shorter the execution time for the instruction // Increasing the clock frequency improves performance	**2**

Answer 13



Question	Answer	Marks
4(a)	A – System clock B – Control unit C – Main memory E – Control bus F – Data bus	**5**

Answer 14



3 (a) Four points from: **[4]**

- The Program Counter (PC) holds the address of the next instruction to be fetched
- The address in the Program Counter (PC) is copied to the Memory Address Register (MAR)
- The Program Counter (PC) is incremented
- The instruction is copied to the Memory Data Register (MDR)
  - …. from the address held in the Memory Address Register (MAR)
- The instruction from the Memory Data Register (MDR) is copied to the Current Instruction Register (CIR)

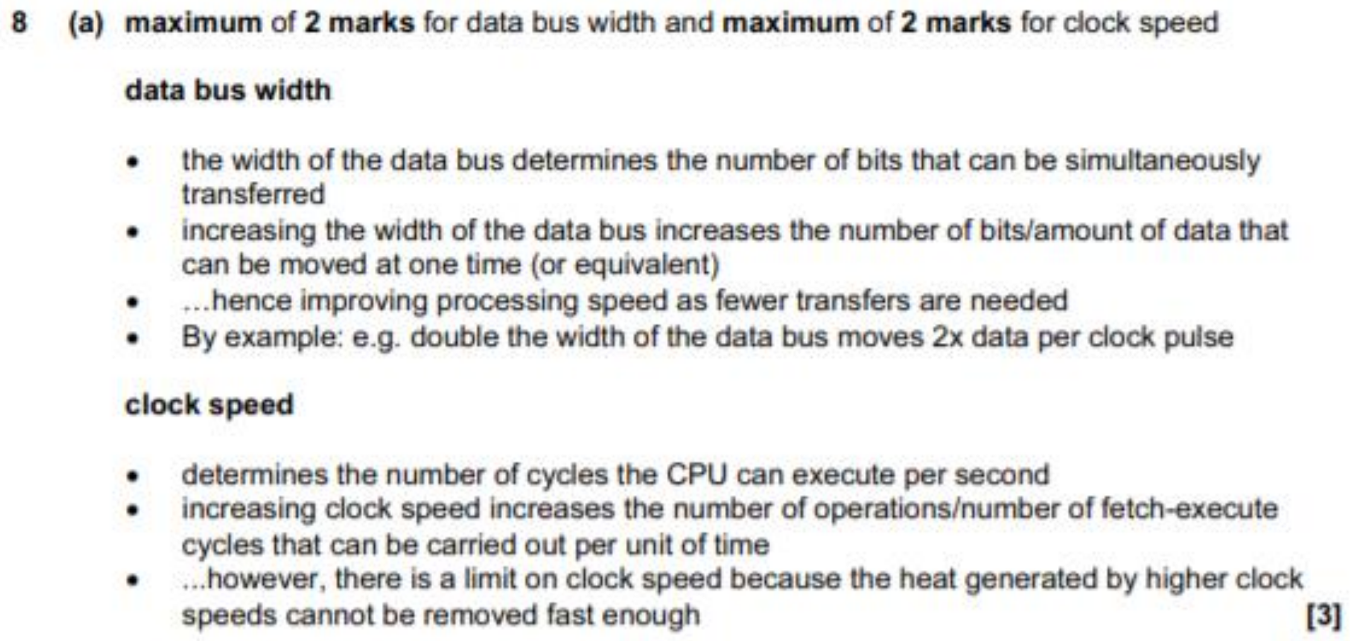
(b) One mark for each statement or letter in the correct place. **[4]**

At the end of the cycle for the current instruction **B**
If the interrupt flag is set, **D**, **A** and **C**
The interrupted program continues its execution

At the end of the cycle for the current instruction the processor checks if there is an interrupt.
If the interrupt flag is set, the register contents are saved, the address of the Interrupt Service Routine (ISR) is loaded to the Program Counter (PC) and when the ISR completes, the processor restores the register contents.

The interrupted program continues its execution.

Answer 15



8 (a) **maximum** of **2 marks** for data bus width and **maximum** of **2 marks** for clock speed

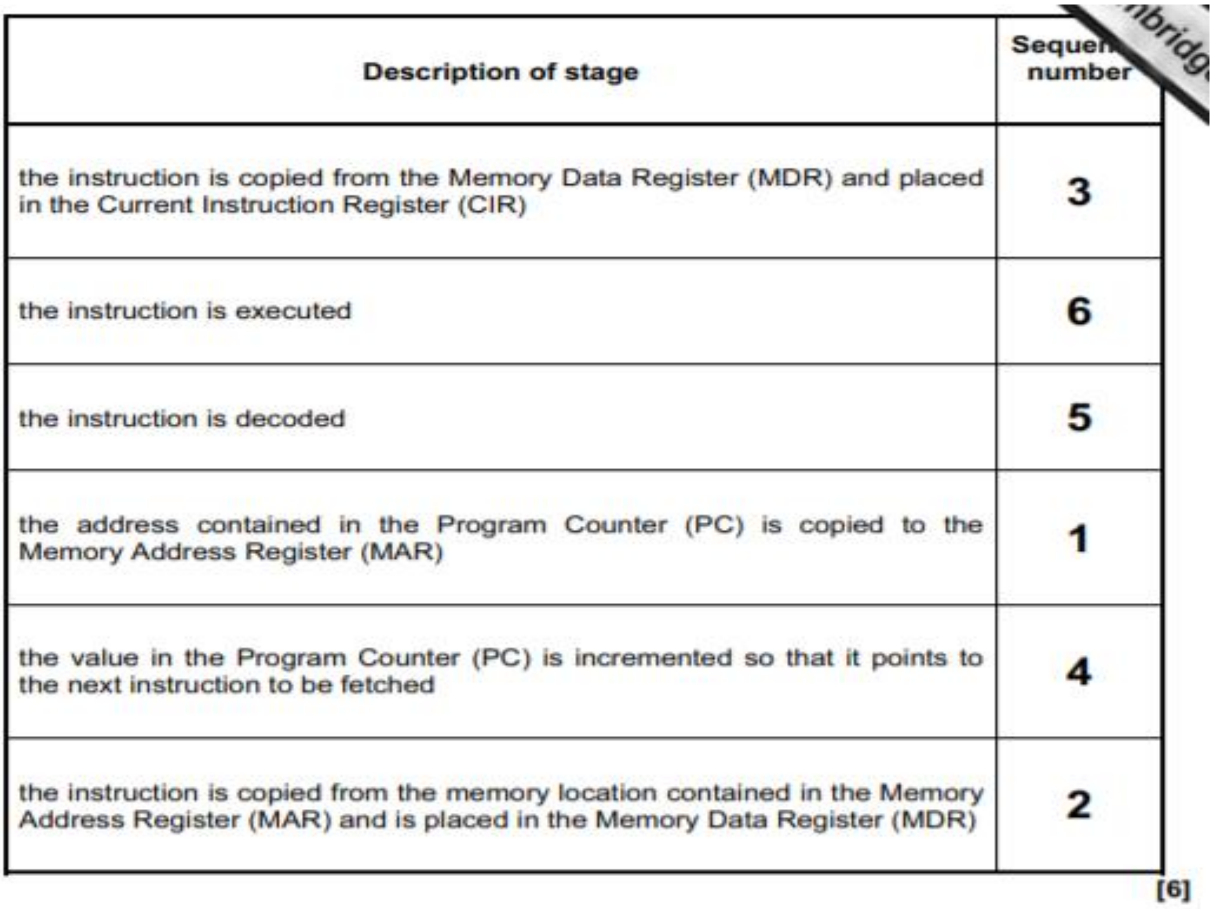
data bus width

- the width of the data bus determines the number of bits that can be simultaneously transferred
- increasing the width of the data bus increases the number of bits/amount of data that can be moved at one time (or equivalent)
- …hence improving processing speed as fewer transfers are needed
- By example: e.g. double the width of the data bus moves 2x data per clock pulse

clock speed

- determines the number of cycles the CPU can execute per second
- increasing clock speed increases the number of operations/number of fetch-execute cycles that can be carried out per unit of time
- ...however, there is a limit on clock speed because the heat generated by higher clock speeds cannot be removed fast enough **[3]**

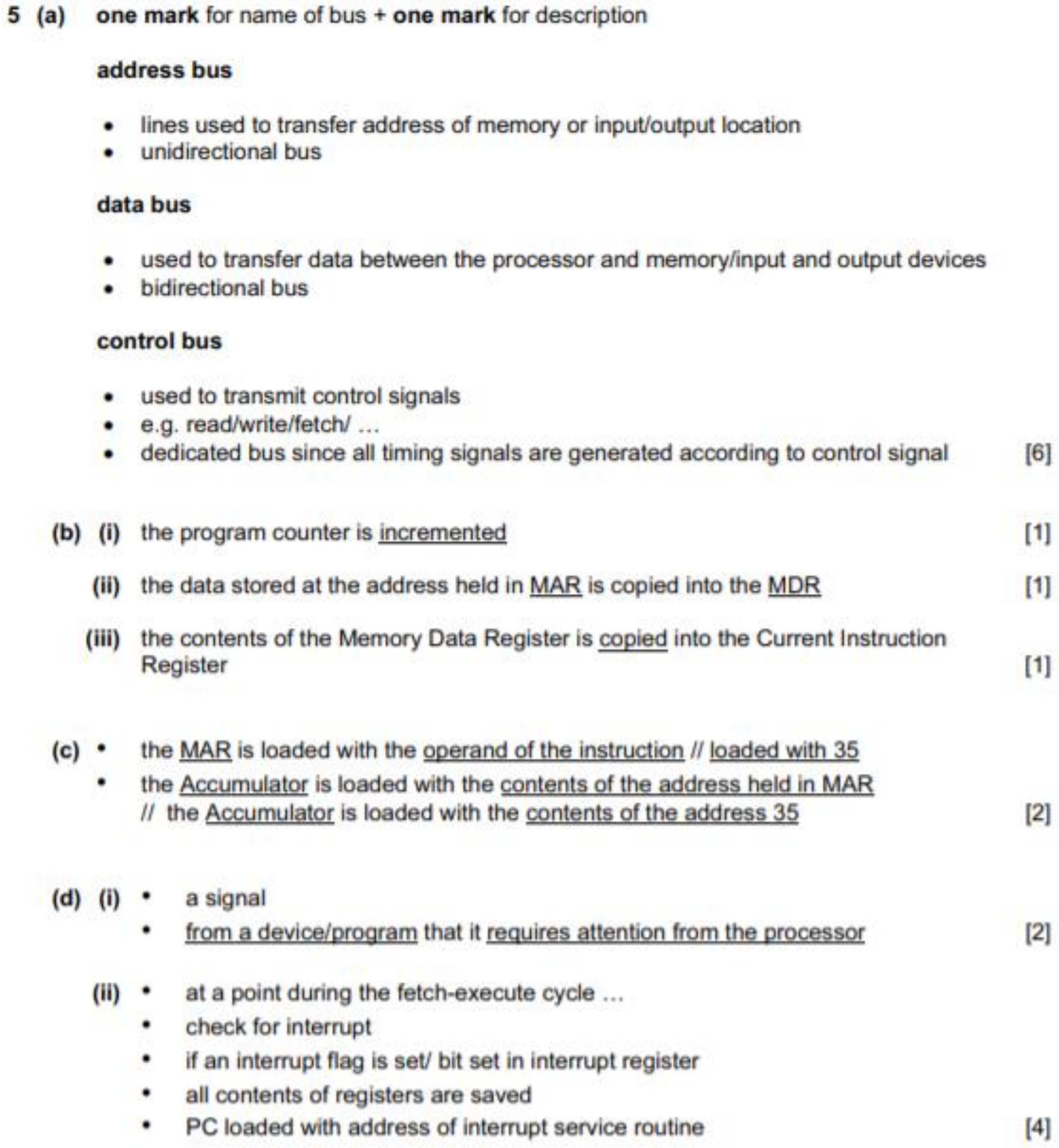
(c)



Description of stage	**Seque number**
the instruction is copied from the Memory Data Register (MDR) and placed in the Current Instruction Register (CIR)	**3**
the instruction is executed	**6**
the instruction is decoded	**5**
the address contained in the Program Counter (PC) is copied to the Memory Address Register (MAR)	**1**
the value in the Program Counter (PC) is incremented so that it points to the next instruction to be fetched	**4**
the instruction is copied from the memory location contained in the Memory Address Register (MAR) and is placed in the Memory Data Register (MDR)	**2**

[6]

Answer 16



5 (a) **one mark** for name of bus + **one mark** for description

address bus

- lines used to transfer address of memory or input/output location
- unidirectional bus

data bus

- used to transfer data between the processor and memory/input and output devices
- bidirectional bus

control bus

- used to transmit control signals
- e.g. read/write/fetch/ …
- dedicated bus since all timing signals are generated according to control signal [6]

(b) (i) the program counter is incremented [1]

(ii) the data stored at the address held in MAR is copied into the MDR [1]

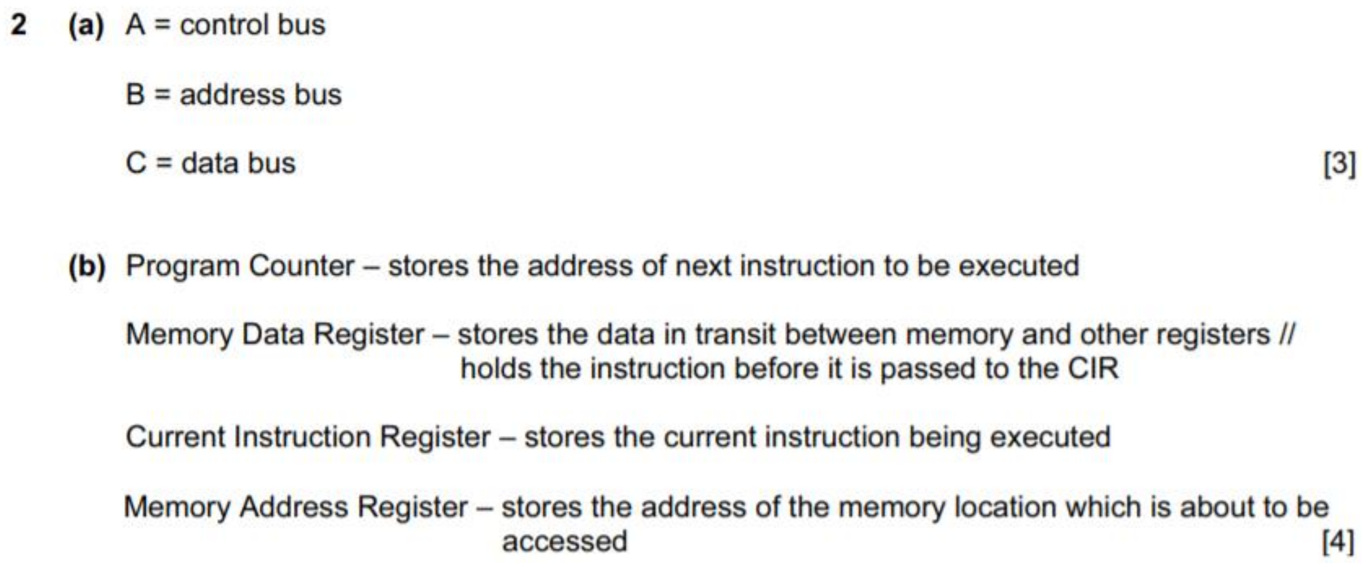
(iii) the contents of the Memory Data Register is copied into the Current Instruction Register [1]

(c)
- the MAR is loaded with the operand of the instruction // loaded with 35
- the Accumulator is loaded with the contents of the address held in MAR // the Accumulator is loaded with the contents of the address 35 [2]

(d) (i)
- a signal
- from a device/program that it requires attention from the processor [2]

(ii)
- at a point during the fetch-execute cycle …
- check for interrupt
- if an interrupt flag is set/ bit set in interrupt register
- all contents of registers are saved
- PC loaded with address of interrupt service routine [4]

Answer 17



2 **(a)** A = control bus

B = address bus

C = data bus [3]

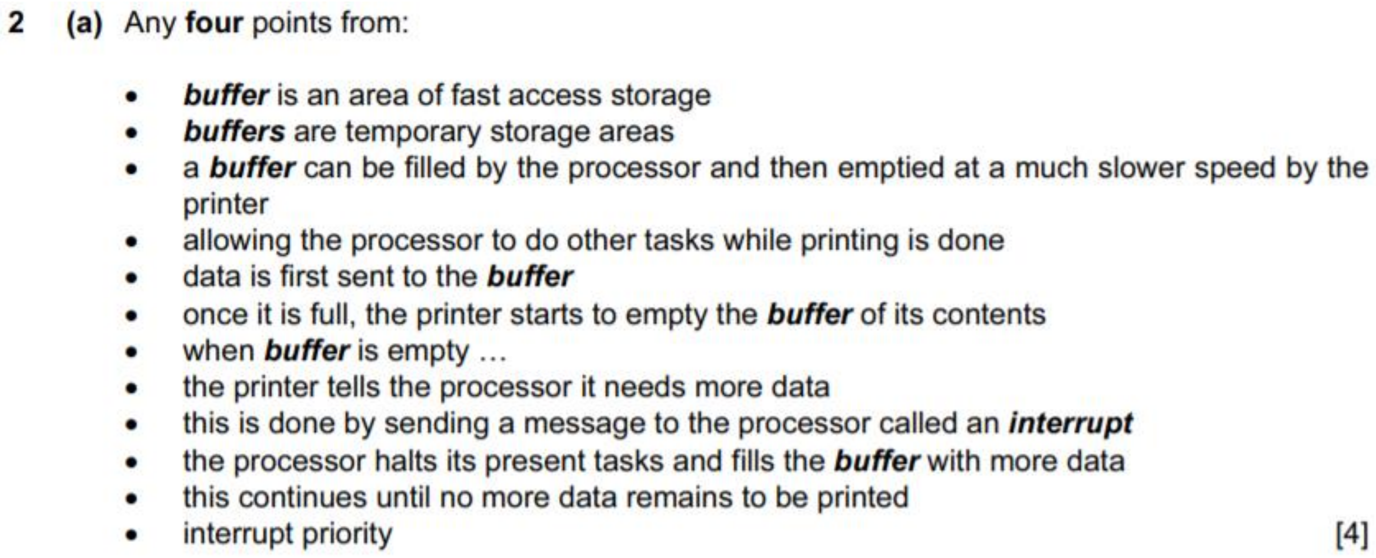
(b) Program Counter – stores the address of next instruction to be executed

Memory Data Register – stores the data in transit between memory and other registers // holds the instruction before it is passed to the CIR

Current Instruction Register – stores the current instruction being executed

Memory Address Register – stores the address of the memory location which is about to be accessed [4]

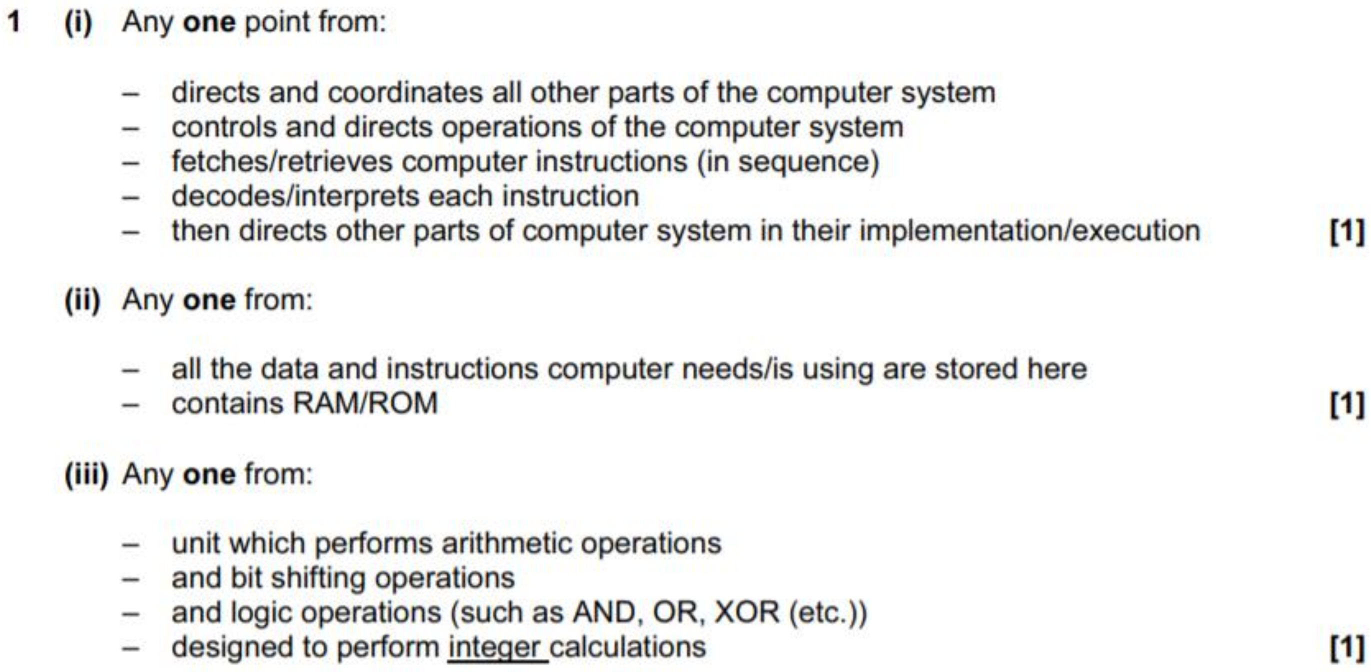
Answer 18



2 **(a)** Any **four** points from:

- ***buffer*** is an area of fast access storage
- ***buffers*** are temporary storage areas
- a ***buffer*** can be filled by the processor and then emptied at a much slower speed by the printer
- allowing the processor to do other tasks while printing is done
- data is first sent to the ***buffer***
- once it is full, the printer starts to empty the ***buffer*** of its contents
- when ***buffer*** is empty …
- the printer tells the processor it needs more data
- this is done by sending a message to the processor called an ***interrupt***
- the processor halts its present tasks and fills the ***buffer*** with more data
- this continues until no more data remains to be printed
- interrupt priority [4]

Answer 19



1 (i) Any **one** point from:

- directs and coordinates all other parts of the computer system
- controls and directs operations of the computer system
- fetches/retrieves computer instructions (in sequence)
- decodes/interprets each instruction
- then directs other parts of computer system in their implementation/execution **[1]**

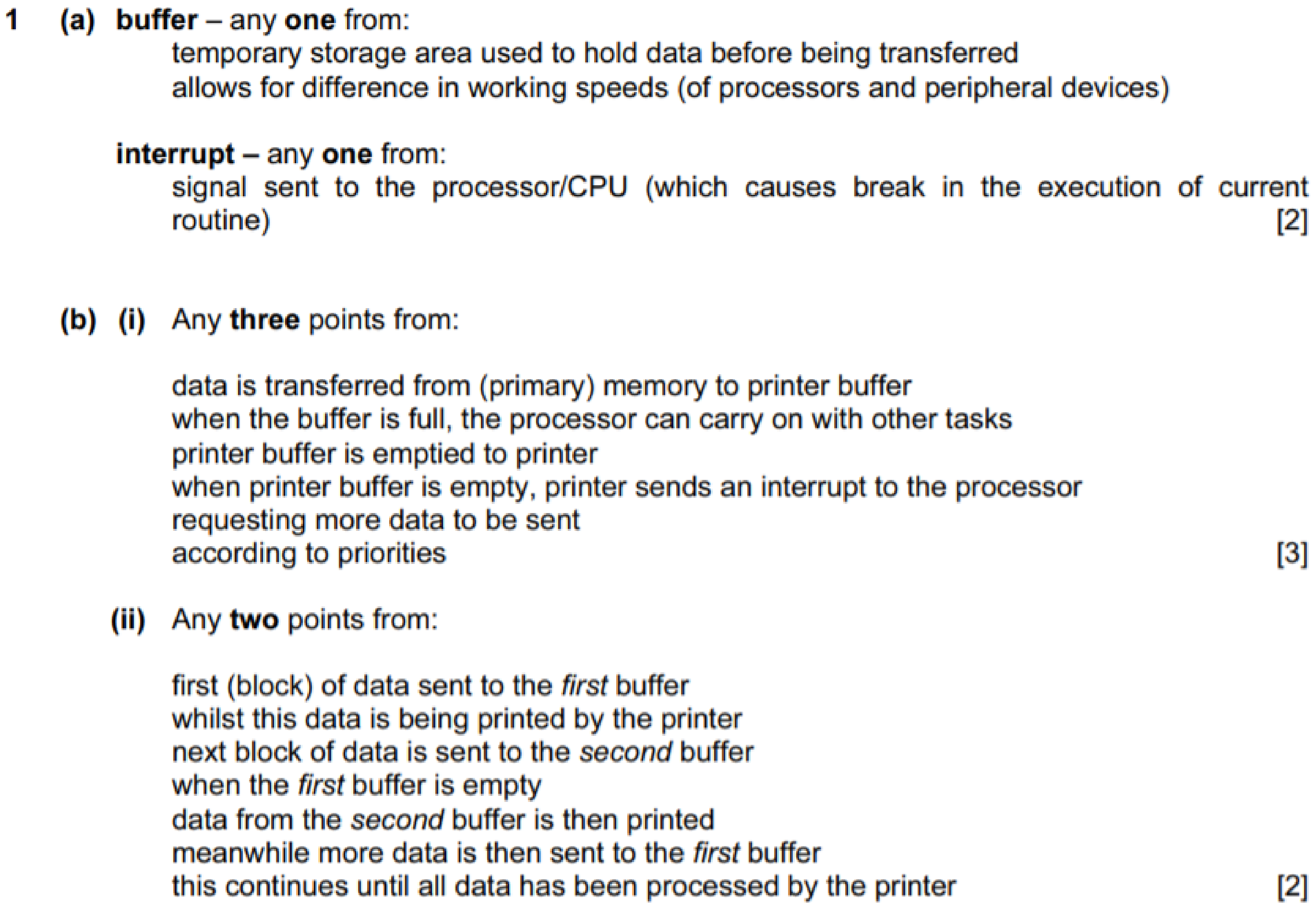
(ii) Any **one** from:

- all the data and instructions computer needs/is using are stored here
- contains RAM/ROM **[1]**

(iii) Any **one** from:

- unit which performs arithmetic operations
- and bit shifting operations
- and logic operations (such as AND, OR, XOR (etc.))
- designed to perform integer calculations **[1]**

Answer 20



1 (a) **buffer** – any **one** from:

temporary storage area used to hold data before being transferred
allows for difference in working speeds (of processors and peripheral devices)

interrupt – any **one** from:

signal sent to the processor/CPU (which causes break in the execution of current routine) [2]

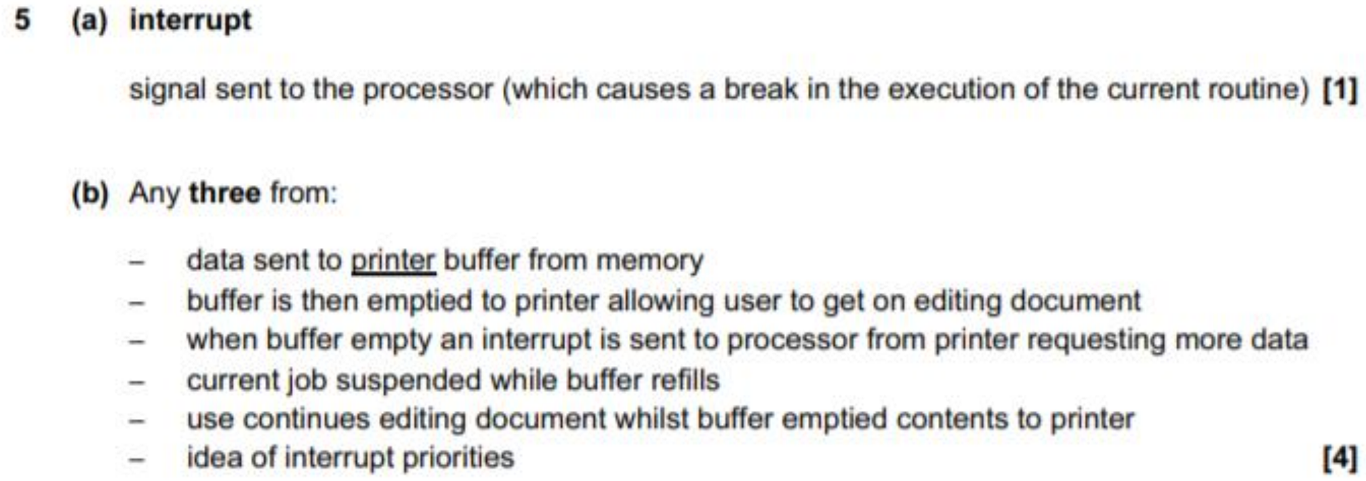
(b) (i) Any **three** points from:

data is transferred from (primary) memory to printer buffer
when the buffer is full, the processor can carry on with other tasks
printer buffer is emptied to printer
when printer buffer is empty, printer sends an interrupt to the processor
requesting more data to be sent
according to priorities [3]

(ii) Any **two** points from:

first (block) of data sent to the *first* buffer
whilst this data is being printed by the printer
next block of data is sent to the *second* buffer
when the *first* buffer is empty
data from the *second* buffer is then printed
meanwhile more data is then sent to the *first* buffer
this continues until all data has been processed by the printer [2]

Answer 21



5 (a) **interrupt**

signal sent to the processor (which causes a break in the execution of the current routine) **[1]**

(b) Any **three** from:

- data sent to printer buffer from memory
- buffer is then emptied to printer allowing user to get on editing document
- when buffer empty an interrupt is sent to processor from printer requesting more data
- current job suspended while buffer refills
- use continues editing document whilst buffer emptied contents to printer
- idea of interrupt priorities **[4]**