

# Central Processing Unit

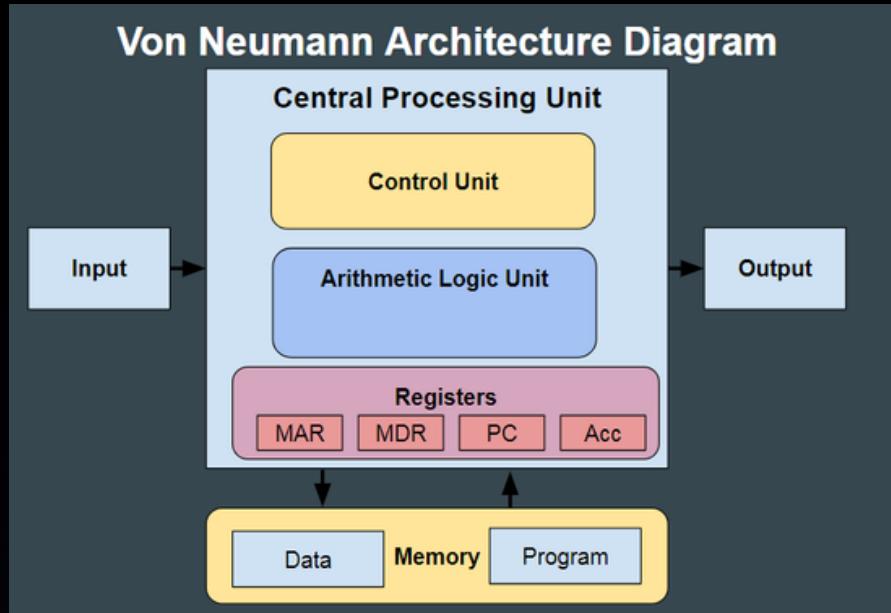


Papers Dock

---

COMPUTER SCIENCE 9618 PAPER I

# Central Processing Unit



The Von Neumann Architecture consists of single, shared memory for program and data

what is meant by the stored program concept in the Von Neumann model of a computer system?

Instructions and data are stored in the same memory space / in main memory.

What are the features of Von Neuman Architecture?

- a central processing unit (CPU or processor)
- a processor able to access the memory directly
- computer memories that could store programs as well as data
- stored programs made up of instructions that could be executed in sequential order.

# Purpose Of Control Unit

- to coordinate / synchronize the actions of other components in the CPU
- to send / receive control signals along the control bus
- to manage the execution of instructions (in sequence)
- to control the communication between the components of the CPU
- Fetches Computer instructions
- Decodes each instructions
- Executes these instructions

# Purpose Of Arithmetic Logic Unit (ALU)

- A unit which performs arithmetic operations (+, -, \*, /)
- and bit shifting operation
- logic operation (AND, OR, XOR)
- Designed to perform integer calculation

**OPERAND** : Part of the computer instruction which specifies what data is to be processed

**OPCODE** : Is the instruction that is executed by the CPU

# Purpose Of Registers

- Small piece of memory
- Part of the processor
- Temporary storage of data which is about to or has been processed.

## Difference between general purpose and special purpose registers

- Special purpose registers hold the status of a program whereas general purpose registers hold the temporary data while performing operations.
- Special purpose registers are specialised for a specific use, whereas general purpose registers are used for any purpose.
- General purpose registers can be used by most instructions, whereas special purpose can only be used by certain instructions.

## Special Purpose Registers

400	42
401	40
402	54
403	INC
404	ADD

MAR  
Memory Address Register



MDR  
Memory Data Register



CIR  
Current Instruction Register



PC  
Program Counter



- **Program Counter:** Stores the address of the next instruction to be fetched.
- **MAR:** Stores memory address where data is to be read from.
- **MDR:** Stores data that has just been read from memory.
- **CIR:** Stores the instruction that is currently being executed.
- **IX:** (Index Register) used for indirect addressing.
- **ACC:** (Accumulator) Is the register where the data is worked on (General Purpose register).
- **Status Register:** Is interpreted as independent flags, each flag is set depending on an event.

## Purpose Of Memory Address Register

- Stores the next address to be fetched
- held in the Program Counter (PC)
- The data at this address is then fetched

## Purpose Of Memory Data Register

- Stores the data from the address pointed to by the MAR
- The data in it is copied to the Current Instruction Register (CIR)

## Purpose Of Program Counter

- to store the address / location / memory location of the next instruction to be fetched

## Purpose Of Index Register

- to store a value that is added to an address to give another address
- used in indexed addressing

## Purpose Of Status Register

- to store flags which are set by events // from the results of arithmetic and logic operations and interrupt flags

## System Clock

The system clock is a small part of the computer that sends out regular electrical pulses. These pulses are like tiny "ticks" or "beats" that happen very, very quickly (often millions or billions of times per second). Every "tick" tells the computer, "Do the next step now!"

Describe the purpose of system clock ?

- synchronize operations
- by creating timing signals
- to keep track of the date and time / timestamp files
- to process operations in the correct order / sequence

# **Immediate Access Store**

**The Immediate Access Store (IAS) is a part of RAM where the computer keeps data, instructions, and programs it's currently using, like having everything you need for homework on a desk for easy access. The IAS keeps important information close by, allowing the computer to work faster and more efficiently.**

**Describe what is meant by the Immediate Access Store (IAS) in a computer system.?**

- **Immediate Access Store holds all the data / instructions / programs currently in use**
- **Immediate Access Store is volatile memory**
- **Immediate Access Store has fast access times**



# how data are transferred between various components of the computer system

## Data Bus

- Carries data between processor and memory
- The data bus is bi-directional, meaning it can carry data both to and from the processor.
- used to exchange data

## Address Bus

- The address bus is uni-directional, meaning it only sends information in one direction, from the processor to memory or other components.
- that carries signal related to memory address
- between processor and memory

## Control Bus

- The control bus carries control signals sent by the Control Unit to coordinate and control the actions of different components.
- Bi-directional
- Sends Interrupt signal, Timing signal, Read Signal and Write Signal

## What type of signal Control Bus can transfer?

- Interrupt signal, Timing signal, Read Signal and Write Signal

# Computer Components And Buses Used

Program Counter (PC)	Address Bus	Transfers address to MAR
Memory Data Register (MDR)	Data Bus	The MDR holds data being transferred, so it uses the data bus to move data to/from memory.
Memory Address Register (MAR)	Address Bus	The MAR directly places the memory address on the address bus to access specific memory locations.
Accumulator (ACC)	Data Bus	The ACC interacts with the ALU for calculations, using the data bus for data exchange.
Index Register (IX)	Address Bus	The IX is used for effective address calculation and places these addresses on the address bus when needed.
Current Instruction Register (CIR)	Control Bus	The CIR holds the current instruction, and the control bus coordinates its decoding and execution.
Status Register	Control Bus	The Status Register contains flags that influence control signals, requiring interaction over the control bus.
Arithmetic and Logic Unit (ALU)	Data Bus	The ALU performs calculations, needing the data bus to interact with registers like the ACC.
Control Unit (CU)	Control Bus	The CU sends control signals to manage all components, using the control bus to coordinate actions.
System Clock	Control Bus	The system clock sends timing signals to synchronize operations, using the control bus for communication.
Immediate Access Store (IAS)	Data Bus	The IAS is part of RAM where data is accessed quickly, relying on the data bus to transfer data.

**Explain how the CU, system clock and control bus operate to transfer data between the components of the computer systems?**

- The system clock gives out timing signals
- which are sent on the control bus
- to synchronize the other system components
- The Control Unit initiates data transfer
- by generating signals that are sent on the control bus to other components

## **Factors Affecting The Computer Performance**

### **Data Bus Width**

- The width of a bus determines the number of bits that can be simultaneously transferred
- Increasing the width of a data bus increases the number of bits that can be transferred
- and will allow the transfer of more data each time
- Hence improves processing speed as fewer transfers are needed.

### **Clock Speed**

- One F-E cycle is run on each clock pulse
- So the clock speed dictates the number of instructions that can be run per second and clock speed is measured in GHz
- The faster the clock speed the more instructions can be run per second
- So the processing speed would increase.

## Number Of Cores

- Each core processes one instruction per clock pulse
- More cores mean that sequence of instructions can be split between them and so more than one instruction is executed per clock pulse
- More cores decreases the time taken to complete task.

## Cache Memory

- cache is fast access memory (close to the CPU) and it uses SRAM
- cache stores frequently used instructions / data
- more cache means more instructions / data can be transferred faster
- less swapping between RAM and cache
- prevents the CPU idling while waiting for data which means CPU can continue working without waiting for the data.

## Quantity of RAM

- Increasing the number of RAM will allow more application to reside in memory at the same time, saving disk access time. Less swapping will be there from virtual memory to RAM

**Note : Quantity of RAM is not mentioned separately in 9618 syllabus but is used in Mark Schemes so avoid using Ram as your first priority while answering.**

**A computer has a single 2.1 GHz CPU, Describe how increasing the clock speed to 4GHz can increase the performance of the computer?**

- The CPU can now perform nearly twice as many F-E cycles per second
- Instead of 2.1 billion F-E cycles per second, the CPU can now perform 4 billion F-E cycles per second

**A second computer has a CPU with two 2.1 GHz cores. Explain why the second computer does not always run twice as fast as the computer with one 2.1 GHz CPU.**

- Multiple cores introduce additional overheads means extra work or resources needed.
- because of the need for communication between cores
- Software may not be designed for multiple cores...
- so one of the cores will be left idle
- Memory access speed may not match speed of cores...
- so causing delay
- The two computers may have more differences than just the cores
- one may have more RAM which allows faster multitasking
- one may have a GPU

# Ports

**Acts as an interface between computers  
and other peripheral devices**

## Universal Serial Bus (USB)

**The USB cable consists of a four-wired shielded cable, with two wires for power and the earth, and two wires used for data transmission. When a device is plugged into a computer using one of the USB ports**

### Benefits Of USB

- **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 drivers.
- **USB is now an industrial standard**  
All new computers will be equipped with a USB port.
- **Device may be powered or charged through USB**  
External devices could be charged while working at the computer.

### Drawbacks of USB Port

- **The present transmission rate is limited to less than 500 megabits per second.**
- **The maximum cable length is currently about five meters.**
- **Older USB standards (such as 1.1) may not be supported in the near future.**

## Describe how a data is transmitted through a USB port?

- 1 bit is transferred at a time
- Can be synchronous or asynchronous which means data can be transferred with or without a set timing.
- USB-3 is full duplex and earlier versions are half-duplex which means USB-3 can send and receive data at the same time, while earlier versions can only send or receive data at one time, not both simultaneously.

## High-Definition Multimedia Interface (HDMI)



## Describe how a data is transmitted through a HDMI port?

- Multiple Bits Transferred Simultaneously: HDMI transfers multiple bits of data at once, allowing it to carry high-definition video and audio signals at high speeds.
- Synchronous Transmission: HDMI uses a synchronous transmission method, which means data is sent with a set timing to ensure continuous and stable video and audio playback.
- One-Way Full Duplex Transmission: HDMI is designed for one-way communication, where it transmits both video and audio data simultaneously from a source device (like a computer or Blu-ray player) to a display (like a TV or monitor), ensuring high-quality multimedia output.

## Benefits of HDMI

- Current standard for modern televisions and monitors.
- Allows for a very fast data transfer rate.
- Improved security, helps prevent piracy by the help of HDCP.
- Supports modern digital system.

## Drawbacks of HDMI

- Not a very robust connection (easy to break connection when simply moving devices)
- Limited cable length.

## Video Graphics Array (VGA)



The Video Graphics Array (VGA), introduced by IBM in 1987, is a foundational display standard for computer graphics. It supports a resolution of 640x480 pixels with 16 colors and uses 15-pin D-sub connectors to connect monitors and projectors.

## Describe how data is transmitted through a VGA port??

- **Analog Signal Transmission:** VGA transmits data as analog signals, converting digital video signals from the source device into analog format to display on a monitor. The voltage level of each signal corresponds to the intensity of the respective color channel.
- **Multiple Signals Transferred Simultaneously:** VGA uses multiple signal lines to transmit data, including red, green, and blue (RGB) color signals, along with horizontal and vertical sync signals to coordinate the display.
- **Synchronous Transmission:** VGA relies on synchronous transmission, where the horizontal and vertical sync signals ensure that the image is displayed in the correct position on the screen without distortion.
- **One-Way Communication:** VGA supports one-way communication, transmitting video data from the source device to the display without audio support, as it is not designed to carry sound signals.

### Advantages of VGA:

- Simpler technology
- Only one standard available
- Easy to split the signals and connect a number of devices from one source
- The connection is very secure
- VGA supports a higher refresh rate

### Disadvantages of VGA:

- It does not support audio
- Outdated as modern laptops and computers rarely have a VGA port
- Easy to bend the pins when making connections

## Advantages of HDMI over VGA:

- Current Standard for Modern Devices:

HDMI is the standard for modern televisions, monitors, and multimedia devices, making it more widely compatible compared to the outdated VGA.

- Faster Data Transfer Rate:

HDMI supports a very fast data transfer rate, allowing for high-definition video and multi-channel audio to be transmitted seamlessly, unlike VGA, which is limited to lower resolutions.

- Enhanced Security:

HDMI incorporates HDCP (High-bandwidth Digital Content Protection) to prevent piracy, ensuring secure content transmission, a feature that VGA lacks.

- Support for Modern Digital Systems:

HDMI is designed for digital signals, ensuring better quality video and audio output compared to VGA's analog transmission.

## Disadvantages of HDMI Compared to VGA:

- Less Robust Connection:

HDMI connections can be fragile and prone to breaking when devices are moved, while VGA connections, with screw locks, are more secure.

- Cable Length Limitations:

HDMI cables have limited effective lengths before signal quality degrades, while VGA can maintain signal quality over longer distances with proper shielding.

## Fetch Execute Cycle

- PC holds the address of the next instruction to be fetched
- The address in program counter is copied to MAR
- Program counter is incremented.
- The instruction is copied from address stored in MAR to MDR
- Then instruction from the MDR is copied to CIR.

**MAR  $\leftarrow$  [PC]**

**PC  $\leftarrow$  [PC] + 1**

**MDR  $\leftarrow$  [[MAR]]**

**CIR  $\leftarrow$  [MDR]**

The third instruction [[MAR]] has double brackets.

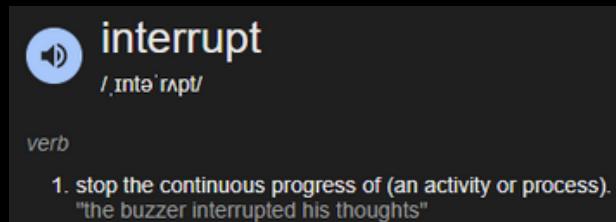
State the purpose of double brackets.

Ans : The content of MAR is an address, it is the contents of that address which is transferred to MDR.

Describe the role of the registers in the Fetch-Execute Cycle

- The Program Counter (PC) holds the address of the next instruction ...
- ...and the contents are incremented/changed to the next address each cycle
- The Memory Address Register (MAR) holds the address to fetch the data (from the PC)
- The Memory Data Register (MDR) holds the data at the address in MAR
- The instruction is transferred to the Current Instruction Register (CIR) for decoding and execution

# Interrupt



## What is meant by Interrupt? ?

- A signal from source/device.
- Telling the processor that its attention is needed.

### Hardware Interrupt

- Printer out of paper.
- No CD in drive.

### Software Interrupt

- Run-time error.
- A running program needs input.
- Buffer Overflow
- Division By Zero
- Out of memory bounds

## Explain how processor handles interrupt ?

- At the start of each Fetch Execute Cycle, the processor checks for interrupt.
- Checks if an interrupt flag is set.
- Processor identifies the source of interrupts.
- Processor checks priority of interrupts.
- If interrupt priority is high enough, the lower priority interrupts are disabled.
- Processor saves current content of registers on a stack.
- Processor calls ISR (Interrupt Service Routine).
- Address of ISR is loaded in PC (Program Counter).
- When servicing of interrupt is complete, processor restores registers.

**An interrupt is generated when a key is pressed on a computer keyboard**

**Explain how processor handles this interrupt ?**

- An interrupt flag is raised in the (interrupt) register.
- At the end of the current FE cycle // at the start of the next FE cycle.
- The system checks the interrupt register for higher priority interrupts than the current process.
- If true, it stores the current contents of the registers on the stack.
- The appropriate interrupt service routine (ISR) for the key press is called.
- The input data from the keyboard is processed.
- The contents of the registers are restored from the stack.
- ... and control is passed back to the previous process.

**Give three different events that can generate an interrupt?**

- Hardware faults
- I/O requests
- Software errors

# Central Processing Unit

## Question 1

3 A processor has one general purpose register, the Accumulator (ACC), and several special purpose registers.

(a) Complete the following description of the role of the registers in the fetch-execute cycle by writing the missing registers.

The ..... holds the address of the next instruction to be loaded. This address is sent to the .....

The ..... holds the data fetched from this address.

This data is sent to the ..... and the Control Unit decodes the instruction's opcode.

The ..... is incremented.

[5]

## Question 2

5 Seth uses a computer for work.

(a) Complete the following descriptions of internal components of a computer by writing the missing terms.

The ..... transmits the signals to coordinate events based on the electronic pulses of the .....

The ..... carries data to the components, while the ..... carries the address where data needs to be written to or read from.

The ..... performs mathematical operations and logical comparisons.

[5]

- (b)** Describe the ways in which the following factors can affect the performance of his laptop computer.

Number of cores

.....  
.....  
.....  
.....

Clock speed

.....  
.....  
.....  
.....

[4]

### Question 3

6 (a) There are **two** errors in the following register transfer notation for the fetch-execute cycle.

```
1  MAR ← [PC]  
2  PC ← [PC] - 1  
3  MDR ← [MAR]  
4  CIR ← [MDR]
```

Complete the following table by:

- identifying the line number of each error
- describing the error
- writing the correct statement.

Line number	Description of the error	Correct statement

[4]

## Question 4

7 A computer has system software.

- (a) The Operating System handles interrupts.

Tick () **one** box in each row to identify whether each event is an example of a hardware interrupt or a software interrupt.

Event	Hardware interrupt	Software interrupt
Buffer full		
Printer is out of paper		
User has pressed a key on the keyboard		
Division by zero		
Power failure		
Stack overflow		

[3]

## Question 5

8 The Von Neumann model for a computer system uses registers.

- (a) Describe the role of the following special purpose registers in the fetch-execute (F-E) cycle.

(i) Memory Address Register (MAR) .....

.....  
.....  
.....

Memory Data Register (MDR) .....

.....  
.....  
.....

[4]

- (ii) Another special purpose register is the Index Register.

Identify **one other** special purpose register used in the Von Neumann model for a computer system.

.....  
.....

[1]

## Question 6

- 6 (a) A computer system is designed using the basic Von Neumann model.

- (i) Describe the role of the registers in the Fetch-Execute (F-E) cycle.

[5]

- (ii) Describe when interrupts are detected in the F-E cycle **and** how the interrupts are handled.

Detected .....

Handled .....

.....

.....

.....

.....

.....

.....

.....

[5]

- (b) Identify **one** factor that can affect the performance of the computer system **and** state how it impacts the performance.

Factor \_\_\_\_\_

**Impact** .....

[View Details](#) | [Edit](#) | [Delete](#)

.....

[2]

## Question 7

- 2 (a) The Fetch-Execute (F-E) cycle is represented in register transfer notation.

Describe each of the given steps.

Step	Description
$PC \leftarrow [PC] + 1$	..... ..... .....
$MDR \leftarrow [[MAR]]$	..... ..... .....
$MAR \leftarrow [PC]$	..... ..... .....

[3]

**(b)** Explain how interrupts are handled during the F-E cycle.

[51]

## Question 8

- 5 (a) State what is meant by the **stored program concept** in the Von Neumann model of a computer system.

.....  
..... [1]

- (b) A Central Processing Unit (CPU) contains several special purpose registers and other components.

- (i) State the role of the following registers.

Program Counter (PC) .....

.....  
.....

Index Register (IX) .....

.....  
.....

Status Register (SR) .....

.....  
.....

[3]

- (ii) Tick () **one** box in each row to identify the system bus used by each CPU component.

CPU component	Data bus	Address bus	Control bus
System clock			
Memory Address Register (MAR)			

[1]

- (iii) Describe the purpose of the Control Unit (CU) in a CPU.

.....  
.....  
.....  
..... [2]

- (c) Describe the purpose of an interrupt in a computer system.

.....  
.....  
.....  
.....

[2]

- (d) Identify **two** causes of a software interrupt.

1 .....

.....

2 .....

.....

[2]

## Question 9

- (d) Explain the reasons why increasing the amount of cache memory can improve the performance of a CPU.

.....  
.....  
.....  
.....

[2]

- (e) State the name of a peripheral device port that provides a physical connection in the computer for each of these peripherals.

3D printer .....

Monitor .....

[2]

## Question 10

- (c) Write the register transfer notation for each of the stages in the fetch-execute cycle described in the table.

Description	Register transfer notation
Copy the address of the next instruction into the Memory Address Register.	
Increment the Program Counter.	
Copy the contents of the Memory Data Register into the Current Instruction Register.	

[3]

## Question 11

- 4 (a) A Central Processing Unit (CPU) contains several special purpose registers and other components.

- (i) State the roles of the following registers.

Memory Address Register (MAR) .....  
.....  
.....

Memory Data Register (MDR) .....  
.....  
.....

[2]

- (ii) State when interrupts are detected during the Fetch-Execute (F-E) cycle.

.....  
.....

[1]

- (b)** A computer system contains a system clock.

Describe the purpose of the system clock.

.....  
.....  
.....  
.....

[2]

- (c)** Upgrading secondary storage to solid state typically improves the performance of computer systems.

Identify **one other** upgrade to the hardware **and** explain why it improves the performance of a computer system.

Upgrade .....

Explanation .....

.....  
.....

[2]

## Question 12

- 6** An interrupt is generated when a key is pressed on a computer keyboard.

Explain how the computer handles this interrupt.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

[5]

## Question 13

5 A student has purchased a new laptop.

- (a) The laptop is designed using the Von Neumann model for a computer system.

Identify **two** types of signal that a control bus can transfer.

1 .....

2 .....

[2]

- (b) Describe **two** ways the hardware of a laptop can be upgraded to improve the performance **and** explain how each upgrade improves the performance.

1 .....

.....

.....

.....

.....

2 .....

.....

.....

.....

.....

[4]

(c) Peripherals are connected to the laptop using ports.

(i) A printer is connected to a Universal Serial Bus (USB) port.

Describe how data is transmitted through a USB port.

.....  
.....  
.....  
.....

[1]

(ii) A monitor is connected to the laptop using a different type of port.

Identify **one** other type of port that can be used to connect the monitor.

.....

[1]

## Question 14

- (b) The computer has a Control Unit (CU), system clock and control bus.

Explain how the CU, system clock and control bus operate to transfer data between the components of the computer system.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
..... [4]

- (c) Complete the table by writing the register transfer notation for each stage of the Fetch-Execute (F-E) cycle given in the table.

Stage description	Register transfer notation
The Program Counter (PC) is incremented	.....
The data in the address stored in the Memory Address Register (MAR) is copied to the Memory Data Register (MDR)	.....

[2]

## Question 15

5 The Central Processing Unit (CPU) of the basic Von Neumann model for a computer system contains several special purpose registers.

- (a) The Memory Data Register (MDR), Index Register (IX) and the Accumulator (ACC) are examples of special purpose registers.

Identify **two other** special purpose registers **and** state their role in the CPU.

Special purpose register 1 .....

Role .....

.....

Special purpose register 2 .....

Role .....

.....

[4]

- (b) Describe what is meant by the **Immediate Access Store (IAS)** in a computer system.

.....

.....

.....

[2]

(c) A computer has a single 2.1 GHz CPU.

- (i) Describe how increasing the clock speed to 4 GHz can increase the performance of the computer.

.....  
.....  
.....

[1]

- (ii) A second computer has a CPU with two 2.1 GHz cores.

Explain why the second computer does not always run twice as fast as the computer with one 2.1 GHz CPU.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

[5]

## Question 16

- (d) Identify **one** port that could be used to connect the virtual reality headset to the laptop.

Justify your choice.

Port .....

Justification .....

.....  
.....  
.....

[3]

## Question 17

- 9 A computer system is designed using the basic Von Neumann model.

Registers and buses are components in the Von Neumann model.

- (a) (i) Identify **three other** components in the Von Neumann model of a computer system.

Do not include registers or buses in your answers.

1 .....

2 .....

3 .....

[3]

- (ii) Identify **two** differences between special purpose registers and general purpose registers.

1 .....

.....

2 .....

.....

[2]

- (b) The following incomplete table contains steps of the Fetch-Execute (F-E) cycle and their descriptions.

Complete the table by writing the missing steps using register transfer notation **and** the missing descriptions.

Step	Description
.....	The address in PC is incremented.
MDR $\leftarrow$ [ [MAR] ]	..... ..... .....
MAR $\leftarrow$ [PC]	..... ..... .....
.....	The contents of MDR are copied into CIR.

[4]

- (c) Interrupts can be caused by software programs or hardware devices.

State **one** cause of a software interrupt.

.....  
..... [1]

- (d) The following statements describe the stages that the CPU performs when an interrupt is detected.

There are **three** missing statements.

Write the letter of the missing statements from the table in the correct place to complete the description.

- 1 At the end of each Fetch-Execute (F-E) cycle, the processor checks if an interrupt flag is set.
- 2 .....
- 3 If the interrupt priority is high enough, the processor saves the current contents of the registers.
- 4 .....
- 5 When servicing of the interrupt is complete, the processor restores the registers.
- 6 .....

Letter	Stage
A	The address of the Interrupt Service (ISR) handling routine is loaded into the Program Counter (PC).
B	Lower priority interrupts are re-enabled.
C	The device causing the interrupt transfers data to the CPU.
D	The processor identifies the source of the interrupt and checks the priority of the interrupt.
E	The ISR is incremented.

[3]

## Question 18

- (b) The ACC is a general purpose register. The IX is a special purpose register.

Identify two other special purpose registers used in the fetch-execute cycle and describe their role in the cycle.

## **Register 1**

**Role** \_\_\_\_\_

## **Register 2**

## **Role**

141

## Question 19

- (iii) The peripheral devices are plugged into USB ports of the computer.

Describe **two** benefits of connecting the peripheral devices using a USB port.

---

Digitized by srujanika@gmail.com

---

Digitized by srujanika@gmail.com

2 .....

---

Digitized by srujanika@gmail.com

---

[View Details](#) | [Edit](#) | [Delete](#)

[4]

## Question 20

(c) Billy's computer has several ports.

(i) State the purpose of a port.

..... [1]

(ii) Identify one type of port.

..... [1]

## Question 21

- 3 The fetch-execute cycle is shown in register transfer notation.

```
01      MAR ← [PC]  
02      PC ← [PC] - 1  
03      MDR ← [MAR]  
04      CIR ← [MAR]
```

- (a) There are **three** errors in the fetch-execute cycle shown.

Identify the line number of each error and give the correction.

Line number .....

Correction .....

Line number .....

Correction .....

Line number .....

Correction .....

[3]

- (b) A processor's instruction set can be grouped according to their function. For example, one group is the input and output of data.

Identify **two** other groups of instructions.

1 .....

.....

2 .....

.....

[2]

## Question 22

(c) The processor handles interrupts within the fetch-execute cycle.

(i) Give **one** example of a hardware interrupt and **one** example of a software interrupt.

Hardware .....

.....

Software .....

[2]

(ii) Explain how the processor handles an interrupt.

.....

.....

.....

.....

.....

.....

.....

[5]

## Question 23

- 1 Von Neumann is an example of a computer architecture.
- (a) The diagram has registers used in Von Neumann architecture on the left and descriptions on the right.

Draw **one** line to match each register with its correct description.

Register	Description
Current Instruction Register	Stores the data that has just been read from memory, or is about to be written to memory
Memory Address Register	Stores the instruction that is being decoded and executed
Program Counter	Stores the address of the input device from which the processor accesses the instruction
Memory Data Register	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]

- (b) Many components of the computer system transfer data between them using buses. One example of a bus is an address bus.

- (i) Name **two** other buses that exist within a computer and give the purpose of each.

Bus 1 .....

Purpose .....

.....

Bus 2 .....

Purpose .....

.....

[4]

- (ii) State the benefit of increasing the address bus width from 16 bits to 32 bits.

.....

[1]

## Question 24

- 8 The Von Neumann model uses a series of registers.

- (a) Explain what is meant by the term register.

12

12

- (b) (i) Explain the purpose of the Memory Data Register (MDR).

[2]

12

- (iii) Name two registers, other than the MDR, that are used in the fetch-execute cycle.

**Register 1** .....

**Register 2** \_\_\_\_\_

四

- (e) X is a register. The current contents of X are:

1	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

- (i) The current contents of register X represent an unsigned binary integer.

Convert the value in X into denary.

11

- (iii) The current contents of register X represent a Binary Coded Decimal.

Convert the value in X into decimal.

111

- (iii) The current contents of register X stores a two's complement binary integer.

Convert the value in X into degrees

14

## Question 25

3 A computer is designed using the Von Neumann model.

- (a) Describe the role of the Arithmetic and Logic Unit (ALU) and Control Unit (CU) in the Von Neumann model.

ALU .....

---

---

---

CU .....

---

---

---

[4]

- (b) Describe the role of the Status Register and Program Counter (PC).

Status Register .....

---

---

---

PC .....

---

---

---

[4]

## Question 26

- 4 A student has written the steps of the fetch stage of the fetch-execute (FE) cycle in register transfer notation. The student has made some errors.

Line 1      MDR  $\leftarrow$  [PC]  
Line 2      PC  $\leftarrow$  PC + 1  
Line 3      MDR  $\leftarrow$  [MAR]  
Line 4      CIR  $\leftarrow$  PC

- (a) Identify the line numbers of three errors that the student has made. Write the correct notation for each error.

Line number of error	Correct notation

[3]

- (b) One stage of the FE cycle includes checking for interrupts.

- (i) Give three different events that can generate an interrupt.

- 1 .....  
2 .....  
3 .....

[3]

- (iii) Explain how interrupts are handled during the fetch-execute cycle.

15

- (c) The processor uses buses in the FE cycle.

The diagram shows three buses and two descriptions.

**Draw one line from each bus to its appropriate description.**

Bus	Description
Control bus	Unidirectional (one direction)
Address bus	
Data bus	Bidirectional (two directions)

12

## Question 27

6 The fetch-execute (FE) cycle uses special purpose registers.

- (a) The stages in the FE cycle are shown in register transfer notation.

MAR  $\leftarrow$  [.....]

PC  $\leftarrow$  PC + 1

.....  $\leftarrow$  [ [MAR] ]

.....  $\leftarrow$  [MDR]

- (i) The steps shown in part (a) are incomplete.

Write the missing register names in the spaces in part (a).

[3]

- (ii) The third instruction [ [MAR] ] has double brackets.

State the purpose of the double brackets.

.....  
.....  
.....

[1]

- (b) One stage of the FE cycle includes checking for interrupts.

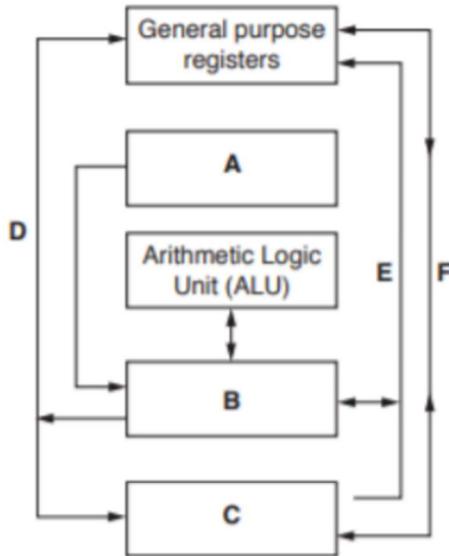
State what is meant by an interrupt.

.....  
.....  
.....  
.....

[2]

## Question 28

- 4 (a) The diagram shows the components and buses found inside a typical Personal Computer (PC).



Some components and buses only have labels **A** to **F** to identify them.

For each label, choose the appropriate title from the following list. The title for label **D** is already given.

- Control bus
- System clock
- Data bus
- Control unit
- Main memory
- Secondary storage

**A** .....

**B** .....

**C** .....

**D** Address bus

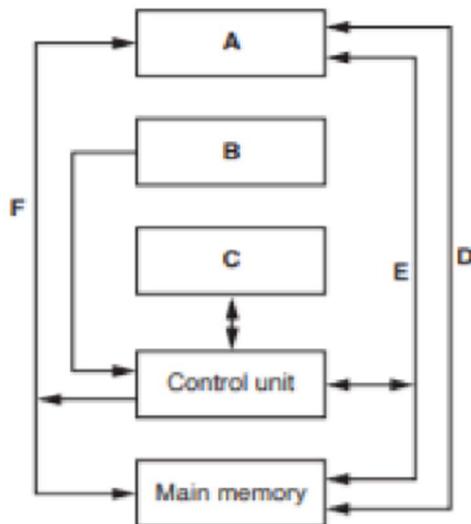
**E** .....

**F** .....

[5]

## Question 29

- 4 The following diagram shows the components and buses found inside a typical personal computer (PC).



- (a) Some components and buses only have labels A to F to identify them.

For each label, choose the appropriate title from the following list. The title for label D is already given.

- Control bus
- Address bus
- Arithmetic Logic Unit (ALU)
- General purpose registers
- Secondary storage
- System clock

- A .....  
B .....  
C .....  
D Data bus  
E .....  
F .....

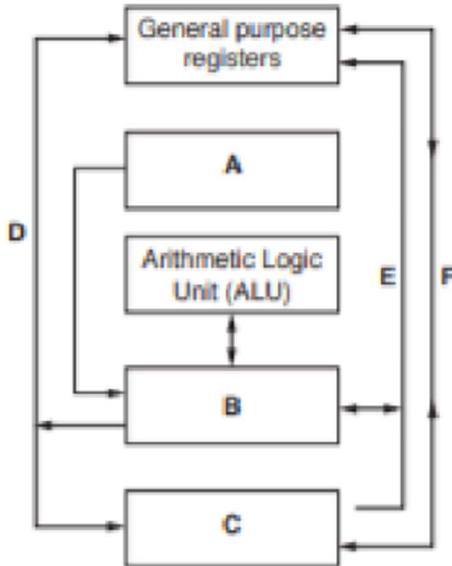
[5]

- (b) Clock speed is a factor that affects the performance of a PC. Explain this statement.

[2]

## Question 30

- 4 (a) The diagram shows the components and buses found inside a typical Personal Computer (PC).



Some components and buses only have labels A to F to identify them.

For each label, choose the appropriate title from the following list. The title for label D is already given.

- Control bus
- System clock
- Data bus
- Control unit
- Main memory
- Secondary storage

- A .....  
B .....  
C .....  
D Address bus  
E .....  
F .....

[5]

## Question 31

- 3 (a) Describe how special purpose registers are used in the fetch stage of the fetch-execute cycle.

[4]

.[4]

- (b) Use the statements A, B, C and D to complete the description of how the fetch-execute cycle handles an interrupt.

- A the address of the Interrupt Service Routine (ISR) is loaded to the Program Counter (PC).
- B the processor checks if there is an interrupt.
- C when the ISR completes, the processor restores the register contents.
- D the register contents are saved.

At the end of the cycle for the current instruction .....

If the interrupt flag is set, ..... and .....

The interrupted program continues its execution.

[4]

## Question 32

8 (a) Explain how the width of the data bus and system clock speed affect the performance of a computer system.

Width of the data bus .....

.....  
.....  
.....

Clock speed .....

.....  
.....  
.....

[3]

(c) The table shows six stages in the von Neumann fetch-execute cycle.

Put the stages into the correct sequence by writing the numbers 1 to 6 in the right hand column.

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

[6]

## Question 33

5 (a) Name and describe three buses used in the von Neumann model.

Bus 1.....

Description.....

.....  
.....

Bus 2.....

Description.....

.....  
.....

Bus 3 .....

Description.....

.....

[6]

(b) The sequence of operations shows, in register transfer notation, the fetch stage of the fetch-execute cycle.

1 MAR  $\leftarrow$  [PC]  
2 PC  $\leftarrow$  [PC] + 1  
3 MDR  $\leftarrow$  [MAR]  
4 CIR  $\leftarrow$  [MDR]

- [register] denotes contents of the specified register or memory location
- step 1 above is read as "the contents of the Program Counter are copied to the Memory Address Register"

(i) Describe what is happening at step 2.

.....

[1]

(ii) Describe what is happening at step 3.

.....  
.....  
.....

[1]

(iii) Describe what is happening at step 4.

..... [1]

(c) Describe what happens to the registers when the following instruction is executed:

LDD 35

.....

[2]

(d) (i) Explain what is meant by an interrupt.

.....

[2]

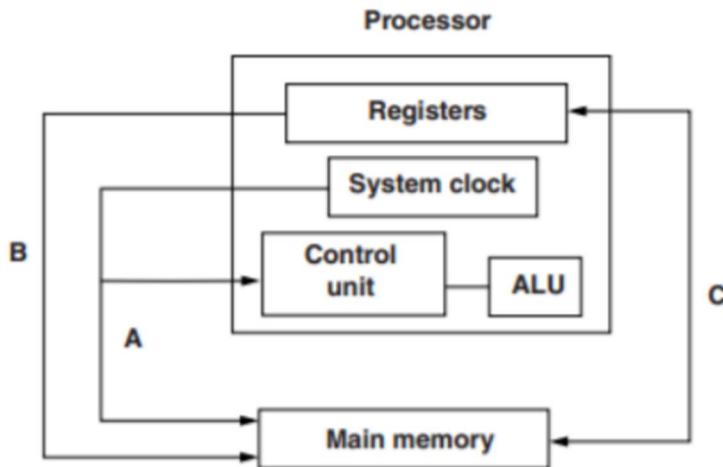
(ii) Explain the actions of the processor when an interrupt is detected.

.....

[4]

## Question 34

2 (a)



The diagram above shows a simplified form of processor architecture.

Name the three buses labelled A, B and C.

A .....

B .....

C .....

[3]

(b) State the role of each of the following special purpose registers used in a typical processor.

Program Counter .....

.....

Memory Data Register .....

.....

Current Instruction Register .....

.....

Memory Address Register .....

.....

[4]

## Question 35

- 2 (a) Describe how buffers and interrupts are used when printing a large document stored on a hard drive.

[4]

[4]

## Question 36

- 1** Describe the function of the following parts of a processor:

- (i) control unit

[1]

...[1]

- (ii) main memory unit

..... [1]

[11]

- ### (iii) arithmetic and logic unit (ALU)

[1]

[11]

## Question 37

- 1 (a)** Describe the terms *buffer* and *interrupt*.

buffer

---

---

---

interrupt .....  
.....  
..... [2]

- (b) (i)** Explain the role of the buffer and interrupts when a large document of over 200 pages is sent to a laser printer.

[3]

- (ii) The use of two buffers would speed up the printing process.

**Explain why.**

[2]

## Question 38

5 (a) What is meant by an interrupt?

.....  
.....  
.....

[1]

(b) A user starts the printing of a document, and then carries on editing a second document while printing continues.

Explain how interrupts make this possible.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

[4]

## Answer

### Answer 1

3(a)	<p><b>1 mark</b> for each completed statement</p> <p>The <b>Program Counter</b> holds the address of the next instruction to be loaded. This address is sent to the <b>Memory Address Register</b>. The <b>Memory Data Register</b> holds the data fetched from this address. This data is sent to the <b>Current Instruction Register</b> and the Control Unit decodes the instruction's opcode. The <b>Program Counter</b> is incremented.</p>	5
------	--	---

### Answer 2

5(a)	<p><b>1 mark</b> for each term correctly inserted</p> <p>The <b>control unit/bus</b> transmits the signals to coordinate events based on the pulses of the (<b>system</b>) <b>clock</b>.</p> <p>The <b>data bus</b> carries data to components, while the <b>address bus</b> carries the address where data is being written to or read from.</p> <p>The <b>arithmetic logic unit/ALU</b> performs mathematical operations and logical comparisons.</p>	5
5(b)	<p><b>1 mark</b> per bullet point to <b>max 3</b> per factor. <b>max 4</b> overall.</p> <p>Number of cores:</p> <ul style="list-style-type: none"><li>• Each core processes one <u>instruction</u> per clock pulse</li><li>• More/multiple cores mean that <b>sequences of instructions</b> can be split between them</li><li>• ... and so more than one <u>instruction</u> is executed per clock pulse // more <b>sequences of instructions</b> can be run at the same time</li><li>• <b>More</b> cores decreases the time taken to complete task</li></ul> <p>Clock speed:</p> <ul style="list-style-type: none"><li>• Each <u>instruction</u> is executed on a clock pulse // one F-E cycle is run on each clock pulse</li><li>• ... so the clock speed dictates the number of <u>instructions</u> that can be run per second</li><li>• The <b>faster</b> the clock speed the more <u>instructions</u> can be run per second</li></ul>	4

### Answer 3

6(a)	<p><b>1 mark for identification of line and description of error 1 mark for the correct statement</b></p> <table border="1"><thead><tr><th>Line number</th><th>Description of the error</th><th>Correct statement</th></tr></thead><tbody><tr><td>2</td><td>Program Counter should be incremented, not decremented</td><td><math>PC \leftarrow [PC] + 1</math></td></tr><tr><td>3</td><td>It should be the contents of the address in the MAR</td><td><math>MDR \leftarrow [MAR]</math></td></tr></tbody></table>	Line number	Description of the error	Correct statement	2	Program Counter should be incremented, not decremented	$PC \leftarrow [PC] + 1$	3	It should be the contents of the address in the MAR	$MDR \leftarrow [MAR]$	4
Line number	Description of the error	Correct statement									
2	Program Counter should be incremented, not decremented	$PC \leftarrow [PC] + 1$									
3	It should be the contents of the address in the MAR	$MDR \leftarrow [MAR]$									

### Answer 4

7(a)	<p><b>1 mark per pair of rows (shaded &amp; unshaded)</b></p> <table border="1"><thead><tr><th>Event</th><th>Hardware Interrupt</th><th>Software Interrupt</th></tr></thead><tbody><tr><td>Buffer full</td><td></td><td>✓</td></tr><tr><td>Printer is out of paper</td><td>✓</td><td></td></tr><tr><td>User has pressed a key on the keyboard</td><td>✓</td><td></td></tr><tr><td>Division by zero</td><td></td><td>✓</td></tr><tr><td>Power failure</td><td>✓</td><td></td></tr><tr><td>Stack overflow</td><td></td><td>✓</td></tr></tbody></table>	Event	Hardware Interrupt	Software Interrupt	Buffer full		✓	Printer is out of paper	✓		User has pressed a key on the keyboard	✓		Division by zero		✓	Power failure	✓		Stack overflow		✓	3
Event	Hardware Interrupt	Software Interrupt																					
Buffer full		✓																					
Printer is out of paper	✓																						
User has pressed a key on the keyboard	✓																						
Division by zero		✓																					
Power failure	✓																						
Stack overflow		✓																					

## Answer 5

8(a)(i)	<p><b>1 mark</b> for each bullet point to <b>max 2</b> for each register</p> <p><b>MAR</b></p> <ul style="list-style-type: none"> <li>• Stores the next <u>address</u> to be fetched</li> <li>• ... held in the Program Counter (PC)</li> <li>• The data at this address is then fetched</li> </ul> <p><b>MDR</b></p> <ul style="list-style-type: none"> <li>• Stores the data from the address pointed to by the MAR</li> <li>• The data in it is copied to the Current Instruction Register (CIR)</li> </ul>	4
8(a)(ii)	<p><b>1 mark</b> for a correct register e.g. Program Counter (PC) Current Instruction Register (CIR) Status register Interrupt register</p>	1

## Answer 6

6(a)(i)	<p><b>1 mark</b> per bullet point to <b>max 5</b></p> <ul style="list-style-type: none"> <li>• The Program Counter (PC) holds the <b>address</b> of the next instruction ...</li> <li>• ...and the <b>contents</b> are incremented / changed to the next address each cycle</li> <li>• The Memory Address Register (MAR) holds the address to fetch the data (from the PC)</li> <li>• The Memory Data Register (MDR) holds the data at the address in MAR</li> <li>• The instruction is transferred to Current Instruction Register (CIR) for decoding and execution</li> </ul>	5
6(a)(ii)	<p><b>1 mark</b> for detection</p> <ul style="list-style-type: none"> <li>• At the start/end of a FE cycle</li> </ul> <p><b>1 mark</b> for handling to <b>max 4</b></p> <ul style="list-style-type: none"> <li>• Priority is checked</li> <li>• If lower priority than current process continue with F-E cycle</li> <li>• If higher priority than current process ...</li> <li>• ... state of current process is / registers are stored on stack</li> <li>• Location / type of interrupt identified...</li> <li>• ...appropriate ISR is called to handle the interrupt</li> <li>• When ISR finished, check for further interrupts (of high priority) / return to step 1</li> <li>• Otherwise load data from stack and continue with process</li> </ul>	5

6(b)	<p><b>1 mark</b> for factor <b>1 mark</b> for impact</p> <p>e.g.</p> <ul style="list-style-type: none"> <li>• Clock speed...</li> <li>• ...higher clock speed means more FE cycles per second</li> <li>• Number of cores...</li> <li>• ...means more instructions can be carried out simultaneously</li> <li>• Bus width ...</li> <li>• ...allows the transfer of more data each time // allows more memory locations to be directly accessed</li> <li>• Cache ...</li> <li>• ... the higher capacity the more frequently used instructions it can store for fast access</li> </ul>	2
------	---	---

## Answer 7

2(a)	<p><b>1 mark</b> for each correct description</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">Step</th><th style="text-align: center; padding: 5px;">Description</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;">PC <math>\leftarrow</math> [PC] + 1</td><td style="padding: 5px;"><b>Address</b> in PC is incremented</td></tr> <tr> <td style="padding: 5px;">MDR <math>\leftarrow</math> [MAR]</td><td style="padding: 5px;">The data in the address held in the MAR is copied to the MDR</td></tr> <tr> <td style="padding: 5px;">MAR <math>\leftarrow</math> [PC]</td><td style="padding: 5px;">The <b>contents</b> of the PC are copied to the MAR</td></tr> </tbody> </table>	Step	Description	PC $\leftarrow$ [PC] + 1	<b>Address</b> in PC is incremented	MDR $\leftarrow$ [MAR]	The data in the address held in the MAR is copied to the MDR	MAR $\leftarrow$ [PC]	The <b>contents</b> of the PC are copied to the MAR	3
Step	Description									
PC $\leftarrow$ [PC] + 1	<b>Address</b> in PC is incremented									
MDR $\leftarrow$ [MAR]	The data in the address held in the MAR is copied to the MDR									
MAR $\leftarrow$ [PC]	The <b>contents</b> of the PC are copied to the MAR									
2(b)	<p><b>1 mark</b> per point to <b>max 5</b></p> <ul style="list-style-type: none"> <li>• Check for interrupt at start/end of an F-E cycle</li> <li>• Priority is checked</li> <li>• If lower priority than current process continue with F-E cycle</li> <li>• If higher priority than current process ... <ul style="list-style-type: none"> <li>• ... state of current process is / registers are stored on stack</li> <li>• Location/type of interrupt identified</li> <li>• Appropriate ISR is called to handle the interrupt</li> <li>• When ISR finished, check for further interrupts (of higher priority) / return to step 1</li> <li>• Otherwise load data from stack and continue with next F-E cycle (of process)</li> </ul> </li> </ul>	5								

## Answer 8

5(a)	Instructions and data are stored in <b>the same</b> memory space / in main memory.	1												
5(b)(i)	<p><b>1 mark</b> for each special purpose register:</p> <p>Program Counter (PC):</p> <ul style="list-style-type: none"> <li>to <b>store</b> the address / location / memory location of the <u>next</u> instruction to be fetched</li> </ul> <p>Index Register (IX):</p> <ul style="list-style-type: none"> <li>to <b>store</b> a value that is added to an address to give another address</li> </ul> <p>Status Register (SR):</p> <ul style="list-style-type: none"> <li>to <b>store flags</b> which are set by events // from the results of arithmetic and logic operations and interrupt flags</li> </ul>	3												
5(b)(ii)	<p><b>1 mark</b> for both rows:</p> <table border="1"> <thead> <tr> <th>CPU component</th> <th>Data bus</th> <th>Address bus</th> <th>Control bus</th> </tr> </thead> <tbody> <tr> <td>System clock</td> <td></td> <td></td> <td>✓</td> </tr> <tr> <td>Memory Address Register (MAR)</td> <td></td> <td>✓</td> <td></td> </tr> </tbody> </table>	CPU component	Data bus	Address bus	Control bus	System clock			✓	Memory Address Register (MAR)		✓		1
CPU component	Data bus	Address bus	Control bus											
System clock			✓											
Memory Address Register (MAR)		✓												
5(b)(iii)	<p><b>1 mark</b> for each bullet point (<b>max 2</b>):</p> <ul style="list-style-type: none"> <li>to coordinate / synchronise the actions of other components in the CPU</li> <li>to send / receive control signals along the control bus</li> <li>to manage the execution of instructions (in sequence)</li> <li>to control the communication between the components of the CPU</li> </ul>	2												
5(c)	<p><b>1 mark</b> for each bullet point:</p> <ul style="list-style-type: none"> <li>to <b>send a signal</b> from a device or process</li> <li>... seeking the attention of the processor</li> </ul>	2												
5(d)	<p><b>1 mark</b> for each bullet point (<b>max 2</b>).</p> <p>For example:</p> <ul style="list-style-type: none"> <li>division by zero // runtime error in a program</li> <li>attempt to access an invalid memory location</li> <li>array index out of bounds</li> <li>stack overflow</li> </ul>	2												

## Answer 9

7(d)	<b>1 mark</b> for each bullet point ( <b>max 2</b> ): <ul style="list-style-type: none"><li>cache is fast access memory (close to the CPU)</li><li>cache stores frequently used instructions / data</li><li>... more cache means more instructions / data can be transferred faster</li><li>... less swapping between RAM and cache</li><li>prevents the CPU idling while waiting for data</li></ul>	2
7(e)	<b>1 mark</b> for each device. <ul style="list-style-type: none"><li><b>3D printer:</b> USB port / COM port</li><li><b>Monitor:</b> HDMI / VGA / USB / DisplayPort</li></ul>	2

## Answer 10

7(c)	<b>1 mark</b> for each correct row: <table><thead><tr><th>Description</th><th>Register transfer notation</th></tr></thead><tbody><tr><td>Copy the address of the next instruction into the Memory Address Register.</td><td><math>\text{MAR} \leftarrow [\text{PC}]</math></td></tr><tr><td>Increment the Program Counter.</td><td><math>\text{PC} \leftarrow [\text{PC}] + 1</math></td></tr><tr><td>Copy the contents of the Memory Data Register into the Current Instruction Register.</td><td><math>\text{CIR} \leftarrow [\text{MDR}]</math></td></tr></tbody></table>	Description	Register transfer notation	Copy the address of the next instruction into the Memory Address Register.	$\text{MAR} \leftarrow [\text{PC}]$	Increment the Program Counter.	$\text{PC} \leftarrow [\text{PC}] + 1$	Copy the contents of the Memory Data Register into the Current Instruction Register.	$\text{CIR} \leftarrow [\text{MDR}]$	3
Description	Register transfer notation									
Copy the address of the next instruction into the Memory Address Register.	$\text{MAR} \leftarrow [\text{PC}]$									
Increment the Program Counter.	$\text{PC} \leftarrow [\text{PC}] + 1$									
Copy the contents of the Memory Data Register into the Current Instruction Register.	$\text{CIR} \leftarrow [\text{MDR}]$									

## Answer 11

4(a)(i)	<p><b>1 mark for each register:</b></p> <p>MAR:</p> <ul style="list-style-type: none"> <li>• <b>holds</b> address in memory from which data will be read / to which data will be written</li> </ul> <p>MDR:</p> <ul style="list-style-type: none"> <li>• <b>holds</b> the data/instructions which has been read from or is to be written to the address in the MAR</li> </ul>	2
4(a)(ii)	after completion of the execute stage // before the cycle begins	1
4(b)	<p><b>1 mark for each bullet point (max 2):</b></p> <ul style="list-style-type: none"> <li>• synchronise operations</li> <li>• ... by creating timing signals</li> <li>• to keep track of the date and time / timestamp files</li> <li>• to process operations in the correct order / sequence</li> </ul>	2
4(c)	<p><b>1 mark for identification of a correct upgrade:</b>  <b>1 mark for a corresponding explanation:</b></p> <p>Examples:</p> <ul style="list-style-type: none"> <li>• increase quantity of RAM <ul style="list-style-type: none"> <li>... so allowing more applications to reside in memory at the same time, saving disk access times</li> </ul> </li> <li>• increase the size of cache memory <ul style="list-style-type: none"> <li>... so that the CPU can continue working without waiting for data</li> </ul> </li> <li>• increase clock speed <ul style="list-style-type: none"> <li>... so that more instructions are performed in a time period</li> </ul> </li> <li>• increase the number of processors / cores <ul style="list-style-type: none"> <li>... so that more instructions are performed in parallel</li> </ul> </li> </ul>	2

## Answer 12

6	<b>1 mark each to max 5</b> <ul style="list-style-type: none"><li>• An interrupt flag is raised in the (interrupt) register</li><li>• At the end of the current FE cycle // at the start of the next FE cycle</li><li>• The system checks the interrupt register for higher priority interrupts than current process</li><li>• If true, it stores the current contents of the registers on the stack</li><li>• The appropriate interrupt service routine (ISR) for the key press is called</li><li>• The input data from the keyboard is processed</li><li>• The contents of the registers are restored from the stack</li><li>• ... and control is passed back to previous process</li></ul>	5
---	---	---

## Answer 13

5(a)	<b>1 mark each to max 2</b>  Examples: <ul style="list-style-type: none"><li>• Interrupt</li><li>• Timing</li><li>• Read</li><li>• Write</li></ul>	2
5(b)	<b>1 mark for description; 1 mark for corresponding explanation</b>  Examples <ul style="list-style-type: none"><li>• Increase number of cores</li><li>• Each core can <b>independently</b> carry out a process at the same time // so that more instructions are performed <b>in parallel</b></li><li>• Increase RAM capacity</li><li>• ... allowing more applications to reside in memory at the same time, saving disk access times</li><li>• Increase cache memory</li><li>• More data can be stored in fast access so less time is spent accessing from RAM</li><li>• Increase clock speed</li><li>• More Fetch-Decode-Execute (FDE) cycles can run each second / <b>per unit time</b></li></ul>	4

5(c)(i)	<b>1 mark</b> for a correct answer  <ul style="list-style-type: none"> <li>• 1 bit is transferred at a time</li> <li>• Can be synchronous <b>or</b> asynchronous</li> <li>• USB-3 is full duplex <b>and</b> earlier versions are half-duplex</li> </ul>	1
5(c)(ii)	<b>1 mark</b> for identification of a suitable port  Examples <ul style="list-style-type: none"> <li>• HDMI</li> <li>• DisplayPort</li> </ul>	1

## Answer 14

7(b)	<b>1 mark</b> each to max 4  <ul style="list-style-type: none"> <li>• The <b>system clock</b> gives out <b>timing</b> signals</li> <li>• ... which are sent on the <b>control bus</b></li> <li>• ...to <b>synchronise</b> the other system components</li> <li>• The <b>Control Unit</b> initiates data transfer</li> <li>• ...by generating signals that are sent on the <b>control bus</b> to other components</li> </ul>	4						
7(c)	<b>1 mark</b> for each Register transfer notation  <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">Stage description</th> <th style="text-align: center; padding: 5px;">Register transfer notation</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">The Program Counter (PC) is incremented</td> <td style="padding: 5px;"><math>PC \leftarrow [PC] + 1</math></td> </tr> <tr> <td style="padding: 5px;">The data in the address stored in the Memory Address Register (MAR) is copied to the Memory Data Register (MDR)</td> <td style="padding: 5px;"><math>MDR \leftarrow [[MAR]]</math></td> </tr> </tbody> </table>	Stage description	Register transfer notation	The Program Counter (PC) is incremented	$PC \leftarrow [PC] + 1$	The data in the address stored in the Memory Address Register (MAR) is copied to the Memory Data Register (MDR)	$MDR \leftarrow [[MAR]]$	2
Stage description	Register transfer notation							
The Program Counter (PC) is incremented	$PC \leftarrow [PC] + 1$							
The data in the address stored in the Memory Address Register (MAR) is copied to the Memory Data Register (MDR)	$MDR \leftarrow [[MAR]]$							

## Answer 15

5(a)	<p><b>1 mark</b> for identification of the register and <b>1 mark</b> for role (<b>max 2</b> for each register)</p> <ul style="list-style-type: none"> <li>• Program Counter (PC)</li> <li>• stores the <u>address</u> where the <u>next</u> instruction is to be read from</li> <li>• Memory Address Register (MAR)</li> <li>• stores the address of the memory location (or an I/O component) <b>currently being read from or written to</b></li> <li>• Current Instruction Register (CIR)</li> <li>• holds the instruction currently being decoded and/or executed</li> <li>• Status Register</li> <li>• Contains bits which can be referenced individually and set or cleared depending on the operation e.g. overflow, underflow</li> </ul>	4
5(b)	<p><b>1 mark</b> for each bullet point (<b>max 2</b>)</p> <ul style="list-style-type: none"> <li>• Immediate Access Store holds all the data / instructions / programs currently in use</li> <li>• Immediate Access Store is volatile memory</li> <li>• Immediate Access Store has fast access times</li> </ul>	2
5(c)(i)	<p><b>1 mark</b> for each bullet point (<b>max 1</b>)</p> <ul style="list-style-type: none"> <li>• The CPU can now perform nearly twice as many <b>F-E cycles</b> per second</li> <li>• Instead of 2.1 billion <b>F-E cycles</b> per second, the CPU can now perform 4 billion <b>FE cycles</b> per second</li> </ul>	1
5(c)(ii)	<p><b>1 mark</b> for each bullet point (<b>max 5</b>)</p> <ul style="list-style-type: none"> <li>• Multiple cores introduce additional overheads</li> <li>• ...because of the need for communication between cores</li> <li>• Software may not be designed for multiple cores...</li> <li>• ...so one of the cores will be left idle</li> <li>• Memory access speed may not match speed of cores...</li> <li>• ...so causing delay</li> <li>• The two computers may have more differences than just the cores</li> <li>• ...one may have more RAM which allows faster multitasking</li> <li>• ...one may have a GPU</li> <li>• ...etc.</li> </ul>	5

## Answer 16

7(d)	<p><b>1 mark</b> for naming a correct port <b>2 marks</b> for matching justification</p> <ul style="list-style-type: none"> <li>• USB</li> <li>• ...has fast data transfer speeds for data (to the headset)</li> <li>• ...is a universal/popular cable // universal standard</li>   <li>• HDMI</li> <li>• ...allows video and audio to be transferred (on the same cable)</li> <li>• ...convenience of HDMI as no need for two cables</li> </ul>	3
------	--	---

## Answer 17

9(a)(i)	<p><b>1 mark</b> for each bullet point (<b>max 3</b>).</p> <ul style="list-style-type: none"> <li>• Control unit (CU)</li> <li>• Arithmetic <b>and</b> Logic Unit (ALU)</li> <li>• Immediate Access Store (IAS)</li> <li>• (System clock)</li> </ul>	3										
9(a)(ii)	<p><b>1 mark</b> for each bullet point (<b>max 2</b>).</p> <ul style="list-style-type: none"> <li>• Special purpose registers hold the status of a program whereas general purpose registers hold the temporary data while performing operations.</li> <li>• Special purpose registers are specialised for a specific use, whereas general purpose registers are used for any purpose.</li> <li>• General purpose registers can be used by most instructions, whereas special purpose can only be used by certain instruction</li> </ul>	2										
9(b)	<p><b>1 mark</b> for each correct answer (shaded cells)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">Step</th> <th style="text-align: center; padding: 5px;">Description</th> </tr> </thead> <tbody> <tr> <td style="background-color: #f2f2f2; text-align: center; padding: 5px;">PC <math>\leftarrow</math> [PC] + 1</td> <td style="text-align: center; padding: 5px;">The address in PC is incremented.</td> </tr> <tr> <td style="background-color: #f2f2f2; text-align: center; padding: 5px;">MDR <math>\leftarrow</math> [MAR]</td> <td style="text-align: center; padding: 5px;">The <b>data in the location</b> pointed to by the MAR is copied to the MDR.</td> </tr> <tr> <td style="background-color: #f2f2f2; text-align: center; padding: 5px;">MAR <math>\leftarrow</math> [PC]</td> <td style="text-align: center; padding: 5px;">The <b>contents</b> of PC are copied to the MAR.</td> </tr> <tr> <td style="background-color: #f2f2f2; text-align: center; padding: 5px;">CIR <math>\leftarrow</math> [MDR]</td> <td style="text-align: center; padding: 5px;">The contents of MDR are copied into CIR.</td> </tr> </tbody> </table>	Step	Description	PC $\leftarrow$ [PC] + 1	The address in PC is incremented.	MDR $\leftarrow$ [MAR]	The <b>data in the location</b> pointed to by the MAR is copied to the MDR.	MAR $\leftarrow$ [PC]	The <b>contents</b> of PC are copied to the MAR.	CIR $\leftarrow$ [MDR]	The contents of MDR are copied into CIR.	4
Step	Description											
PC $\leftarrow$ [PC] + 1	The address in PC is incremented.											
MDR $\leftarrow$ [MAR]	The <b>data in the location</b> pointed to by the MAR is copied to the MDR.											
MAR $\leftarrow$ [PC]	The <b>contents</b> of PC are copied to the MAR.											
CIR $\leftarrow$ [MDR]	The contents of MDR are copied into CIR.											

9(c)	<b>1 mark</b> for an appropriate example.  Examples: <ul style="list-style-type: none"><li>• Division by zero</li><li>• Runtime error</li><li>• Out of memory bounds</li><li>• Program requesting an external device / input</li><li>• Buffer overflow</li></ul>	1
9(d)	<b>1 mark</b> for each letter in the correct position (2, 4 and 6)  D A B	3

## Answer 18

Question	Answer	Marks
6(b)	<b>1 mark</b> for correctly naming register, <b>1 mark</b> for appropriate role <ul style="list-style-type: none"><li>• Program counter // PC</li><li>• Stores the address of the next instruction to be fetched</li><li>• Memory address register // MAR</li><li>• Stores the address where data/instruction is to be read from or saved to</li><li>• Memory data register // MDR</li><li>• Stores data that is about to be written to memory // Stores data that has just been read from memory</li><li>• Current instruction register // CIR</li><li>• Stores the instruction that is currently being decoded/executed</li></ul>	4

## Answer 19

2(c)(iii)	<b>1 mark</b> for benefit, <b>1 mark</b> for expansion for <b>max 2</b> benefits <ul style="list-style-type: none"><li>• Fast data transfer</li><li>• ... useful when transferring large files such as video files</li><li>• Automatic connection / plug-and-play</li><li>• ... so usually there is no need to install separate device drivers</li><li>• USB is a standard adopted by many manufacturers</li><li>• ... meaning all new computers will be equipped with USB ports</li><li>• Devices may be powered or charged through USB</li><li>• ... so external devices may be charged while working at the computer</li></ul>	4
-----------	---	---

## Answer 20

2(c)(i)	<b>1 mark</b> <ul style="list-style-type: none"><li>• To provide the connection to peripheral devices</li><li>• To provide an interface between the computer and other devices</li></ul>	1
2(c)(ii)	<b>1 mark</b> <ul style="list-style-type: none"><li>• USB</li><li>• HDMI</li><li>• SCSI</li><li>• Firewire</li><li>• Ethernet</li><li>• Any other acceptable port</li></ul>	1

## Answer 21

Question	Answer	Marks
3(a)	<b>1 mark</b> for each error <b>and</b> correction <ul style="list-style-type: none"><li>• Line 02 should be +1 not -1 // PC ← [PC] + 1</li><li>• Line 03 should be double brackets around MAR // MDR ← [[MAR]]</li><li>• Line 04 should be MDR not MAR // CIR ← [MDR]</li></ul>	3
3(b)	<b>1 mark</b> for each group to <b>max. 2</b> <ul style="list-style-type: none"><li>• Data movement</li><li>• Arithmetic operations</li><li>• (Unconditional and conditional) jump instructions</li><li>• Compare instructions</li><li>• Modes of addressing</li></ul>	2
3(c)	<b>1 mark</b> per bullet <ul style="list-style-type: none"><li>• Storing 0 in 401 (line 51)</li><li>• Loading memory location 300, value 2 to ACC (line 52)</li><li>• Adding 64 to ACC to give 66 (line 55)</li><li>• Outputting B (line 56)</li><li>• Load 0 (line 57), increment ACC (line 58) and store 1 in 401 (line 59)</li><li>• Incrementing IX (line 60)</li><li>• 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)</li><li>• Load 0 (line 52) <u>and end</u></li></ul>	8

## Answer 22

Question	Answer	Marks
4(c)(i)	<p><b>1 mark</b> for hardware interrupt <b>1 mark</b> for software interrupt</p> <p>For example:</p> <p>Hardware interrupt</p> <ul style="list-style-type: none"><li>o Printer out of paper</li><li>o No CD in drive</li></ul> <p>Software interrupt</p> <ul style="list-style-type: none"><li>o A running program needs input</li><li>o Runtime error, e.g. division by zero</li></ul>	<b>2</b>
4(c)(ii)	<p><b>1 mark</b> per bullet to <b>max 5</b></p> <ul style="list-style-type: none"><li>o At the start / end of each fetch-execute cycle the processor checks for interrupt(s)</li><li>o Check if an interrupt flag is set // Check if bit set in interrupt register</li><li>o Processor identifies source of interrupt</li><li>o Processor checks priority of interrupt</li><li>o If interrupt priority is high enough // Lower priority interrupts are disabled</li><li>o Processor saves current contents of registers // saves current job on stack</li><li>o Processor calls interrupt handler / Interrupt Service Routine (ISR)</li><li>o Address of ISR is loaded into Program Counter (PC)</li><li>o When servicing of interrupt complete, processor restores registers // job from stack is restored</li><li>o Lower priority interrupts are re-enabled</li><li>o Processor continues with next F-E cycle</li></ul>	<b>5</b>

## Answer 23

Question	Answer	Marks												
1(a)	<p><b>1 mark for each correct line</b></p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding-bottom: 10px;">Register</th> <th style="text-align: center; padding-bottom: 10px;">Description</th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; padding: 5px; width: 30%;">Current Instruction Register</td> <td style="border: 1px solid black; padding: 5px; width: 60%;">Stores the data that has just been read from memory, or is about to be written to memory</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">Memory Address Register</td> <td style="border: 1px solid black; padding: 5px;">Stores the instruction that is being decoded and executed</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">Program Counter</td> <td style="border: 1px solid black; padding: 5px;">Stores the address of the input device from which the processor accesses the instruction</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">Memory Data Register</td> <td style="border: 1px solid black; padding: 5px;">Stores the address of the next instruction to be read</td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 5px;">Stores the address of the memory location about to be written to or read from</td> </tr> </tbody> </table>	Register	Description	Current Instruction Register	Stores the data that has just been read from memory, or is about to be written to memory	Memory Address Register	Stores the instruction that is being decoded and executed	Program Counter	Stores the address of the input device from which the processor accesses the instruction	Memory Data Register	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
Register	Description													
Current Instruction Register	Stores the data that has just been read from memory, or is about to be written to memory													
Memory Address Register	Stores the instruction that is being decoded and executed													
Program Counter	Stores the address of the input device from which the processor accesses the instruction													
Memory Data Register	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													
1(b)(i)	<p><b>1 mark for naming, 1 mark for purpose for each bus</b></p> <ul style="list-style-type: none"> <li>• Data bus</li> <li>• Carries data between the processor and memory / carries data that is currently being processed.</li>   <li>• Control bus</li> <li>• Transmits signals between the control unit and the other components</li> </ul>	4												
1(b)(ii)	<p>Significant increase in the number of directly addressed memory locations // increases the number of directly addressable memory locations from <math>2^{16}</math> to <math>2^{32}</math></p>	1												

## Answer 24

Question	Answer	Marks
8(a)	<p><b>1 mark per bullet to max 2</b></p> <ul style="list-style-type: none"> <li>• Small piece / word of (fast) memory</li> <li>• Part of the processor</li> <li>• Temporary storage of data</li> <li>• Data is about to be / has been processed</li> </ul>	2
8(b)(i)	<p><b>1 mark per bullet to max 2</b></p> <ul style="list-style-type: none"> <li>• Stores / holds data / instruction when fetched from memory</li> <li>• Stores / holds data which is being written to memory</li> <li>• The location accessed is the address held in the Memory Address Register (MAR)</li> </ul>	2

Question	Answer	Marks
8(b)(ii)	<p><b>1 mark per bullet to max 2</b></p> <ul style="list-style-type: none"> <li>• Current Instruction Register (CIR)</li> <li>• Memory Address Register (MAR)</li> <li>• Program Counter (PC)</li> <li>• Accumulator (ACC)</li> <li>• Index Register (IX)</li> <li>• Status Register</li> <li>• Interrupt Register</li> </ul>	2
8(c)(i)	135	1
8(c)(ii)	87	1
8(c)(iii)	-121	1

## Answer 25

Question	Answer	Marks
3(a)	<p><b>1 mark</b> per bullet to <b>max 2</b> for each group</p> <ul style="list-style-type: none"> <li>• <b>ALU</b> performs arithmetic operations</li> <li>• And logical operations / comparisons</li> <li>• <b>Control Unit</b> sends / receives signals</li> <li>• Synchronises operations</li> <li>• to control operations // execution of instructions</li> <li>• Accept by example e.g. Input output // flow of data</li> </ul>	<b>4</b>
3(b)	<p><b>1 mark</b> per bullet to <b>max 2</b> for each group</p> <ul style="list-style-type: none"> <li>• <b>Status Register</b> is interpreted as independent bits / flags</li> <li>• Each flag is set depending on an event</li> <li>• An example: addition overflow / result of operation is zero etc.</li> <li>• <b>Program Counter</b> stores the <u>address</u> of the <u>next</u> instruction to be fetched</li> </ul>	<b>4</b>
3(c)(i)	193	<b>1</b>
3(c)(ii)	C1	<b>1</b>
3(c)(iii)	- 63	<b>1</b>
3(c)(iv)	The <u>first 4 bits / first nibble</u> (would give 12 which) is <u>&gt; 9 / 2 digits</u> (which is not valid for BCD)	<b>1</b>

## Answer 26

Question	Answer	Marks										
4(a)	<p><b>1 mark per correct line, max 3</b></p> <table border="1"> <thead> <tr> <th>Line number of error</th><th>Correct notation</th></tr> </thead> <tbody> <tr> <td>1</td><td><math>\text{MAR} \leftarrow [\text{PC}]</math></td></tr> <tr> <td>3</td><td><math>\text{MDR} \leftarrow [ \text{[MAR]} ]</math></td></tr> <tr> <td>4</td><td><math>\text{CIR} \leftarrow [\text{MDR}]</math></td></tr> <tr> <td>2</td><td><math>\text{PC} \leftarrow [\text{PC}] + 1</math></td></tr> </tbody> </table>	Line number of error	Correct notation	1	$\text{MAR} \leftarrow [\text{PC}]$	3	$\text{MDR} \leftarrow [ \text{[MAR]} ]$	4	$\text{CIR} \leftarrow [\text{MDR}]$	2	$\text{PC} \leftarrow [\text{PC}] + 1$	3
Line number of error	Correct notation											
1	$\text{MAR} \leftarrow [\text{PC}]$											
3	$\text{MDR} \leftarrow [ \text{[MAR]} ]$											
4	$\text{CIR} \leftarrow [\text{MDR}]$											
2	$\text{PC} \leftarrow [\text{PC}] + 1$											
4(b)(i)	<p><b>1 mark for each event to max 3</b></p> <p>For example:</p> <ul style="list-style-type: none"> <li>• Hardware fault // Example of hardware fault</li> <li>• I/O request // Example of I/O request</li> <li>• Program/software error // Example of software error</li> <li>• End of a time-slice</li> </ul>	3										
4(b)(ii)	<p><b>1 mark per bullet point to max 5</b></p> <ul style="list-style-type: none"> <li>• At the end of each fetch–execute cycle the processor checks for interrupt(s)</li> <li>• Check if an interrupt flag is set // Check if bit set in interrupt register</li> <li>• Processor identifies source of interrupt</li> <li>• Processor checks priority of interrupt</li> <li>• If interrupt priority is high enough // Lower priority interrupts are disabled</li> <li>• Processor saves current contents of registers</li> <li>• Processor calls interrupt handler / Interrupt Service Routine (ISR)</li> <li>• Address of ISR is loaded into Program Counter (PC)</li> <li>• When servicing of interrupt complete, processor restores registers</li> <li>• Lower priority interrupts are re-enabled</li> <li>• Processor continues with next F-E cycle</li> </ul>	5										

Question	Answer	Marks												
4(c)	<p>1 mark for 1 correct connection 2 marks for all 3 correct connections</p> <table style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">Bus</td> <td style="width: 33%;">Description</td> <td style="width: 33%;"></td> </tr> <tr> <td>Control Bus</td> <td>Uni-directional (one direction)</td> <td></td> </tr> <tr> <td>Address Bus</td> <td>Bi-directional (two directions)</td> <td></td> </tr> <tr> <td>Data Bus</td> <td></td> <td></td> </tr> </table>	Bus	Description		Control Bus	Uni-directional (one direction)		Address Bus	Bi-directional (two directions)		Data Bus			2
Bus	Description													
Control Bus	Uni-directional (one direction)													
Address Bus	Bi-directional (two directions)													
Data Bus														

## Answer 27

Question	Answer	Marks
6(a)(i)	$\text{MAR} \leftarrow [\text{PC}]$ $\text{PC} \leftarrow \text{PC} + 1$ $\text{MDR} \leftarrow [ \text{[MAR]} ]$ $\text{CIR} \leftarrow [\text{MDR}]$	3
6(a)(ii)	<b>1 mark from:</b> <ul style="list-style-type: none"> <li>The contents of the MAR is an address, it is the contents of that address which is transferred to MDR</li> <li>The contents of the address pointed to by the MAR is transferred to the MDR</li> </ul>	1
6(b)	<b>1 mark per bullet point to max 2</b> <ul style="list-style-type: none"> <li>A signal from a source / device</li> <li>Telling the processor its attention is needed</li> </ul>	2

## Answer 28

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

## Answer 29

Question	Answer	Marks
4(a)	<b>1 Mark</b> for each correct answer A – General purpose registers B – System clock C – ALU E – Control bus F – Address bus	5
4(b)	<b>1 Mark</b> per bullet, max 2 <ul style="list-style-type: none"><li>▫ The clock sends out a number of pulses in a given time interval (clock speed)</li><li>▫ Each processor instruction takes a certain number of clock cycles to execute</li><li>▫ The higher the clock frequency, the shorter the execution time for the instruction</li><li>// Increasing the clock frequency improves performance</li></ul>	2

## Answer 30

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

## Answer 31

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 32

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	Sequence 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 33

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 34

- 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 35

- 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 36

- 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 37

- 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 38

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]