

Objective:

- ☒ Show understanding of **Von Neumann model** and **stored program concept**.
- ☒ Show understanding of purpose and role of registers, including difference between general purpose and special purpose registers. **Special purpose registers** including:
 - Program Counter (PC) • Memory Data Register (MDR) • Memory Address Register (MAR) • Accumulator (ACC) • Index Register (IX) • Current Instruction Register (CIR) • Status Register
- ☒ Show understanding of purpose and roles of **Arithmetic and Logic Unit (ALU)**, Control Unit (CU) and system clock, Immediate Access Store (IAS).
- ☒ Show understanding of how data are transferred between various components of computer system using address bus, data bus and control bus.
- ☒ Show understanding of how factors contribute to **performance** of computer system. Including: • processor type and number of cores • bus width • clock speed • cache
- ☒ Understand how different ports provide connection to peripheral devices Including connection to: • USB • High Definition Multimedia Interface (HDMI) • Video Graphics Array.
- ☒ Describe stages of Fetch-Execute (F-E) cycle. Describe and use 'register transfer' notation.
- ☒ Show understanding of purpose of interrupts Including: • possible causes of interrupts • applications of interrupts • use of an Interrupt service (ISR) handling routine • when interrupts are detected during fetch execute cycle • how interrupts are handled.

Introduction

Early computers were fed data while machines were running. It was not possible to store programs or data.

Stored Program Concept:

John Von Neumann introduced idea of **stored program** computer. Previously data and programs were stored in separate memories. Von Neumann realized that **data** and **programs** are somewhat of **same type** and can use the **same** memory.

A **Stored-Program** digital computer is one that keeps its **programmed instructions**, as well as its data, in read-write, random-access memory (RAM).

Main features of Von Neumann Architecture:

- There is a **processor** - central processing unit (CPU).
- **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.

Components of Processor (CPU)

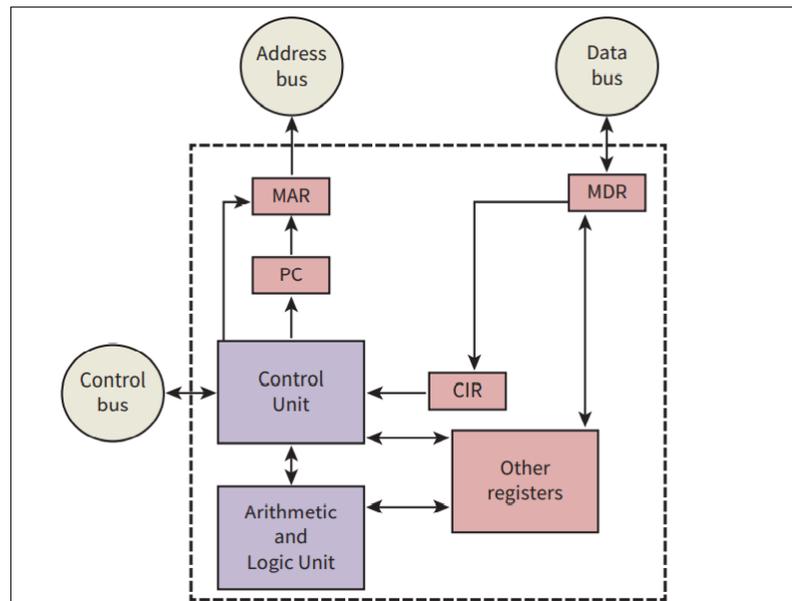
Arithmetic logic Unit

ALU allows required arithmetic or logic operations to be carried out

while a program is being run. It is possible for a computer to have more than one ALU – one will perform **fixed point** operations and other **floating-point** operations. Multiplication and division are carried out by a **sequence** of addition, subtraction and left/right shifting operations. **Accumulator** is a **temporary register** used when carrying out ALU calculations.

Control Unit

Control unit (CU) reads an instruction from **memory** (address of location where instruction can be found is stored in program counter). This instruction is then **interpreted**. During that process, signals are generated along **control bus** to tell other components in computer what to do. CU ensures **synchronisation** of **data** flow and **program** instructions throughout computer.



System Clock

System clock is used to produce **Timing signals** on **control bus** to ensure this vital synchronisation takes place. Without the clock the computer would simply crash.

There are two clocks.

- **Internal Clock** that controls the cycles of activity **within** processor.
- **System Clock** that controls activities **outside** processor. CPU will have a defined frequency for its clock cycle, which is usually referred to as **clock speed**. Frequency defines minimum period of time that separates successive activities within system.

Immediate Access Store

IAS is another name for primary (RAM) memory. IAS holds all data and programs that processor needs to access. CPU takes data and programs held in **backing store** and puts them into IAS temporarily. This is done because read/write operations carried out using IAS are considerably faster than read/write operations to **backing store**. Consequently, any key data needed by an application will be stored temporarily in IAS to speed up operations.

Registers

One of most fundamental components of Von Neumann system is **register**.

Registers can be **General Purpose** or **Special Purpose**.

- **General Purpose** registers hold data that is frequently used by CPU or can be used by programmer when addressing CPU directly.

Example:

Accumulator is one of general purpose registers. A value is stored in Accumulator that is to be used by ALU for execution of an instruction. ALU can then store a different value in Accumulator after execution of instruction.

- **Special Purpose** registers have a **specific function** within CPU and hold **program state**.

Memory Address Register (MAR)	MAR holds memory addresses of data and instructions . MAR is used to access data and instructions from memory during execution of an instruction. Suppose CPU wants to store some data in memory or to read data from memory. It places address of required memory location in MAR.
Memory Data Register MDR (MBR)	MDR is register that contains data to be stored in computer storage (e.g. RAM), or the data after a fetch from the computer storage. It acts like a buffer and holds anything that is copied from memory ready for the processor to use it.
Program Counter PC	It keeps track of next memory address of the instruction that is to be executed once the execution of the current instruction is completed.
Current Instruction Register CIR	Stores the current instruction while it is being decoded and executed, having been fetched from memory.
Index register IX	Known as base register . An index register in a computer's CPU is a processor register used for modifying operand addresses during the run of a program.
Status register SR	Status register contains information about state of processor. Status register can be set, equal to number 1, or cleared, equal to number 0.

Status Register is used when an instruction requires some form of arithmetic or logic processing. Each bit is known as a **flag**.

Most Systems have following Four Flags.

- **Carry Flag (C)** is set to 1 if there is a CARRY following an addition operation.
- **Negative Flag (N)** is set to 1 if the result of a calculation yields a NEGATIVE value.
- **Overflow Flag (V)** is set to 1 if an arithmetic operation results in an OVERFLOW.
- **Zero Flag (Z)** is set to 1 if the result of an arithmetic or logic operation is ZERO.

System Bus

System buses are used in computers as **parallel** transmission component. Each wire in bus transmits one bit of data.

Following are three common buses used in Von Neumann architecture;

Address Bus

Address bus carries **addresses** throughout computer system between CPU and memory. Address bus is **unidirectional** which mean bits can travel in one direction only. This prevents addresses being carried back to CPU, which would be undesirable.

Width of a bus is important. Wider the bus, more memory locations which can be directly

addressed at any given time.

Example: Bus of width 16 bits can address 2^{16} (65 536) memory locations, whereas a bus width of 32 bits allows 4 294 967 296 memory locations to be simultaneously addressed.

Data Bus

Data bus is **bidirectional** which means it allows **data** to be sent in both directions along bus. This means data can be carried from CPU to memory (and vice versa) as well as to and from input/output devices. It is important to point out that data can be an address, an instruction or a numerical value.

Width of data bus is important. Wider the bus, the larger the word length that can be transported. (A **word** is a group of bits which can be regarded as a single unit, for example, 16-bit, 32-bit or 64-bit word lengths are most common). Larger word lengths can improve computer's overall performance.

Control Bus

Control bus is **bidirectional**. It carries **signals** from the CU to all other computer components. It is usually **8-bits wide** since it only carries **control signals**.

Factors Contributing To System Performance

Clock Cycle **synchronises** all computer operations. Control bus transmits **timing signals**, ensuring everything is fully synchronised. By increasing **clock speed**, processing speed of computer is also increased (typical value is 3.5 GHz – which means 3.5 billion clock cycles a second). Although **speed of computer** may have been increased, it is not possible to say that a computer's overall performance is necessarily increased by using a higher clock speed.

Four other factors need to be considered.

Bus Width:

Width of address bus and data bus can affect computer performance.

Clock speed:

Each instruction is executed on a clock pulse. 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.

Faster the clock speed the more instructions can be run per second.

Clock speed can be changed by altering BIOS settings. Using clock speed higher than computer was designed can lead to **unsynchronised operations** and computer would **crash** leading to **unreliable** performance.

Cache Memory:

Use of cache memory can improve processor performance. Cache is similar to RAM in that its contents are lost when power is turned off. Cache uses **SRAM** whereas most computers use DRAM for main memory. Therefore, cache

memories will have **faster access times**, since there is no need to keep **refreshing**, which **slows down** access time.

When a processor reads memory, it first checks out cache and then moves on to main memory if required data is not there. **Cache memory** stores frequently used instructions and data that need to be accessed faster. This improves **processor performance**.

Use of different number of Cores:

One core is made up of an ALU, a CU and registers. Use of a different number of cores can improve computer performance. Each core processes **one instruction** per clock pulse. More cores mean that **sequences of instructions** can be split between them and so more than one instruction is executed per **clock pulse**. More sequences of instructions can be run at same time. More cores decreases time taken to complete task.

Dual core computer CPU is made up of two cores and quad core CPU is made up of four cores. Idea of using more cores alleviates need to increase clock speeds. However, doubling number of cores does not necessarily double computer's performance since CPU to communicate with each core; this will reduce overall performance.

Computer Ports

Input and output devices are connected to a computer via **ports**. Interaction of ports with connected input and output is controlled by **control unit**.

USB Ports:

Universal Serial Bus (USB) is an **asynchronous serial** data transmission method. It is standard method for transferring data between a computer and a number of devices.

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;

- ❖ Computer **automatically** detects that a device is present. This is due to small change in voltage level on data signal wires in the cable.
- ❖ Device is **automatically recognised**, and appropriate device driver is loaded up so that computer and device can communicate effectively.
- ❖ If a new device is detected, computer will look for **device driver** which matches the device. If this is not available, user is prompted to download the appropriate software.

Pros of USB System:

- ❖ Devices plugged into computer are automatically detected and device drivers are automatically loaded up.
- ❖ Connectors can only fit one way, which prevents incorrect connections being made.

- ❖ USB has become industry standard, which means that considerable support is available to users.
- ❖ Several different data transmission rates are supported.
- ❖ Newer USB standards are backward compatible with older USB standards.

Cons of USB System:

- ❖ Present transmission rate is limited to less than 500 megabits per second.
- ❖ Maximum cable length is presently about five metres
- ❖ Older USB standard (such as 1.1) may not be supported in the near future.

HDMI Ports:

High-definition multimedia interface (HDMI) ports allow output (both audio and visual) from a computer to HDMI-enabled device. They support high-definition signals.

HDMI was introduced as a digital replacement for older Video Graphics Array (VGA) analogue system. Modern HD televisions have following features, which are making VGA a redundant technology:

- ❖ They use a widescreen format (16:9 aspect ratio).
- ❖ Screens use a greater number of pixels (typically 1920 × 1080).
- ❖ Screens have a faster refresh rate (such as 120 Hz or 120 frames a second).
- ❖ Range of colours is extremely large (Approx four million different colour variations).

Modern HD televisions require more data, which has to be received at a much **faster rate** than with older televisions. HDMI **increases bandwidth**, making it possible to supply necessary data for **high quality** sound and visual effects.

HDMI can afford some protection against **piracy** since it uses **high-bandwidth digital copy protection** (HDCP). HDCP uses a type of authentication protocol.

Example, Blu-ray player will check **authentication key** of device it is sending data to HD television. If key can be authenticated, then **handshaking** takes place and Blu-ray can start to transmit data to connected device.

Pros of HDMI	Cons of HDMI
Current standard for modern televisions and monitors.	not a very robust connection (easy to break connection when simply moving device)
Allows for a very fast data transfer rate	Limited cable length to retain good signal
Improved security (helps prevent piracy)	There are currently five cable/connection standards
Supports modern digital systems	

VGA Ports:

Video Graphics Array (VGA) was introduced in 1980s. If screen is needed to display a video, VGA port is not suitable because it does not transmit audio component. VGA supports

640 × 480 pixel resolution on a television screen. It can handle a refresh rate of up to 60 Hz (60 frames a second).

Pros of VGA	Cons of VGA
Simpler technology	Old out-dated analogue technology
Only one standard available	Easy to bend pins when making connections
It is easy to split the signal and connect a number of devices from one source	Cables must be of a very high grade to ensure good undistorted signal
Connection is secure	

Fetch Decode Execute Cycle

Steps of Fetch Execute Cycle.

1. Load address that is in program counter (PC) into memory address register (MAR).
2. Increment PC by 1.
3. Load instruction that is in memory address given by the MAR into the MDR.
4. Load the instruction that is now in the MDR into the current instruction register (CIR).
5. Decode the instruction that is in the CIR.
6. If the instruction is a jump instruction then ;
 - a. Load the address part of the instruction into PC
 - b. Reset by going to step 1.
7. Execute the instruction.
8. Reset by going to step 1.

Steps 1 to 4 are the fetch part of the cycle.
Steps 5 are decode part of the cycle.
Steps 6(a) and 7 are the execute part of cycle.

Register Transfer Notation (RTN):

```
MAR ← [PC]      contents of PC copied into MAR
PC ← [PC] + 1  PC is incremented by 1
MDR ← [[MAR]]  data stored at address shown in MAR is
                copied into MDR
CIR ← [MDR]    contents of MDR copied into CIR
```

Double brackets are used in the third line because it is not MAR contents being copied into MDR but it is the data stored at the address shown in MAR that is being copied to MDR.

Interrupts

Interrupt is a **signal** sent from a device or from software to the processor. This will cause processor to **temporarily stop** what it is doing and service the interrupt.

Reason of Interrupts Signal:

- ❖ **Timing Signal**
- ❖ **Input/Output Processes** (a disk drive is ready to receive more data)
- ❖ **Hardware fault** (an error has occurred such as a paper jam in a printer)
- ❖ **User Interaction** (User pressed key to interrupt current process)

- ❖ **Software Error** that cannot be ignored (if an .exe file could not be found to initiate execution of program OR an attempt to divide by zero).

Actions of processor when an interrupt is detected.

9608 May 15

- ❖ At the end of each fetch–execute cycle the processor checks for interrupt(s)
- ❖ Check if an interrupt flag is set means check if any bit set in interrupt register.
- ❖ Processor identifies source of interrupt and checks priority of interrupt.
- ❖ If priority of interrupt is lower than current process then current process is continue. If interrupt priority is high enough lower priority interrupts are disabled.
- ❖ Contents of general purpose and special purpose registers are saved in its PCB on stack.
- ❖ Processor calls interrupt handler i.e. Interrupt Service Routine (ISR)
- ❖ Address of ISR is loaded into Program Counter (PC)
- ❖ ISR code is run.
- ❖ When servicing of interrupt complete, processor restores registers with task in stack.
- ❖ All lower priority interrupts are re-enabled
- ❖ Processor continues with next Fetch–Execute cycle

Interrupts allow computers to carry out **many tasks** or to have several windows open at same time. An example would be downloading a file from internet at same time as listening to some music from the computer library.
