Section 04.1 - Central Processing Unit (CPU) Architecture

Layer 3: ISA Machine

Syllabus Content Section 04: Processor Fundamentals

S04.1.1 Show understanding of the basic Von Neumann model for a computer system and the stored program concept

---

Von neumann architecture only has 3 main parts

1. processor
2. memory
3. I/O
   they connect with each other by Buses

- `Address bus`: transmits an address between the processor and memory. (One way)
- `Data bus`: carries data between the processor and memory. (Two way)
- `Control Bus`: transmits signals between the control unit and the other components. (Two way)

`Bus width`: determines the no. of bits that can be simultaneously transferred
Greater bus width, significant increase in the number of directly addressed memory locations
`Clock speed`: (MHz/GHz) the no. of cycles that are performed by the CPU per second.

> Faster clock speed means more operations executed per unit of time
> however faster clock speed causes processor to heat up

`Clock tick` = `Clock cycle`
`Clock rate:` (MHz/GHz) the speed at which a micro-processor execute instructions

S04.1.2 Show understanding of the purpose and role of registers, including the difference between general purpose and special purpose registers

Special purpose registers including:

- Program Counter (PC)
- Memory Data Register (MDR)
- Memory Address Register (MAR)
- The Accumulator (ACC)
- Index Register (IX)
- Current Instruction Register (CIR)
- Status Register

Abbr.	Full Name	Function
PC	Program counter	contains the location of the instruction that is to be executed next.
MDR	Memory data register	stores the data involved in read/write operation
MAR	Memory address register	stores the address of location for read/write operation
ACC	Accumulator	(single general-purpose register) stores the result of arithmetic and logic operations performed by the processor
IX	Index Register	The Index Register is used if we use Index Addressing Modes
CIR	Current instruction register	holds the instruction that is to be executed
SR	Status Register	interpreted as independent bits; Each bit is set depending on an event

S04.1.3 Show understanding of the purpose and roles of the Arithmetic

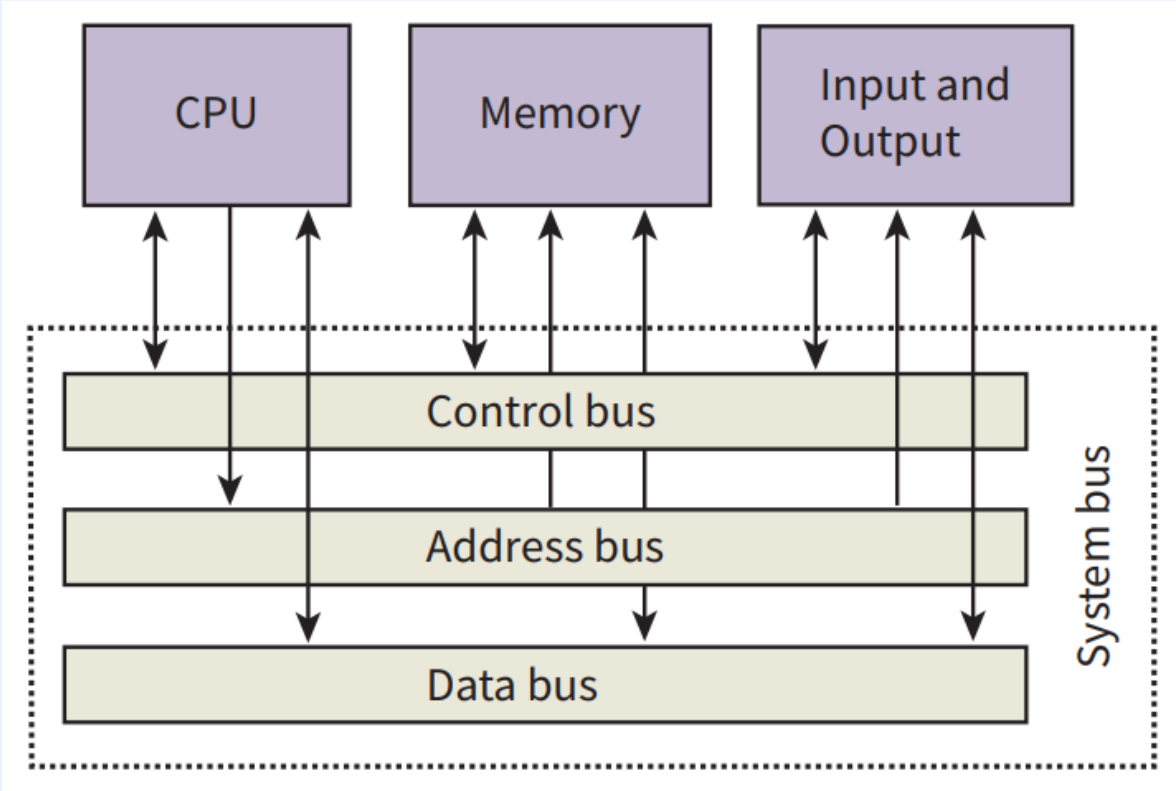
and Logic Unit (ALU), Control Unit (CU) and system clock, Immediate Access Store (IAS)

Register	purpose
ALU	to do all the math calculation(Add, Subtraction, AND, OR, .etc)
CU	manages what everything else in the CPU does. It controls the CPU
system clock	CPU clock, if your clock is 1Hz, it means it does 1 pulse per second
IAS	the place where you hold all the data that you are currently using, It's a group name for registers

S04.1.4 Show understanding of how data are transferred between various components of the computer system using the address bus, data bus and control bus

1. Address Bus
   Processor says the physical address it wants to use
2. Data Bus
   Data is sent or received on this
3. Control Bus
   Used to make sure timing is good and tells other parts what the processor is

doing



S04.1.5 Show understanding of how factors contribute to the performance of the computer system

Including:

- processor type and number of cores
- the bus width
- clock speed
- cache memory

> processor type and number of cores

If you have quad core, then its one CPU chip but behaves like 4 (quad) CPU's because it has 4 physical parts to it, its just 4 parts on one chip.
It means it can do 4 things at the same time.

> the bus width

The bus is just a wire (or usually printed on a motherboard)
Bus width is how many physical wires are connected.

If there are 16 wires connected then your bus width is 16. If 32 wires then your bus width is 32.
If your bus width is 32, it means that you can send 32 bits of data at the same time (parallel)

> clock speed

So if your clock is 1Hz, 1 Hertz, it means it does 1 pulse per second
1MHz, 1 Megahertz means 1 million things per second
1GHz, 1 Gigahertz means 1 billion things per second.

> cache memory

This is a form of DRAM – Dynamic RAM. But its much smaller and much faster.

Cache memory has 3 levels.
Each level describes how physically close the cache memory is to the CPU

`Level 1 (L1 Cache)` - Fastest. Built inside the CPU. Small capacity. Sometimes called Primary cache

`Level 2 (L2 Cache)` – Fast. Can be built in CPU or sometimes built on a separate chip. Has its own bus so it doesn't slow down even if the system bus is slow. Has more capacity then L1 cache. Sometimes called secondary cache

`Level 3 (L3 cache)` – This type of cache is faster than standard RAM but it used to help L1 and L2 cache to hold data. It can pass data to L1 and L2 quicker than RAM.

So the more L1, L2 and L3 cache you have the more you can store in this faster type of memory and the fact it uses its own bus also means it won't slow down even if everything else does.

S04.1.6 Understand how different ports provide connection to peripheral devices

Including connection to:

- Universal Serial Bus (USB)
- High Definition Multimedia Interface (HDMI)
- Video Graphics Array (VGA)

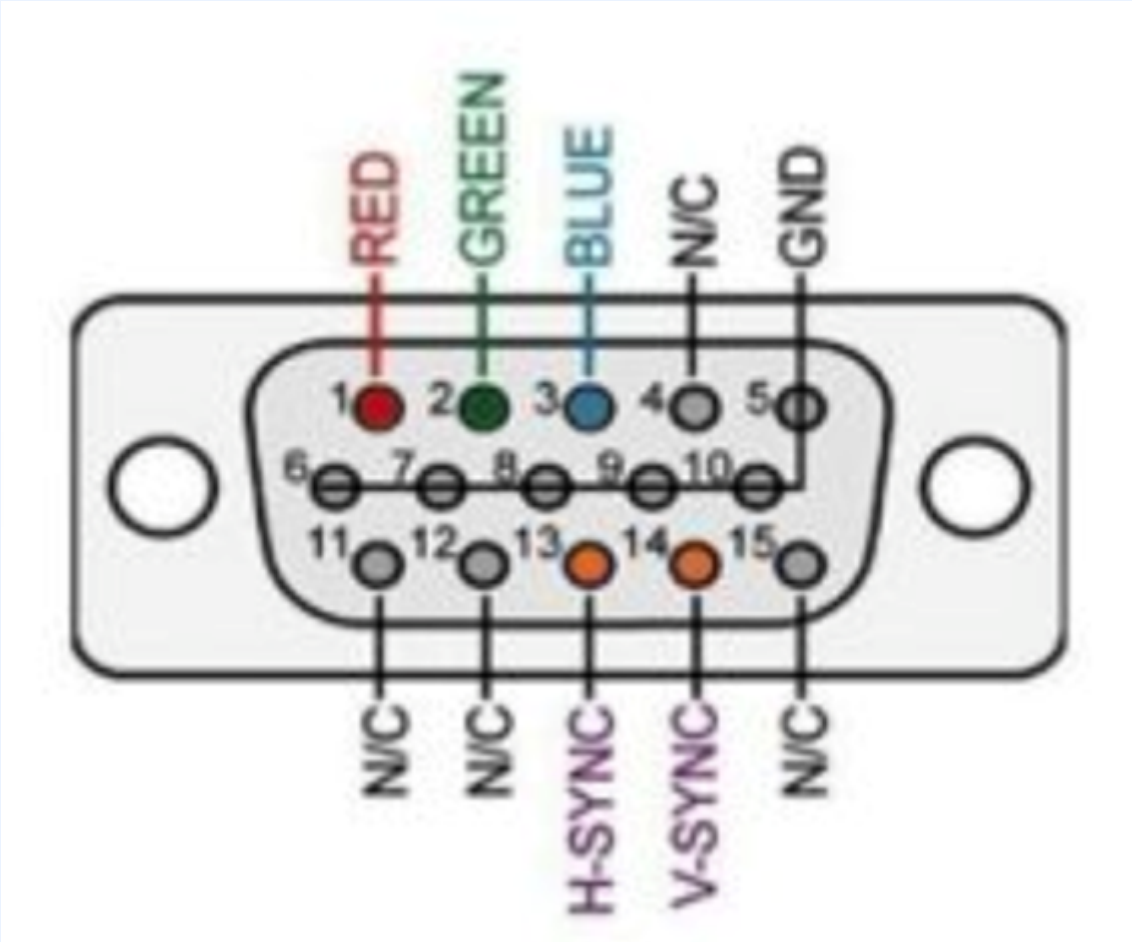
> USB
> Very common, Digital, Many variants
> Can transfer power and data

At least 4 pins.
2 pins are for power
1 pin for data in
1 pin for data out

> HDMI

Newer, Digital, Uses 19 pins
Video / Pictures and sound
Three variants: Full; Mini and Micro
Because its digital it makes no difference what material the cable is made from. But people still try to sell Gold HDMI for better conductivity because Gold VGA cables did make a *very small* difference

> VGA



Pin	What it does

Pin	What it does
1	Outputs Red
2	Outputs Green
3	Outputs Blue
4	Colour Monitor or Mono Monitor (ID2)
5	Ground / Earth
6	Red Ground
7	Green Ground
8	Blue Ground
9	Voltage +-5V
10	Synchronize Grounds
11	Another monitor bit (ID0)
12	Another monitor bit (ID1)
13	Horizontal Sync (HSYNC)
14	Vertical Sync (VSYNC)
15	Another monitor bit (ID3)

- Uses screws to keep the cable in place
- Has 15 pins (usually)
- Can send video signals / picture signals
- Is only used now because some old hardware still uses it
- Old
- Cheap
- Analogue
- Degrades over long length

🖉 S04.1.7 Describe the stages of the Fetch-Execute (F-E) cycle

Describe and use ‘register transfer’ notation to describe the F–E cycle

---

Fetch -> Decode -> Execute

1. MAR <- [PC]
2. PC <- PC + 1
3. [MDR] <- [[MAR]]
4. [CIR] <- [MDR]
5. [CIR] Decode and Execute

Step	Fetch execute cycle steps	Simplified description
1	The PC contains the address of the memory location that has the next instruction which has to be fetched	PC has address of next instruction
2	This address is then copied from the PC to the MAR via the address bus	PC copied to the MAR
3	The contents (instruction) at the memory location (address) contained in MAR are then copied into the MDR	Lookup MAR and get contents. Copy contents into the MDR
4	The contents (instruction) in the MDR is then copied and placed into the CIR	Copy MDR contents into the CIR
5	The value in the PC is then incremented by 1 so that it now points to the next instruction which has to be fetched	PC is then incremented by 1
6	The instruction is finally decoded and then executed by sending out signals (via control bus) to the various components of the computer	The instruction is decoded and then executed
7	Repeat	

S04.1.8 Show understanding of the purpose of interrupts

Including:

- possible causes of interrupts
- applications of interrupts

- use of an Interrupt service (ISR) handling routine
- when interrupts are detected during the fetch-execute cycle
- how interrupts are handled

Interrupt = When you stop Fetch-Execute Cycle to do something more important

- A signal from a source
- Telling the processor its attraction is needed

> Causes

1. Hardware interrupts
   This is when a signal from hardware is given to interrupt the current process. An example is if you press the Escape key when running code or the Mute key when listening to audio

2. Software interrupts
   This is when software decides there is to be an interrupt. There are two main types of software interrupt
   a) Normal Interrupt
   This is when an instruction in the code says there should be an interrupt
   b) Exception
   This is when the interrupt is unplanned and an error happened.

3. Timer Interrupts
   Always happens at a certain time. Example to refresh your display screen

4. Input/Output interrupts
   Most common, you press a key on the keyboard, it sends a signal to the CPU, the CPU stops what it's doing and does what you pressed then returns to what it was doing before.

> Interrupt Service Handling Routine - ISR

The ISR is actually a control line dedicated for interrupts

The program counter is pointing to the next instruction as normal.
An interrupt happens

Now the program counter must stop pointing to what would have happened as normal and now it will point to the first instruction stored in the ISR