❒ Modern computer architecture is accredited to John Von Neumann

A **von Neumann machine** is a model created by **John von Neumann** for a computing machine that uses a single storage structure to hold both the set of instructions (how to perform the computation) and the data required or generated by the computation. Most modern computers use this von Neumann architecture. Computers using this architecture are said to be "von Neumann machines."

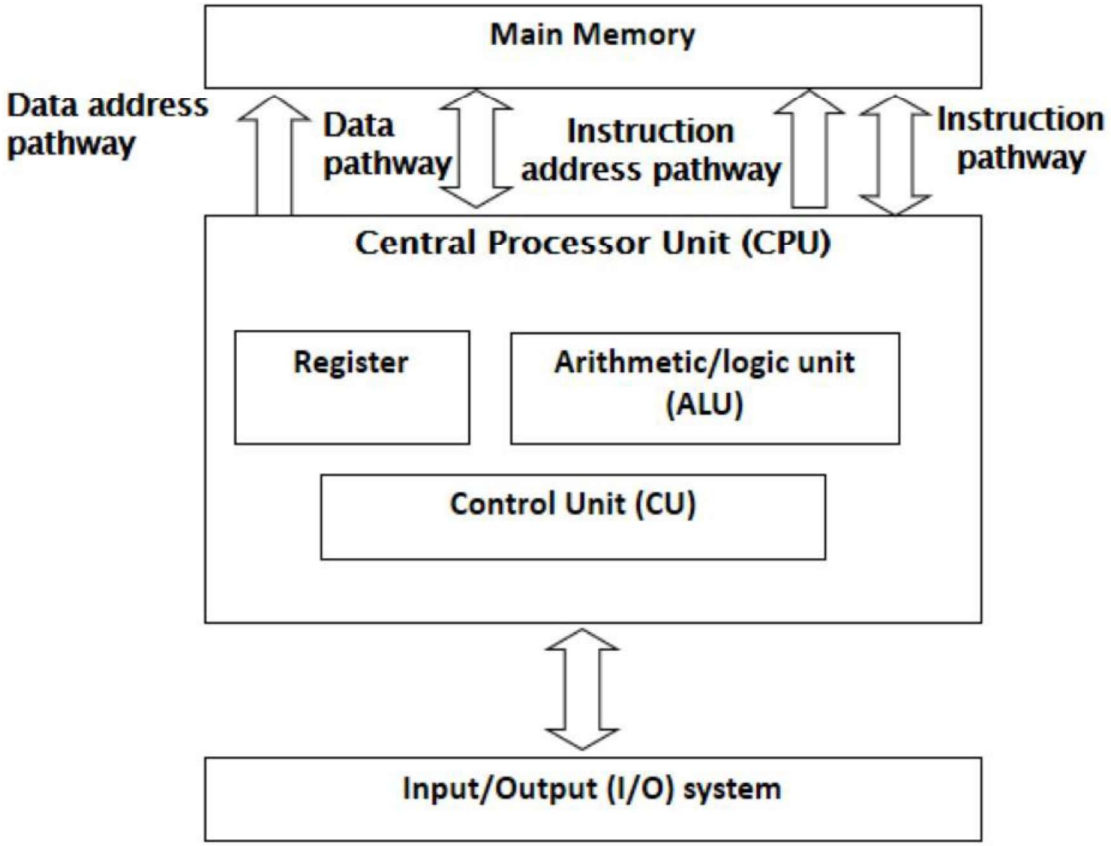
Von-Neumann machine has three basic hardware subsystems

- Central Processing Unit (CPU)
- Main memory
- input/output (I/O)

The main memory holds the program that controls the computer's operation and the computer that can manipulate. It carries out the instructions sequentially. The CPU executes one program at a time.

Conventional Von-Neumann machines provide one pathway for address and a second pathway for data & instructions.

The term Harvard architecture is a class of Von Neumann machines similar to conventional Von-Neumann computers except that they provide independent pathway for data address, data, instruction address, & instruction.



Harvard architecture

In a computer with Harvard architecture, the CPU can read both an instruction and data from memory at the same time. A computer with Harvard architecture can be faster because it is able to fetch the next instruction at the same time it completes the current instruction. Speed is gained at the expense of more complex electrical circuitry.

SYSTEM BUS

The CPU , memory and I/O are normally connected by three groups of connections, each called a **bus:** ***data bus, address bus and control bus.***

1. **Data Bus:** These are lines used by the information that is being moved.

When communicating with memory this information may either instruction or data.

In I/O devices this information may be data, device status or commands or interrupt information.

- Data can be transferred in both directions on the data bus (bidirectional).
- Internal data bus: within the CPU itself.
- External data bus: Between CPU and other components (as RAM, I/O devices,etc.).
- The number of data lines in a bus determines the number of bits that can be transferred simultaneously.
- A microcomputer with n data lines is normally called an n-bit microcomputer.

ex: 8-bits microcomputer can transfer one byte in each bus cycle.

16-bits microcomputer can transfer 2 byte (word) in each bus cycle. (8086 mp).

SYSTEM BUS

EX1: How many bus cycle need to transfer data of size 64bits , if data bus size 16 line?

sol:

$$no\ of\ cycle = \frac{no.\ of\ bits\ (data)}{\text{no. of lines in dada bus}} = \frac{64}{16} = 4\ \text{bus cycle}$$

EX2: How many bus cycle need to transfer 128 byte of data in 32-bits microcomputer ?

$$\because \text{data bus size} = 32\,\text{bits} = 4\,\text{byte can be transfer in each cycle}$$

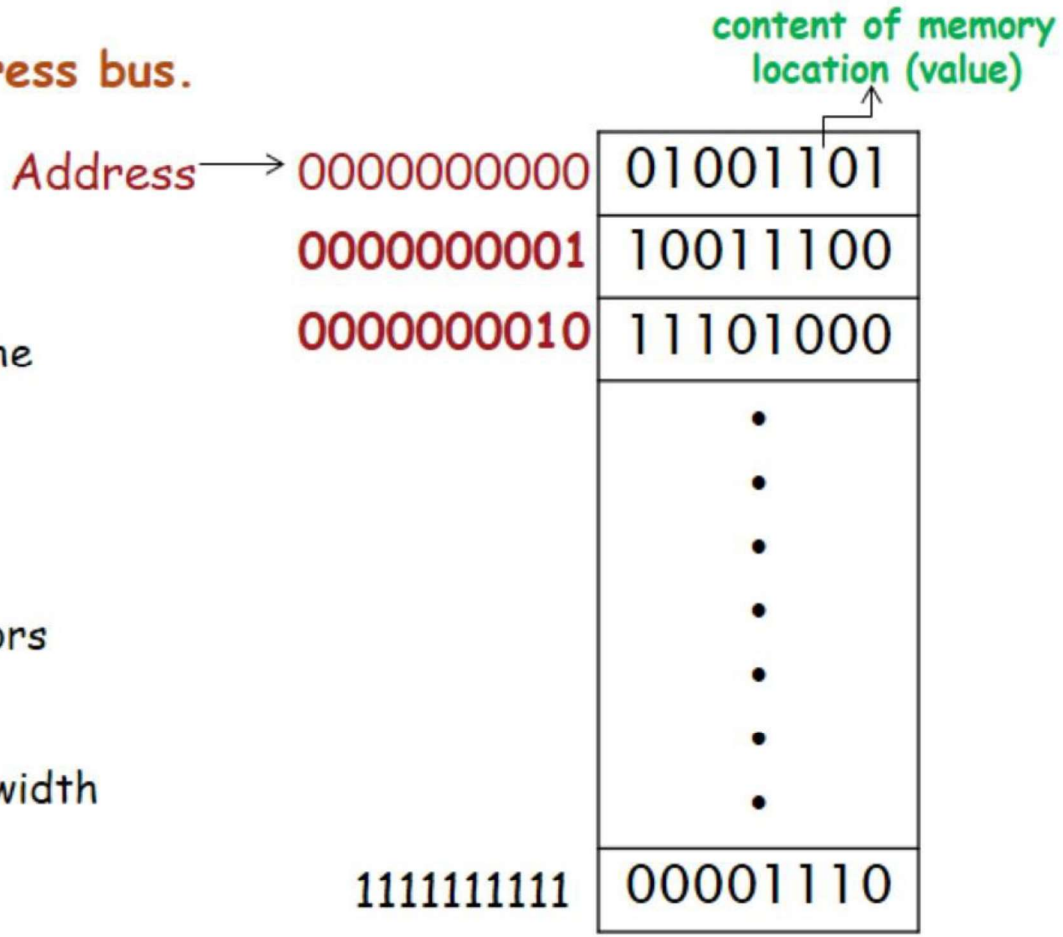
$$no\ of\ cycle = \frac{size\ \text{of data}}{\text{size of data bus}} = \frac{128\,byte}{4\,\text{byte}} = 32\,\text{bus cycle}$$

2. Address Bus:

- unidirectional group of wires which carries address information bits form processor to peripherals
- Each memory locations or interface register has associated with it a unique bit combination called an address.
- The number of bits used to specify an address clearly determines the set of all possible addresses.
- Memory address is put on address bus.

Memory

- Ordered sequence of bytes
  - The sequence number is called the **memory address**
- Byte addressable memory
  - Each byte has a unique address
  - Supported by almost all processors
- Physical address space
  - Determined by the address bus width



The size of address bus determine :
Let n = the number of lines in address bus

1. The size of memory connected with processor (capcity).

$$\text{size of memory} = 2^{n}$$

2. Size of memory address (physical address).

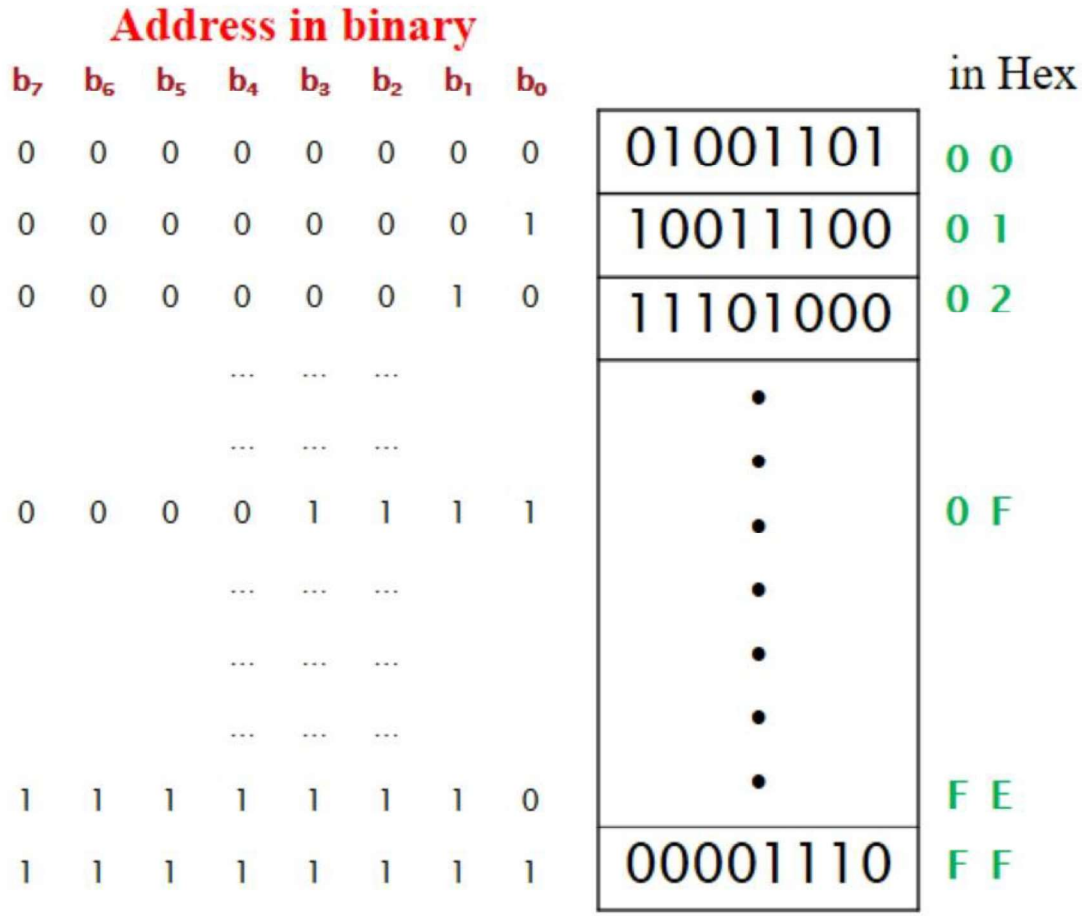
$$\text{size of memory address} = n$$

3. The address space of main memory (set of all possible addresses).

$$\text{address space} = 0 \rightarrow \left[2^{n} - 1\right] \qquad \text{(in decimal)}$$

EX: Processor has a 8 lines address bus

Size of memory = $2^{n} = 2^{8} = 256$ byte
Size of address for each memory location = 8 bits
Address space (decimal) = $0 \longrightarrow 2^{8}-1 = 0 \longrightarrow 255$ different memory location
Address space in hexadecimal



Address Space is the set of memory locations (bytes) that can be addressed
00 -- FF

EX: Processor has a 16 lines address bus

Size of memory = $2^n = 2^{16}$ = 64 K.Byte

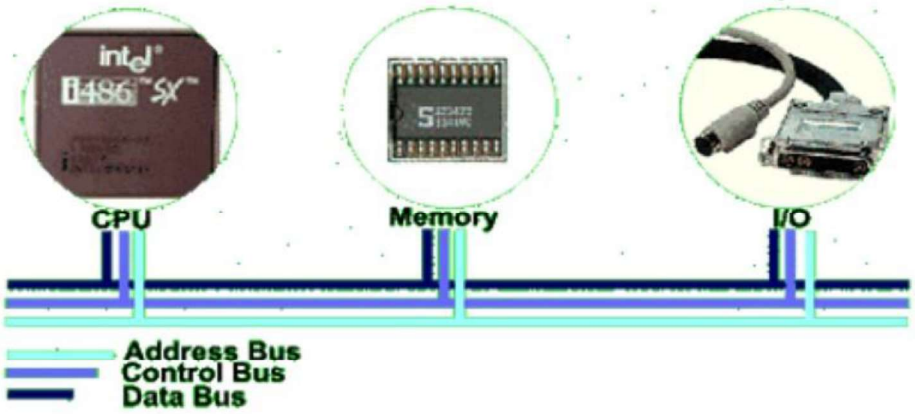
Size of address for each memory location = 16 bits

Address space (decimal) = 0 $\longrightarrow$ $2^{16}-1$ = 0 $\longrightarrow$ 65536 different memory location

Address space in hexadecimal 0000 -- FFFF

b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0		in Hex
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	01001101	0 0 0 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10011100	0 0 0 1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	11101000	0 0 0 2
	...	...	...	...		...	...	...			...	...	...			• • •	
	...	...	...			...	...	...			...	...	...				
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	00001110	0 0 0 F
				...	...	...	...	...	...		...	...	...			• • •	
				...	...	...	...	...	...		...	...	...				
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0		0 0 F F
	...	...	...	...		...	...	...			...	...	...				
																	F F F F

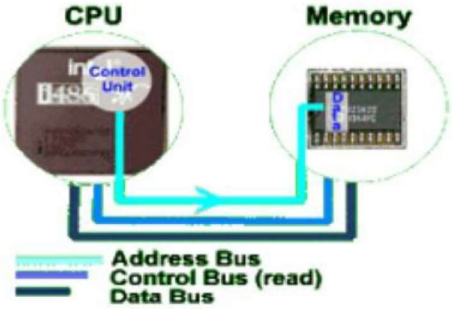
Control Bus: Regardless of the bus arrangement being used, a certain amount of control information must be passed back and forth among the CPU, the memory modules, and the device interfaces.



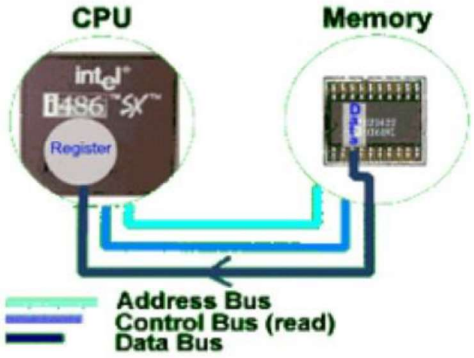
Fetch and Execute

The following pages illustrate the operations performed within a computer during a fetch and execute cycle.

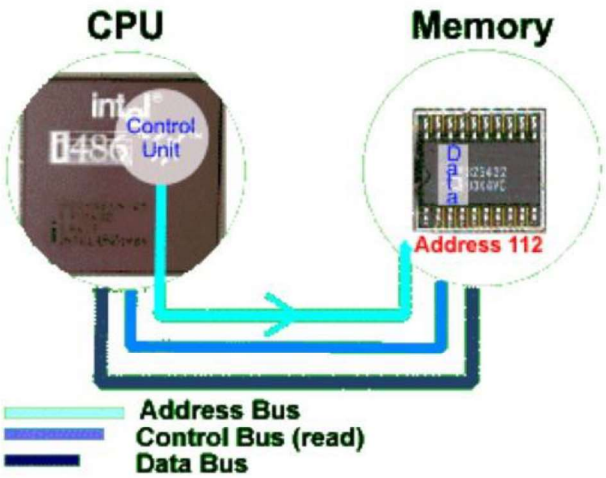
The control bus performs a read operation:

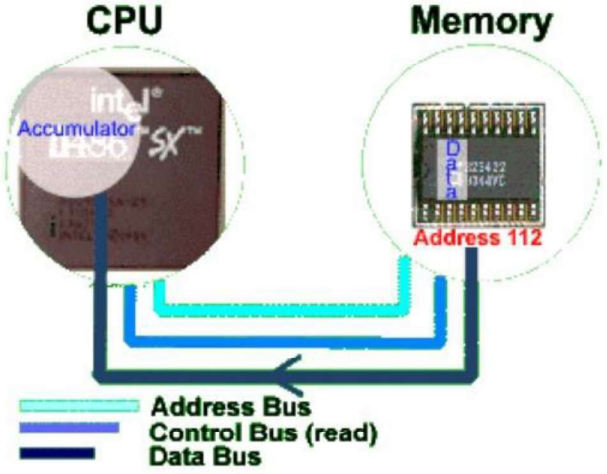


The control unit in the CPU prompts memory to put the instruction onto the data bus enabling the CPU to read the instruction onto its instruction decoder which is part of the control unit:



The next step involves the CPU decoding the instruction. This instruction is then executed. For example, if the instruction is for the Control Unit to load the contents of the memory location 112 into the accumulator:

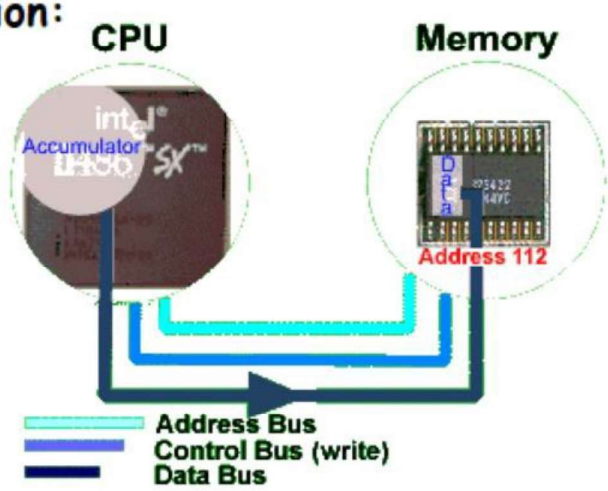




Performing a write operation

The control unit sets the address bus to location 112 and puts the value of the accumulator onto the data bus.

Finally the control bus performs a memory write operation:



Memory

The term memory refers to anything that stores information either permanently or temporarily. The memory of a computer can hold program instructions, data values, and the intermediate results of calculations. All the information in memory is encoded in fixed size cells called **bytes**.

A byte can hold a small amount of information, such as a single character or a numeric value between 0 and 255.

The CPU will perform its operations on groups of one, two, four, or eight bytes, depending on the interpretation being placed on the data, and the operations required.

Each byte can be accessed by an address. The address is a unique combination of bits that is transmitted to the memory via address lines in external bus.

Types of Memory

- **Internal Memory or Primary Memory –**

Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

- **External Memory or Secondary Memory –**

Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.