

## Memory mapped I/O

It is a way to exchange data and instructions between a CPU and peripheral **devices** attached to it. **Memory mapped IO** is one where the processor and the IO **device** share the same **memory** location(**memory**), i.e., the processor and IO **devices** are **mapped** using the **memory** address.

Memory-mapped I/O uses the same **address space** to address both **memory** and **I/O devices**. The memory and **registers** of the I/O devices are mapped to (associated with) address values. So when an address is accessed by the CPU, it may refer to a portion of **physical RAM**, or it can instead refer to memory of the I/O device. Thus, the CPU instructions used to access the memory can also be used for accessing devices. Each I/O device monitors the CPU's address bus and responds to any CPU access of an address assigned to that device, connecting the **data bus** to the desired device's **hardware register**. To accommodate the I/O devices, areas of the addresses used by the CPU must be reserved for I/O and must not be available for normal physical memory.

### ADVANTAGES

- All I/O locations are addressed in exactly the same manner as memory locations; no special repertoire of I/O instructions is therefore. Thus the overall size of the instruction set is reduced.
- All arithmetic and logical operations can be performed on I/O data directly
- Used in system where memory requirement is small

### DISADVANTAGES

- Part of the memory address space is lost. (however, that with ported I/O systems, not all of the available I/O address space is always used.)

**DMA:** The term DMA stands for direct memory access. The hardware device used for direct memory access is called the DMA controller. DMA **controller is**

a **control unit**, part of I/O device's **interface circuit**, which can transfer blocks of data between I/O devices and main memory with minimal intervention from the processor.

## When DMA operates:

