Embedded System Concepts 3

Summary:

1. **Processor:** Address Bus (specify no. of locations) ---- Databus = data width the processor outputs at a time
2. **Memory**: size = wordsize * $2^{AddressBus}$ ------ w.z. = databus
3. **Memory Fragmentation** is the phenomenon of having unused or wasted memory space between the allocated blocks
4. **Homogenous Memory** is where each memory of the same type is stacked after each other

Special cases:

1. Processor dataBus > Memory w.z.
   Result: data loss
2. Processor dataBus < Memory w.z.
   Result: data loss can occur / Memory not used
3. Processor Address space > no. of memory locations
   Result: free space (address not connected to memory location)
4. Processor Address space < no. of memory locations
   Result: Some memory locations can't be accessed by processor
   Then a MMU (Memory Management Unit) is used

What is MMU?

The MMU performs the same address translation and protection functions for the peripheral requests as for the CPU requests to access more memory locations, using a separate set of page tables called paging register, the downside would be more time being consumed to communicate with the MMU and some memory locations aren't accessible

Digital Design:

Von Neuman: Can only communicate to one memory at a time, Uses one addressBus, dataBus and controlBus shared for all memories and switches between them (one instruction per cycle)

Harvard Architecture: Uses multiple addressBus, dataBus and controlBus (That are triggered consecutively) to communicate with several memories at a time (multiple instructions per cycle)

	Harvard	*Von Neuman*
Speed	Faster	Slower
Complexity	High (write additional code)	Low

AVR architecture:

Is a modified Harvard architecture machine, where program and data are stored in separate physical memory systems that appear in different address spaces, but having the ability to read data items from program memory using special instructions.

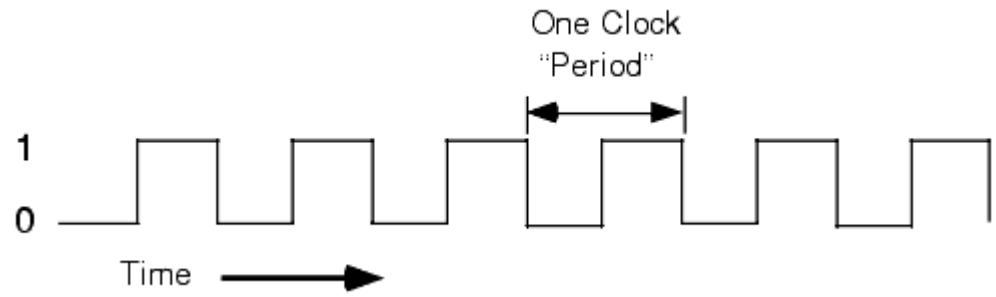
Two main way to map busses to the I/O memory:

1. **Port mapping:** using I/O busses (from processor)
2. **Memory mapping:** using SRAM busses

	Port mapped	*Memory Mapped*
Speed	Fast	Slow
complexity	High	Low

All processors have 3 steps of action: Fetching, Decoding and Executing

For the processor to work it must take an input signal called Clock



Every period has 4 events:

1. Rising edge
2. High level
3. Falling edge
4. Low level

The 3 actions are triggered by these events. Ex. At rising edge: fetch, at high level: decode, and at falling edge: execute

CLK system: ex. 8MHz → 8 Million cycles in one second

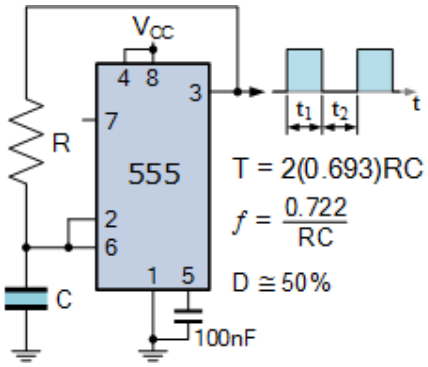
CLK = The frequency of the system

MIPS (Million Instruction Per Second): CLK / no. of cycles per instruction

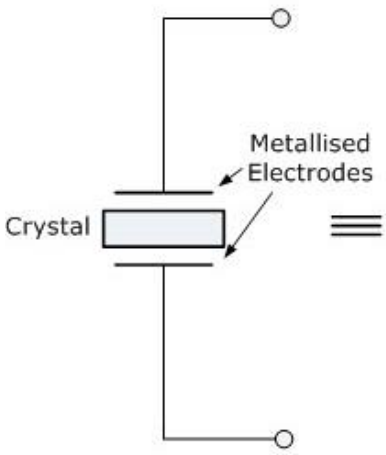
Time of one Instruction = Time of one period of the CLK

CLK Signal Generator:

1. Electrical CLK Source (RC System):



2. Mechanical CLK System: (Ceramic, Crystal) Chip with a small Ceramic or Crystal plate that oscillates when voltage is applied



	RC	*Crystal*	*Ceramic*
Accuracy	Lowest	Highest	Moderate
Cost	Cheapest	Expensive	Affordable
Settling time	Longest	Shortest	Moderate
Temperature noise immunity	Low	High	High
Vibration noise immunity	High	Low	Low
EMI (Electromagnetic Interface) noise immunity	Low	High	High