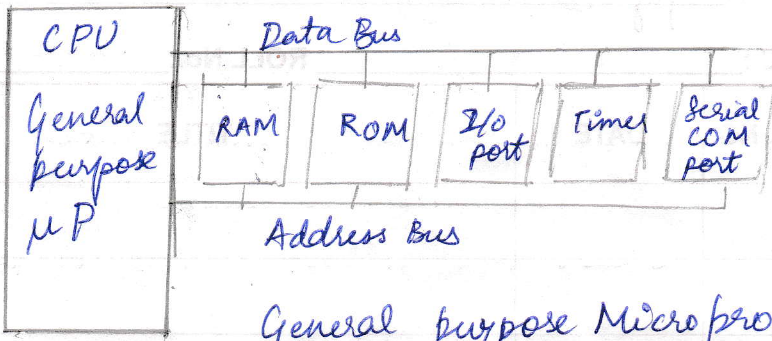
Micro Controller : Programming and Interfacing

	Micro Controller	Micro Processor.
Def^n	- It is a chip optimized to control electronic devices.	- It is a controlling unit of a micro-computer wrapped inside a small chip
Components	- Contains a CPU, memory and I/O all integrated into one chip.	- Only contains Central processing Unit
Usage	- Used in embedded systems	- Used in personal computers
Bus structure	- Has an internal controlling Bus.	- Uses an external Bus. to interface with RAM, ROM, etc.
Architecture used	- Harvard architecture	- Von Neumann model
	- Straight-forward, with ~~a~~ fewer instructions to process	- Complicated, with a lot of instructions to process.
	- Inexpensive	- Expensive.
	- Power Consumption → low	- high.



General purpose Microprocessor

CPU	RAM	ROM
I/O ADC	Timer	Serial COM port

Micro ~~Processor~~ Controller

Embedded System // google

An Embedded System is a computer system(- a combination of a computer processor, computer memory, and input/output, peripheral devices -) that has a dedicated function within a larger mechanical or electrical system

Criteria for choosing a micro-controller // learn.kletec

i) Speed of the MC ; does it meet the demands required.
ii) The type of packaging (eg. dual inline , Quad flat, etc).
iii) on chip RAM & / ROM.
iv) Power consumption.
v) No. of I/O pins ; determines the no. of i/o devices it can cont
vi) Cost per unit.
vii) Capability & ease of upgradation , Scalability.
viii) Availability of Software & hardware dev tools.
Eg → Emulators, debuggers, compilers, etc.
ix) Wide availability and reliable sources for of the MC.
x) adheres to current standards ; will the MC be obsolete in the near future.

Harvard and von Neumann architectures // Mazidi pg 37

Every μC/μP must has memory space to store program (code) and data. While code provides instructions to the CPU, data provides the information to be processed.

The CPU uses buses to access the code ROM & data RAM memory ~~bs~~ spaces. The early computers used the same bus to access both code and data.

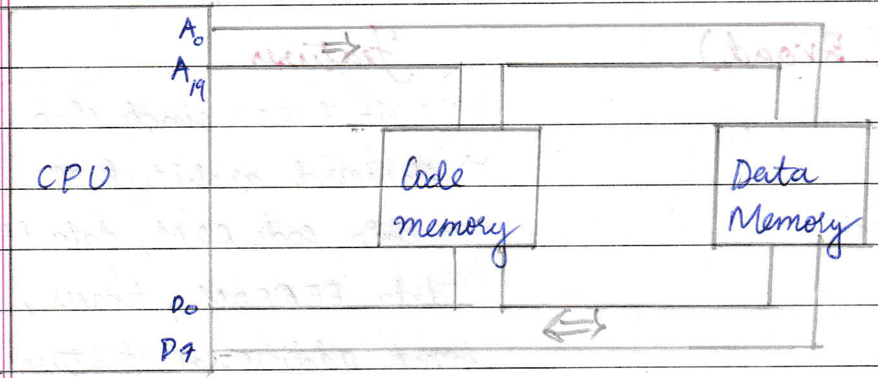
Such an architecture is commonly referred to as von Neumann (princeton) architecture. This meant that, the process of accessing the code & data would cause them to get in each other's way & slow down the processing of the CPU, because each had to wait for the other to finish fetching.

To speed up processing, some CPUs use what is called Harvard architecture. In this architecture, there are 2 seperate buses for code and data memory. Thus there is a need for four sets of buses:

i) a set of data-buses to carry data into/out of the CPU
ii) a set of address-buses for accessing data ~~memory~~.
iii) a set of data-buses to carry code into the CPU.
iv) a address-bus to access code.

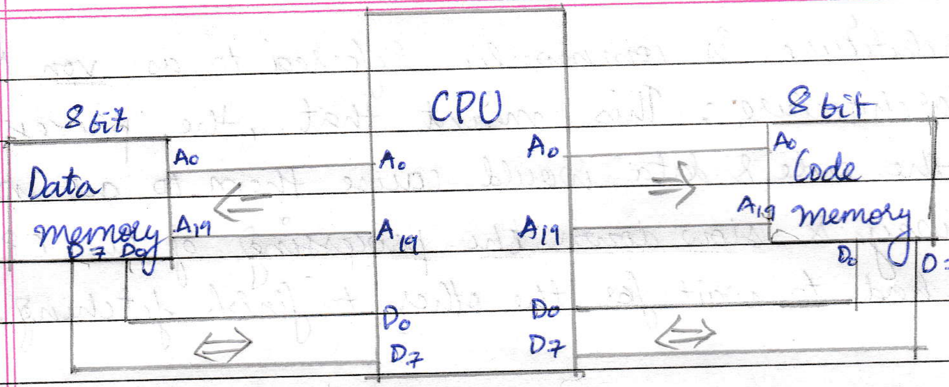
characteristics and tradeoffs.

This is easy to impliment inside an IC chip such as a µC where both data RAM and code ROM are on-chip and distances are on micron/millimeter scale. But implimenting this for systems such as x86 IBM PC-type computers is very expensive because the RAM and ROM that hold the data and code are external to the CPU. Seperate wire traces for code and data will make ~~it~~ the motherboard large and expensive. (more...refer)

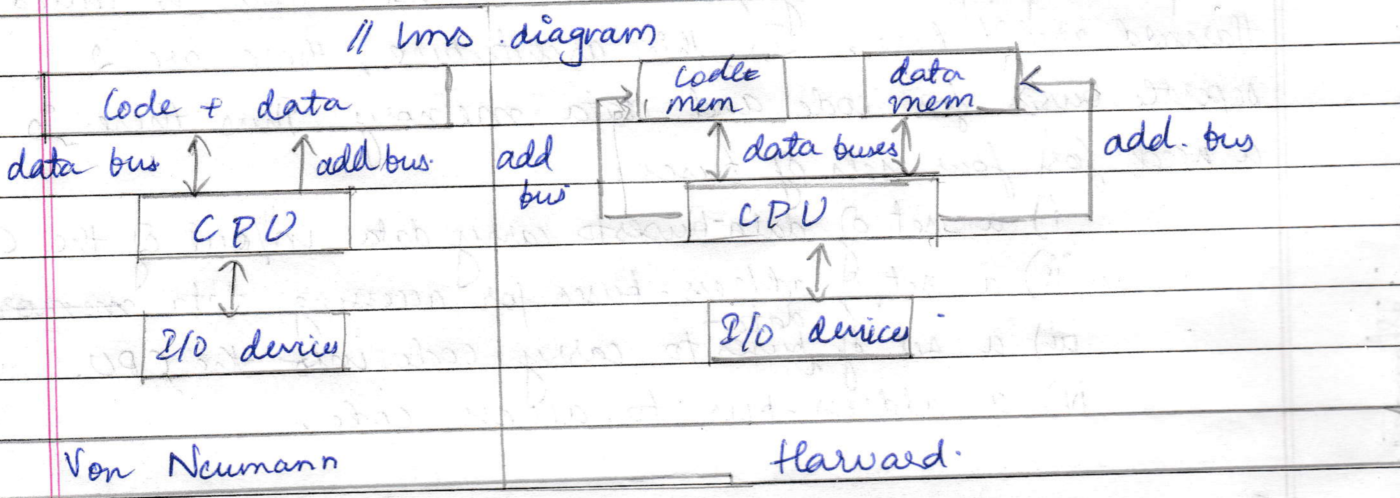


Von Neumann Architecture

Harvard Architectu



// lms diagram



AVR and Arduino

↳ Alf-Egil Bogen Vegard Wollan RISC μC

↳ Advanced Virtual RISC μC

Except for AVR32 (a 32 bit μC), AVRs are all 8 bit μP, meaning the CPU can work on only 8 bits of data at a time

Classifications (Broad)

i) Mega

ii) Tiny

iii) Speeed purpose

iv) Classic

Features

- 8 bit RISC single chip μC
- Harvard architecture
- on chip code ROM, data RAM, data EERROM, timers, I/O p
- some additional features like ADC, PWM & serial interfa such as USART, CAN, USB etc