Defining a Microcontroller...

→ Microcontroller is nothing but the combination of all the peripheral components that are required for the proper execution of a computer system, encapsulated in a single chip.

→ In simple words-

- Microcontroller is a single chip computer.
- It has in built code & data memories.
- It has in built Serial & Parallel Ports.
- It has capability of uploading, storing & running a program or instructions.

Why to Use Microcontroller?

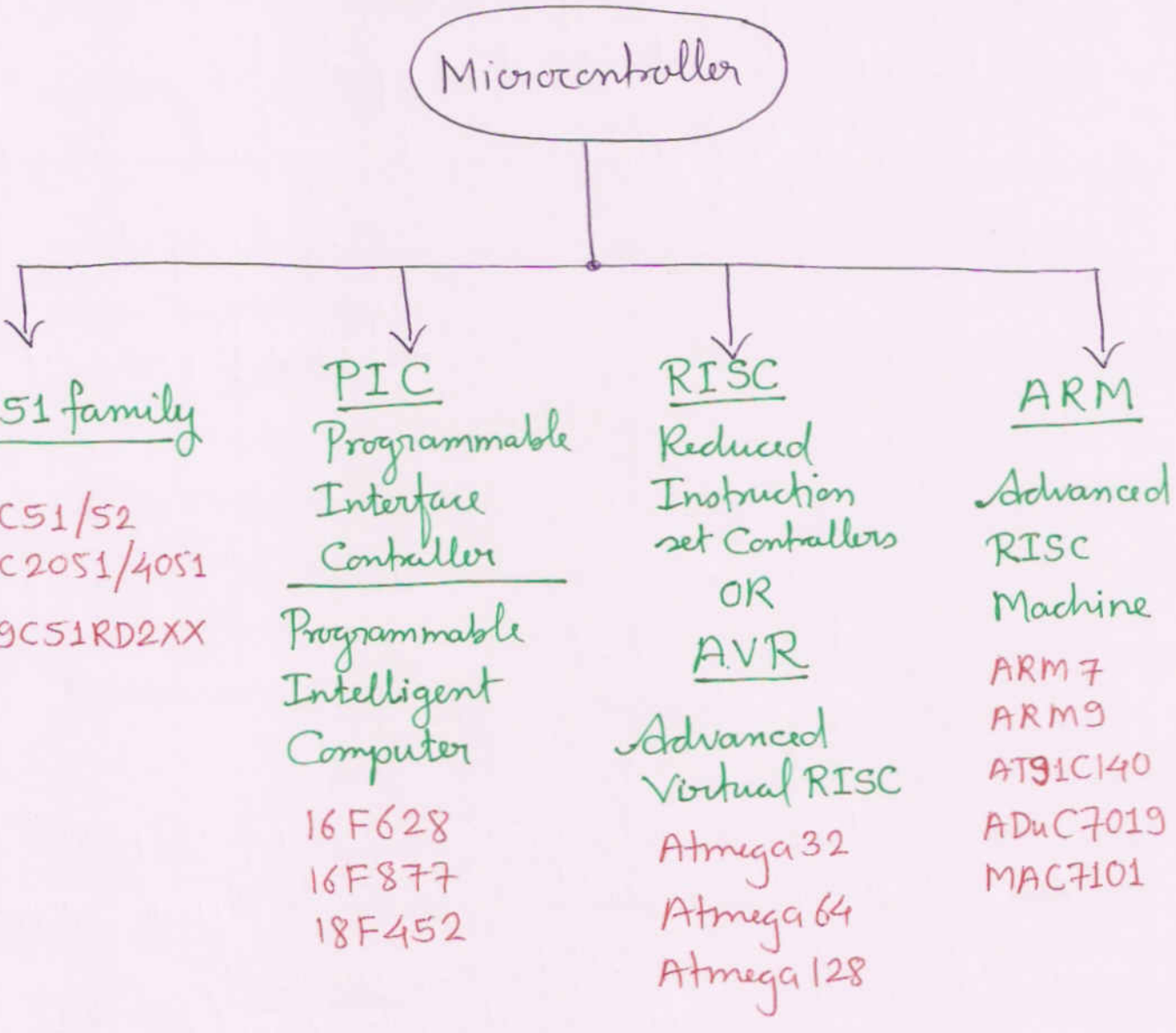
→ Being inexpensive single chip computers, microcontrollers are easy to embed into larger electronic circuit designs. their ability to store and run unique programs makes them extremely versatile.

→ One can program a microcontroller to make decisions and perform functions based on situations (I/O line logics) and events.

→ the math and logic functions allows the microcontroller to mimic sophisticated logic and electronic circuits.

→ Programs can also make the microcontroller behave as a neural network and/or a fuzzy logic controller.

Different Families of Microcontrollers



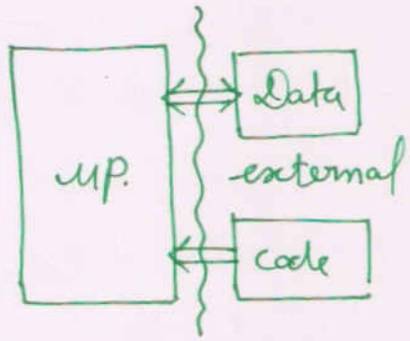
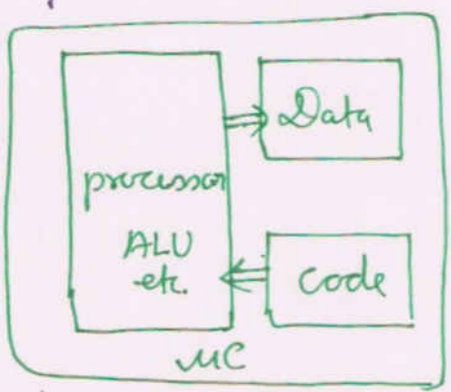
8051 family → it is generally used in low cost, less critical & less complex systems.
- currently only used for project development
- outdated family in embedded systems.

PIC family → it is widely used in industrial control system equipments.
- evergreen controllers to be used for industrial purpose.

AVR family → this is upcomming controller family & are widely used for faster & less power consumption application.

ARM family → in sophisticated high speed military applications

Difference between Microprocessor & Microcontroller

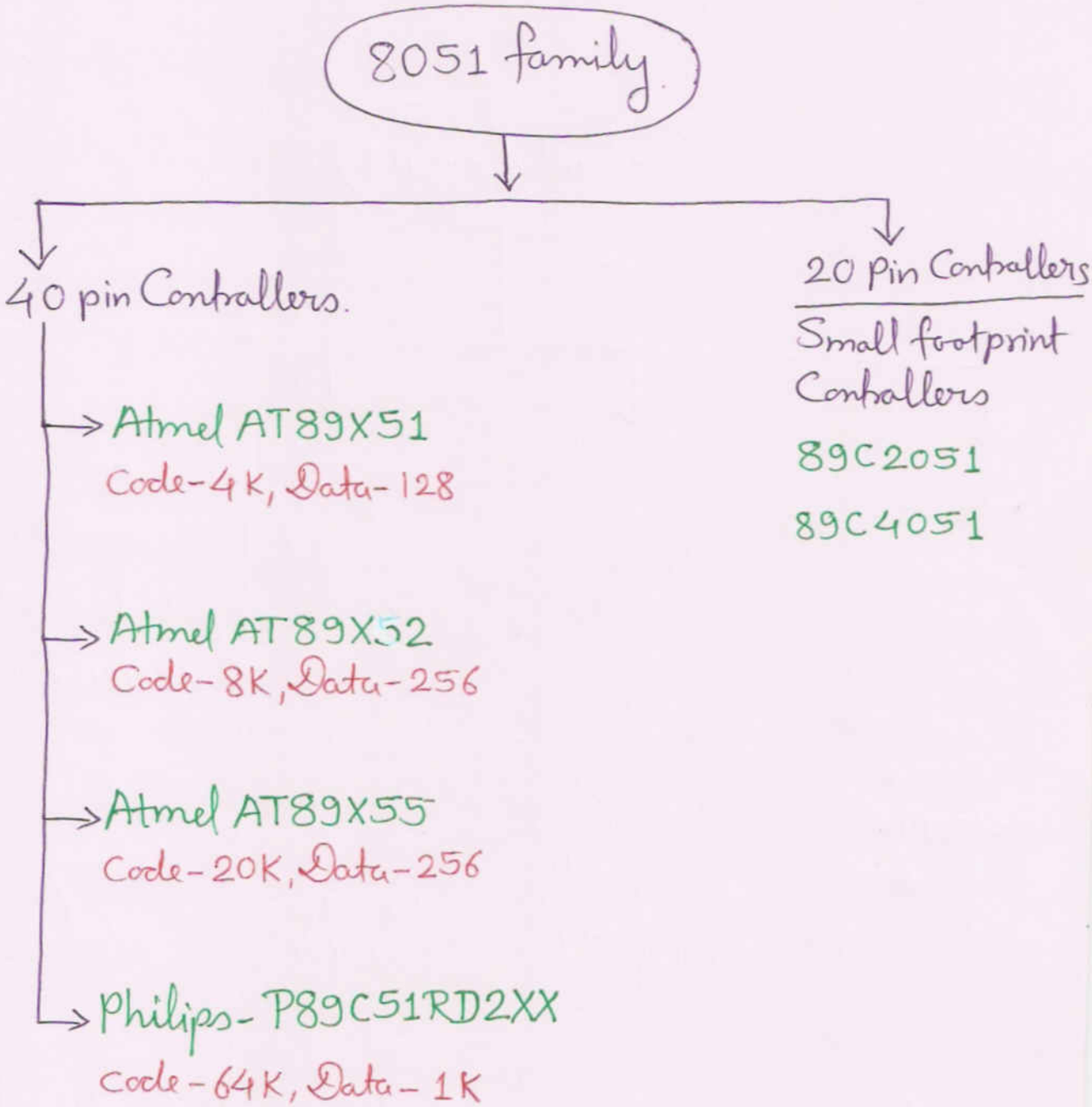
μP	μC.
① μP requires some peripheral Components to run properly i.e RAM / ROM / IO Ports etc. 	① μC has in built RAM / ROM (i.e. Data & instruction memory). so it does not require any peripheral Components to work. 
if the processor in the μC is based on the architecture of 8085 then the controller is 8051 controller family. → thus μC is nothing but the combination of all the minimum system peripherals (like Code & Data memories) that are required along with μP to work properly.	
② μP is byte addressable only. → to toggle one particular bit on the port without Modifying other bits of the port requires multiple instructions (logical)	② μC is byte and bit both addressable. → a single bit on port can be easily modified just by single instruction.

Benefits of Using Microcontroller Over Microprocessor

→ Size of system reduces drastically.
→ Less component Count.
→ Cost is reduced.
→ System Design Complexity becomes less.
→ Debugging in hardware is easy.
→ Power Consumption is also reduced as low component count.
→ Code size is reduced as many works can be done with single instructions.
→ Code complexity is reduced.
→ Less memory is required.
→ Speed of processing a particular task is increased.
→ Debugging in code is easy.
→ Multiple peripherals can be interfaced to the same port. as system is bit addressable.

the above are some of benefits of uC over uP. Hence we are going to use uC in our project and not uP.

The 8051 Family - Choosing One



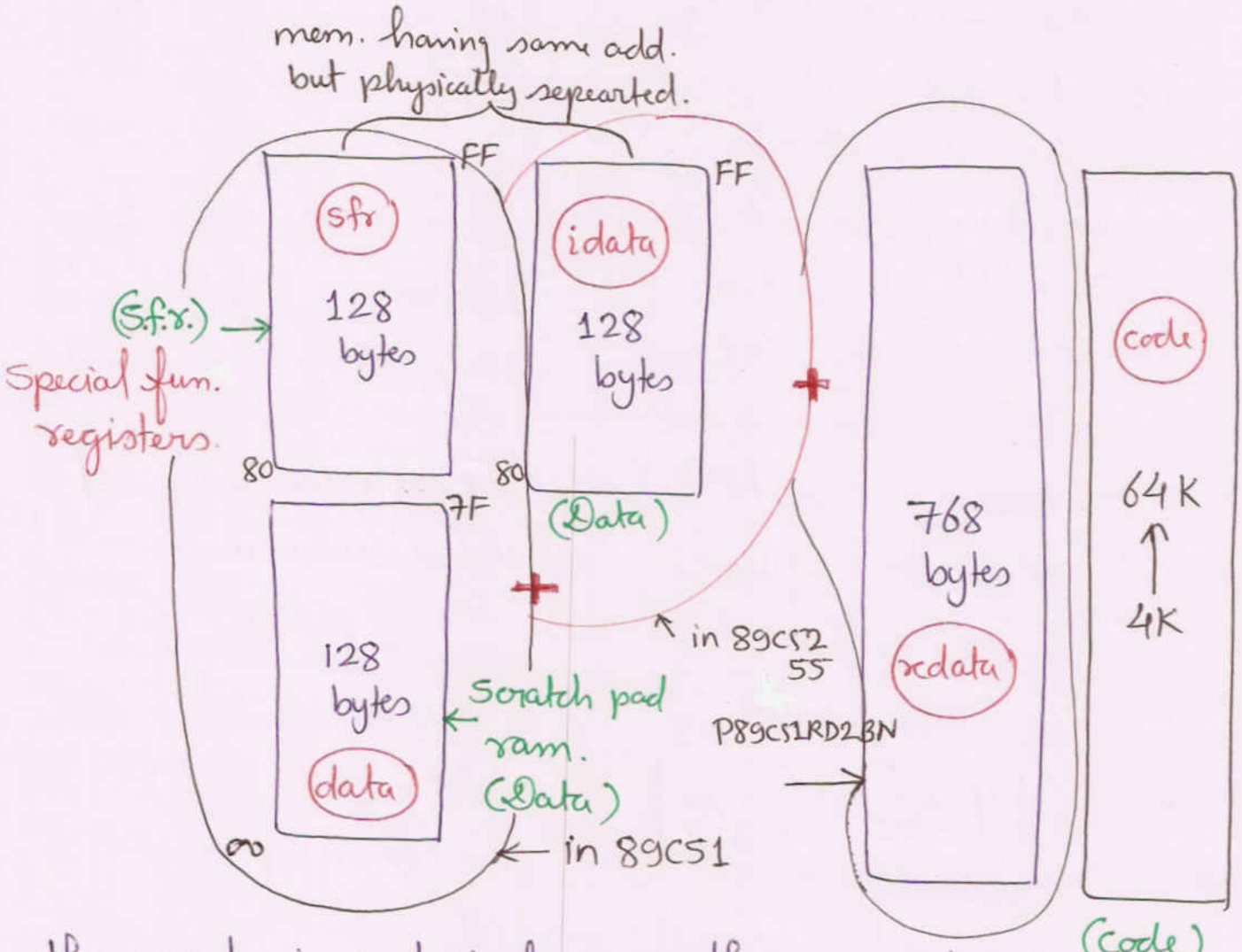
In market there are various controller available in various packages & memory sizes. Depending upon the project System requirement we choose one of the above mentioned Controllers.

Difference between AT89C51 and P89C51RD2BN.

AT89C51	P89C51RD2BN
① Code mem – 4K Data mem – 128	① Code mem – 64K Data mem – 1K
② the external clock is devided by 12. Slow controllers	② the external clock is divided by 12 or 6 – can be choosen by programmer. fast Controllers.
③ this can generate maximum baude rate of 9600 bps.	③ this can generate maximum baude rate upto 115200 bps
④ this requires a programmer / burnner to be programmed.	④ this is ISP. (in System programmable). It can be reprogrammed within the system itself without removing

for low speed & less memory projects we use AT89C51 and for high speed application and/or more memory based applications we can use P89C51RD2BN.

Memory Organization of μC with respect to Programming in Keil S/W.



the wards in red circles are the access/declartion keywords in embedded software i.e. C.

data → used to declare small variables in basic RAM

e.g. unsigned char data i;

- unsigned char → one byte
- data → if nothing is written here it will be assumed as data.

idata → internal data mem.

the upper 128 bytes of data mem. it is used to store Compact variables e.g. long idata i;

xdata → external data mem. used to store large variables e.g. char xdata i[20];

code → used to store constant values in external code mem.