Computer Interfacing

Chapter-2: Popular Microcontrollers

Popular Microcontroller Technologies or families

- **8051 Microcontroller:**

The most universally employed set of microcontrollers come from the 8051 family. The original 8051 microcontroller was initially invented by Intel. The two other members of this 8051 family are-

- 8052 – This microcontroller has 3 timers & 256 bytes of RAM. Additionally, it has all the features of the traditional 8051 microcontroller. 8051 microcontroller is a subset of 8052 microcontroller.
- 8031 – This microcontroller is ROM less, other than that it has all the features of a traditional 8051 microcontroller. For execution an external ROM of size 64K bytes can be added to its chip.

8051 microcontroller brings into play 2 different sorts of memory such as- NV-RAM, UV-EPROM and Flash.

8051 Microcontroller Architecture:

8051 microcontroller is an 8-bit microcontroller launched in the year 1981 by Intel Corporation. It is available in 40 pin DIP (dual inline package). It has 4kb of ROM (on-chip programmable space) and 128 bytes of RAM space which is inbuilt, if desired 64KB of external memory can be interfaced with the microcontroller. There are four parallel 8 bits ports which are easily programmable as well as addressable. An on-chip crystal oscillator is integrated in the microcontroller which has crystal frequency of 12MHz. In the microcontroller there is a serial input/output port which has 2 pins. Two timers of 16 bits are also incorporated in it; these timers can be employed as timer for internal functioning as well as counter for external functioning. The microcontroller comprise of 5 interrupt sources namely- Serial Port Interrupt, Timer Interrupt 1, External Interrupt 0, Timer Interrupt 0, External Interrupt 1. The programming mode of this micro-controller includes GPRs (general purpose registers), SFRs (special function registers) and SPRs (special purpose registers).

- **PIC Microcontroller:**

Peripheral Interface Controller (PIC) provided by Micro-chip Technology to categorize its solitary chip microcontrollers. These appliances have been extremely successful in 8 bit micro-controllers. The foremost cause behind it is that Micro-chip Technology has been constantly upgrading the appliance architecture and included much required peripherals to the micro-controller to go well with clientele necessities. PIC microcontrollers are very popular amid hobbyists and industrialists; this is only cause of wide availability, low cost, large user base & serial programming capability.

PIC Microcontroller Architecture:

The architecture of the 8 bit PIC microcontrollers can be categorized as below:

- **Base Line Architecture:** In the base-line architecture PIC microcontrollers of PIC10F family is included, other than that a fraction of PIC12 & PIC16 families are also included. These gadgets make use of 12-bit program word architecture with six to twenty-eight pin package alternatives. Briefly defined attribute set of baseline architecture allows the most lucrative product solutions. This architecture is perfect for battery enabled gadgets. The PIC10F200 series is another reasonably priced 8-bit flash micro-controller with a 6 pin package.
- **Mid-Range Architecture:** In this midline member of PIC12 & PIC16 families are added that attribute 14-bit program word architecture. The midrange PIC16 gadgets proffer a broad variety of package alternatives (from 8 to 64 package), with low to high levels of peripheral incorporation. This PIC16 appliance attributes a variety of analog, digital & serial peripherals, like- SPI, USART, I2C, USB, LCD & A/D converters. The mid-range PIC16 micro-controllers have suspended controlling ability with an eight level hardware load.
- **High Performance Architecture:** The high performance architecture included the PIC18 family of appliances. These micro-controllers make use of 16-bit program word architecture along with 18 to 100 pin package alternatives. The PIC18 appliances are high performance micro-controllers with incorporated Analog to Digital converters. All PIC18 micro-controllers integrate a highly developed RISC architecture that supports flash appliances. The PIC18 has improved foundation attributes, 32 level deep load and several inner and exterior interrupts.

- **AVR Microcontroller:**

AVR also known as Advanced Virtual RISC, is a customized Harvard architecture 8-bit RISC solitary chip micro-controller. It was invented in the year 1966 by Atmel. Harvard architecture signifies that program & data are amassed in different spaces and are used simultaneously. It was one of the foremost micro-controller families to employ on-chip flash memory basically for storing program, as contrasting to one-time programmable EPROM, EEPROM or ROM, utilized by other micro-controllers at the same time. Flash memory is a non-volatile (constant on power down) programmable memory.

AVR Microcontroller Architecture:

AVR microcontrollers' architecture was developed by Alf-Egil Bogen and Vegard Wollan. The name AVR is derived from the names of the architecture developers of the microcontroller. The AT90S8515 was the foremost micro-controller which was AVR architecture based; on the other hand, the foremost micro-controller to strike the commercial marketplace was AT90S1200 which was launched in the year 1997.

The SRAM, Flash and EEPROM all are incorporated on a single chip, thereby eliminating the requirement of any other external memory in maximum devices. Several appliances comprise of parallel external bus alternative, so as to add extra data memory gadgets. Approximately all appliances, except TinyAVR chips comprise serial interface, which is used to link large serial Flash & EEPROMs chips.

ARM Microcontroller:

ARM is the name of a company that designs micro-processors architecture. It is also engaged in licensing them to the producers who fabricate genuine chips. In actuality ARM is a 32-bit genuine RISC architecture. It was initially developed in the year 1980 by Acorn Computers Ltd. This ARM base microprocessor does not have on-board flash memory. ARM is particularly designed for micro-controller devices, it is simple to be trained and make use of, however powerful enough for the most challenging embedded devices.

ARM Microcontroller Architecture:

The ARM architecture is a 32-bit RISC processor developed by ARM Ltd. Owing to its power-saving attributes, ARM central processing units are prevailing in the mobile electronics marketplace, where less power expenditure is a vital design aim. ARM architecture comprise of the underneath RISC elements:

- Maximum single cycle functioning
- Constant 16×32 bit register file.
- Load or store architecture.
- Preset instruction width of 32 bits so as to simplify pipe-lining and decoding, at minimized code density.
- For misaligned memory access there is no support.

Microcontroller Applications

- **Application of Microcontroller in Day to Day Life Devices:**
  - Light sensing & controlling devices
  - Temperature sensing and controlling devices
  - Fire detection & safety devices
  - Industrial instrumentation devices
  - Process control devices

- **Application of Microcontroller in Industrial Control Devices:**
  - Industrial instrumentation devices
  - Process control devices

- **Application of Microcontroller in Metering & Measurement Devices:**
  - Volt Meter
  - Measuring revolving objects
  - Current meter
  - Hand-held metering systems

Features of a Microcontroller

- **CPU:**
  Microcontrollers brain is named as CPU. CPU is the device which is employed to fetch data, decode it and at the end complete the assigned task successfully. With the help of CPU all the components of microcontroller are connected into a single system. Instruction fetched by the programmable memory is decoded by the CPU.

- **Memory:**
  In a microcontroller memory chip works same as microprocessor. Memory chip stores all programs & data. Microcontrollers are built with certain amount of ROM or RAM (EPROM, EEPROM etc) or flash memory for the storage of program source codes.

  Memory Types:
  - **EEPROM (Electrically Erasable Programmable Read Only Memory):**
    Many microcontrollers have limited amounts of EEPROM on the chip. EEPROM seems more suited (because of its economics) for small amounts of memory that hold a limited number of parameters that may have to be changed from time to time. This type of memory is relatively slow, and the number of erase/write cycles allowed in its lifetime is limited.
  - **FLASH (EPROM)**
    Flash provides a better solution than regular EEPROM when there is a requirement for large amounts of non-volatile program memory. It is both faster and permits more erase/write cycles than EEPROM.
  - **OTP - One Time Programmable**
    An OTP is a PROM (Programmable Read-Only-Memory) device. Once your program is written into the device with a standard EPROM programmer, it cannot be erased or modified.

- **Input/output ports**
  I/O ports are basically employed to interface or drive different appliances such as- printer, LCD, LED etc. For example: USART (Universal Synchronous/Asynchronous Receiver/Transmitter), UART (Universal Asynchronous Receiver/Transmitter).

- **Serial Ports**
  These ports give serial interfaces amid microcontroller & various other peripherals such as parallel port. For example: SPI (Serial Peripheral Interface), SCI (Serial Communication Interface).

- **Timers**

  A microcontroller may be in-built with one or more timer or counters. The timers & counters control all counting & timing operations within a microcontroller. Timers are employed to count external pulses. The main operations performed by timers are-

  - pulse generations
  - clock functions
  - frequency measuring
  - modulations
  - making oscillations

- **ADC (Analog to Digital Converter)**

  ADC is employed to convert analog signals to digital ones. The input signals need to be analog for ADC. The digital signal production can be employed for different digital applications (such as- measurement gadgets).

- **DAC (Digital to Analog Converter)**

  This converter executes opposite functions that ADC perform. This device is generally employed to supervise analog appliances like- DC motors, etc.

AVR Microcontroller

AVR Architecture was conceived by two students (Alf Egil Bogen & Vegard Wollan RISC) at Norwegian Institute of Technology (NTH) and further refined and developed at Atmel Norway (Atmel AVR)

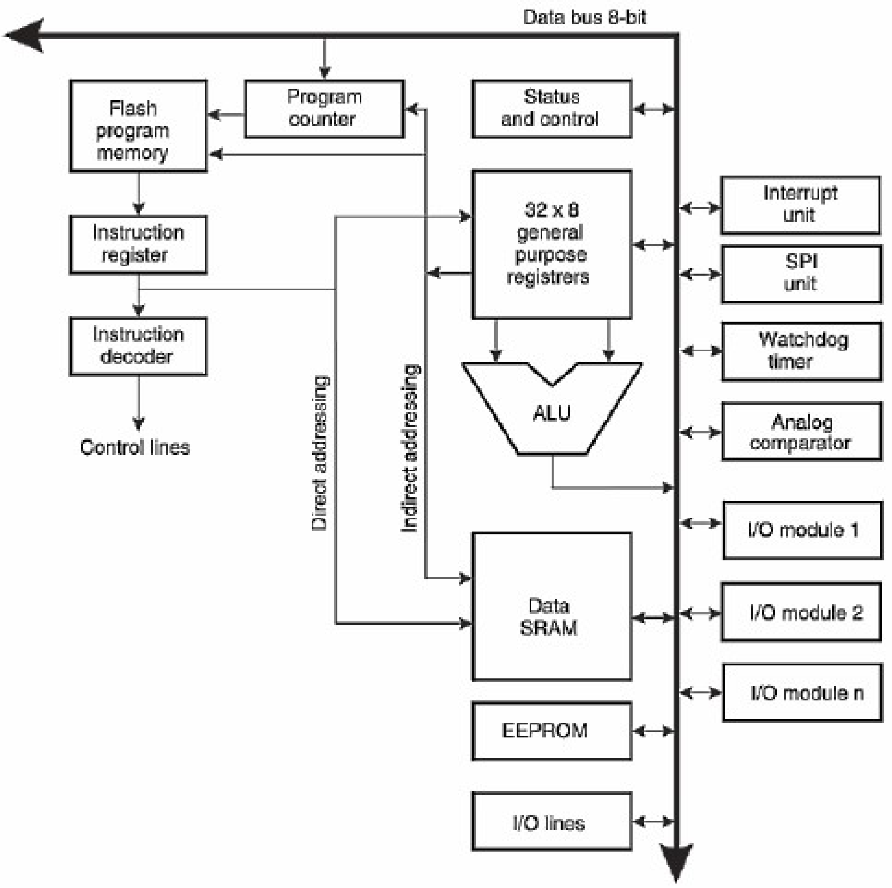
Atmel says that the name AVR is not an acronym and does stand for anything in particular. The creators of the AVR gave no definitive answer as to what term AVR stands for. However it commonly accepted that AVR stands for Alf Egil Bogen & Vegard Wollan's RISC processor.

AVR Core Architecture

In order to maximize performance and parallelism, the AVR uses a Harvard architecture –with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory. In the diagram, we can see Flash Program Memory, Data SRAM and EEPROM.

The AVR Architecture contains two types of Registers, i.e. General Purpose and Special Purpose registers.

- **General Purpose Registers:**
  - There are 32 general purpose registers.
  - Each register has 8-bits storage capacity.
  - Single clock cycle access time.
  - They are named R0 to R31.
  - Can store both data & addresses.
- **Special Purpose Registers:**
  - There are 3 special purpose registers.
  - Program Counter (PC), Stack Pointer (SP), Status Register (SREG)
  - Other I/O registers called SFRs



Indirect Address Register Pointers

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing, enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit **X-register (R27, R26), Y-register (R29, R28) and Z-register (R31, R30).** In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement.

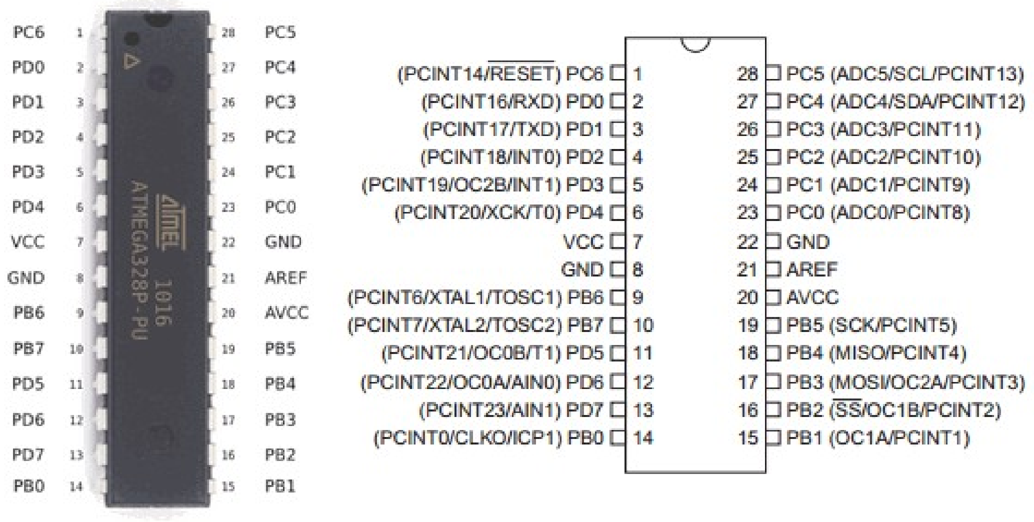
I/O Memory

- The I/O memory is dedicated to specific functions such as timers, serial communication, I/O ports, ADC and so.
- Their functions are fixed in the time of design.
- AVR I/O memory is made of 8-bit registers.
- All AVRs have at least 64 bytes of I/O memory location. This is called standard I/O memory.
- In other microcontrollers, the I/O registers are called Special Functions Registers (SFR).

Program Counter Register (PC)

- Holds address of the next program instruction to be executed.
- Automatically incremented when the ALU executes an instruction.

ATmega328P pin mapping



Interrupts

During **interrupts** and subroutine calls, the return address **Program Counter** (PC) is stored on the **Stack**. **The Stack** is effectively allocated in the general **Data SRAM**, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the **Status Register**. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

ALU

The high-performance **AVR ALU** operates in direct connection with all the 32 general purpose working registers. Within a **single clock cycle**, arithmetic operations between general purpose registers or between a register and an immediate are executed.

The ALU operations are divided into three main categories: **arithmetic, logical, and bit-functions.**

Status Register (SREG)

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

- Contains information about the results of the most recently executed arithmetic operation.
- Information can be used for altering program flow in order to perform conditional operations.
- Updated after any of ALU operations by H/W.

I	T	H	S	V	N	Z	C
7	6	5	4	3	2	1	0

Status Register Bits

- Bit 7 (I): Global Interrupt Enable
    - ❖ Used to enable and disable interrupts.
    - ❖ 1 means enabled and 0 means disabled.
    - ❖ Cleared by hardware after an interrupt has occurred and set by the RETI instruction to enable subsequent interrupts.
- Bit 6 (T): Bit Copy Storage
    - ❖ The Bit Copy Instructions such as BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit.
- Bit 5 (H): Half Carry Flag
    - ❖ Indicates half carry (carry from bit-4).
    - ❖ Useful is BCD arithmetic.
- Bit 4 (S): Sign Bit
    - ❖ Exclusive OR between N and V
    - ❖ $S = N \oplus V$
- Bit 3 (V): Two's Complement Overflow Flag
- Bit 2 (N): Negative Flag
    - ❖ The most significant bit of the result.
    - ❖ 1 means negative and 0 means positive.
- Bit 1 (Z): Zero Flag
    - ❖ Indicates a zero result in an arithmetic or logic operation.
    - ❖ 1 means result is zero and 0 means result is non-zero.
- Bit 0 (C): Carry Flag
    - ❖ Set if there was carry from the MSD of the result

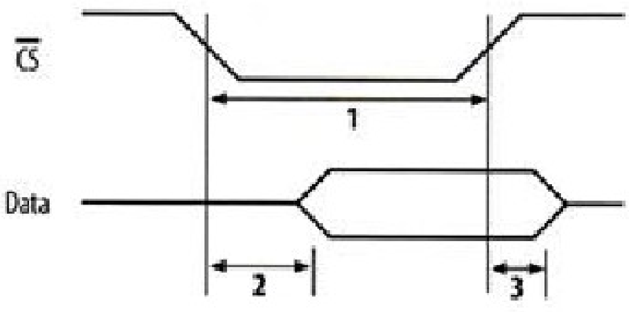
Stack and Stack Pointer (SP)

- The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls.
- The Stack Pointer Register always points to the top of the Stack.
- Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.
- The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space.
- The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt.
- The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

Watchdog Timer (WDT)

A watchdog timer (WDT) is a hardware timer that automatically generates a system reset if the main program neglects to periodically service it. It is often used to automatically reset an embedded device that hangs because of a software or hardware fault.

Timing Diagram



1. Code Select Hold Time
2. CS to Data Valid Time
3. Data Hold Time