AVR Bare Metal Programming Prerequisites

History:

AVR is a family of micro controllers developed since 1996 by Atmel, acquired by Microchip Technology in 2016. These are modified Harvard architecture 8-bit RISC single-chip micro controllers

AVR was one of the first micro controller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other micro-controllers at the time. AVR micro controllers find many applications as embedded systems.

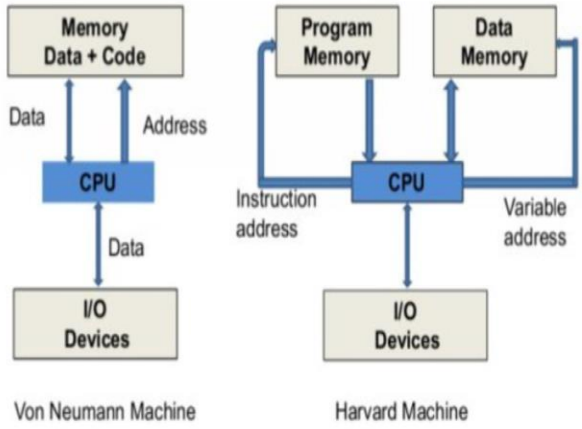
AVR stands for **Alf and Vegard's RISC processor**. Note that the use of "AVR", generally refers to the 8-bit RISC line of Atmel AVR microcontrollers.

RISC (Reduced Instruction Set Computer)

Reduced Instruction Set Computer, it is a micro-controller architecture which utilizes small and highly optimizes instruction set.

RISC provides high performance per watt for battery operated devices where energy efficiency is key. A RISC processor executes one action per instruction. By taking just one cycle to complete, operation execution time is optimized.

HARVARD & VON NEUMANN ARCHITECTURE



Von-Neuman	Harvard
First digital computer architecture. Introduced stored program concept	Computer architecture based on Harvard
One memory module for data and instructions	Have different memory modules for data and instructions.
common bus for data and instructions	Individual buses for data and instructions
CPU takes 2 clocks to execute one instruction. Because fetch data before executing an instruction.	Can execute instruction one clock cycle.
CPU can not fetch instructions and data read/write at the same time.	CPU can not fetch instructions and data read/write at the same time.
Slow	Fast

Difference between microprocessor and micro-controller

Summary	Microprocessor	Microcontroller
Applications	Advanced data processing, video, computer vision, personal computers, fast communications, multi-core computation.	Embedded devices, control systems, smartphones, consumer electronics.
Processing Power	Higher	Lower
Memory	External - Flexible	Internal – Limited Size
Power Consumption	Higher	Lower
Size	Larger	Smaller
Price	Expensive	Cheaper
I/O	Need external peripherals with I/O pins	Programmable digital and analog I/O pins

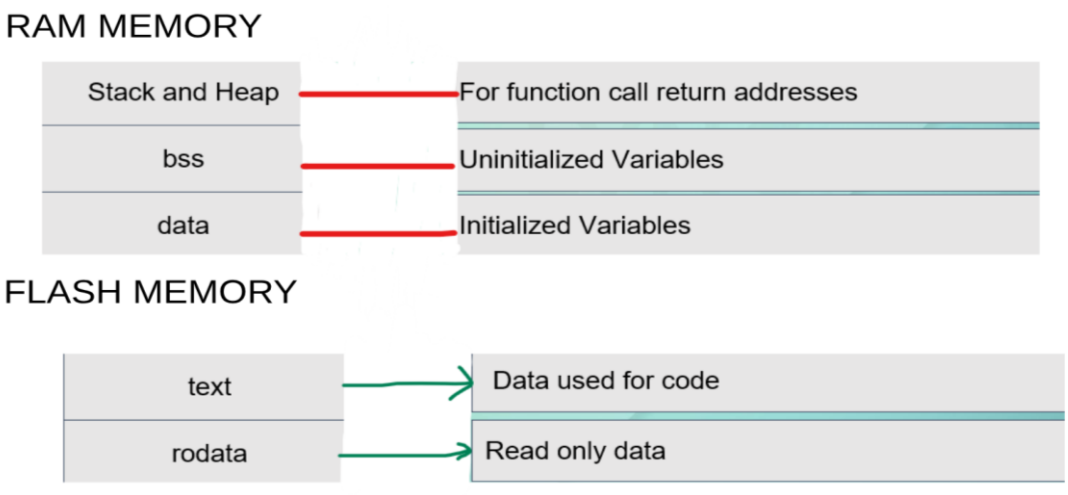
Memory Block in Micro-controller

Flash(Program Memory):

Flash memory is a long-life and non-volatile storage chip that is widely used in embedded systems. It can keep stored data and information even when the power is off. It can be electrically erased and reprogrammed. Flash memory was developed from EEPROM (electronically erasable programmable read-only memory).

Text : the data used for code is stored.

RoM data: it's a Read only memory



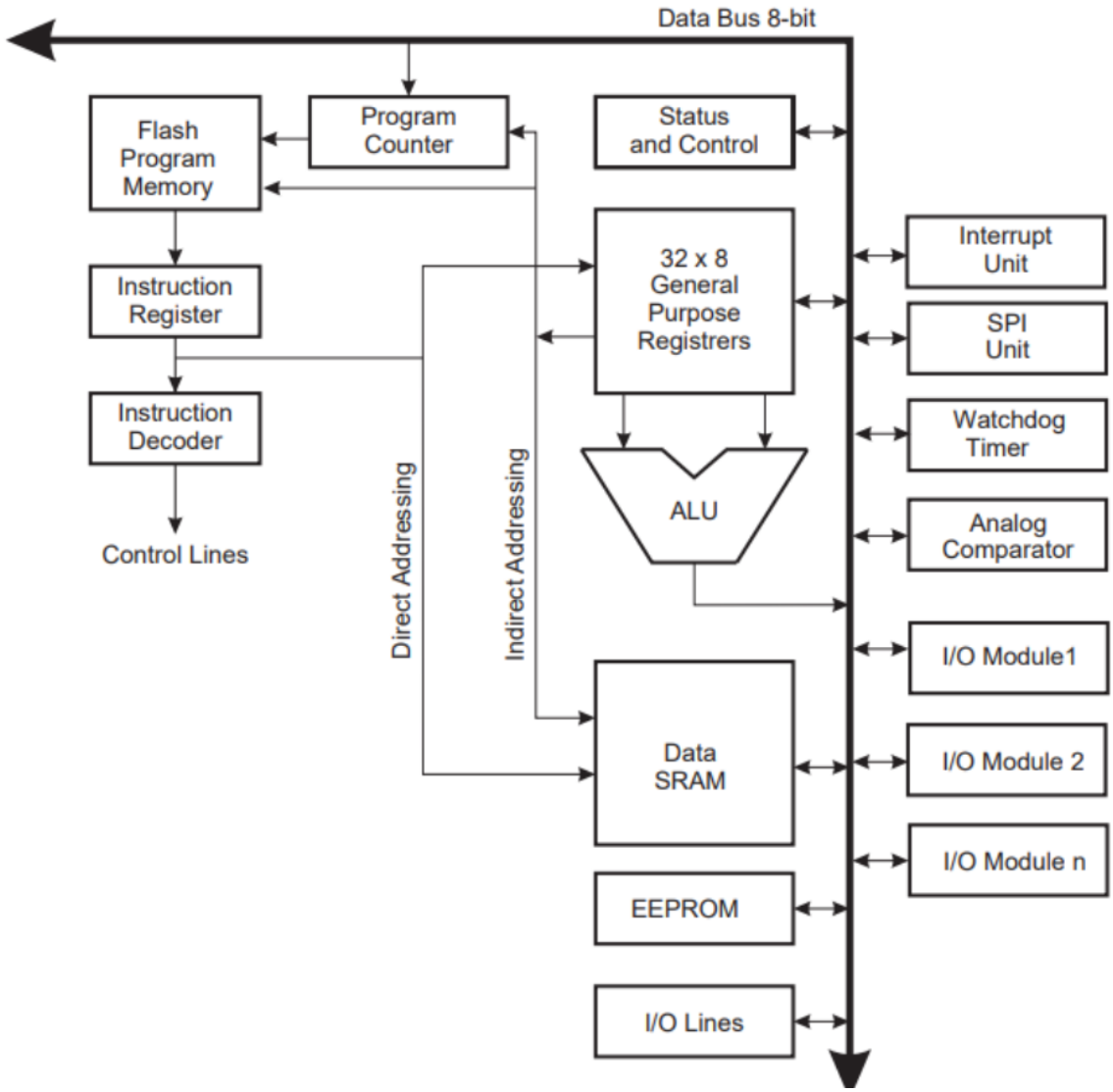
Stack and heap memory:

Stack stores data of functions to keep track of returning from function calls, however, heap is used to store data that is dynamically allocate by the user in the program i.e. using malloc or calloc.

Bss(uninitialized data segment):
Bss-Block started by symbol,it contains object file where all statically allocated variables are stored.

Data(Initialised data segment):
Storing the program data, used for temporarily storing constant & variables values that are used by microcontroller during normal program execution.

Block Diagram of AVR-Microcontroller



Architecture Detail

Data BUS 8 bit :
Its an 8-bit parallel data lines by which the data travels inside the MCU (NOTE: this is the reason why AVR is an 8-bit MCU).

ALU:
Arithmetic Logic Unit, the core of the entire system where typically all commands get executed. Its is part of CPU that carries all arithmetic and logic operations on operands in computer instructions

Data SRAM:
It is similar to the RAM (Random access memory) we see inside our computers, it retains(hold) the data in its memory as long as power is supplied to it.

EEPROM: (Electrically Erasable Programmable Read Only Memory):
It is very similar to another component in our computer namely HardDisk, i.e. a permanent storage.

I/O lines:
These are the bunch of registers which is used as a switches or controls for different features of AVR.

32X8 GPR (General Purpose Registers):
This are 32 registers each having 8-bit which is a general storage space for data. But, remember, SRAM is also a temporary storage but these registers have some specialty among all.

Status Register & Control Register :
The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations.

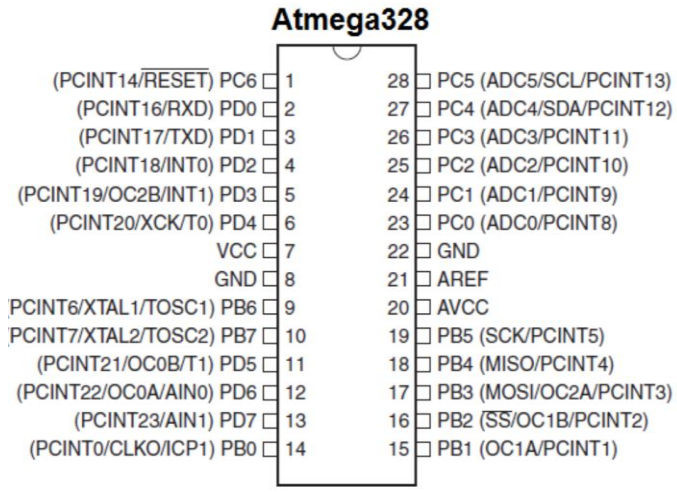
Program Counter:
It is 14 bit, this is a register which has a responsibility to track the position of the program that is currently executing.

Flash Memory:
It is also a permanent storage but its only for storing the program we write to it.
This memory retains data in the absence of power supply.

Instruction Register & Decoder:
The instruction copied from program phase and transferred to decoder
The decoder will fetched instructions and decode the current program and read by Micro-controller

Watchdog Timer:
In this micro-controller, the watchdog timer is present with an internal oscillator. The main function of this is to monitor and reset the controller continuously if the code gets trapped while executing in a defined time interval.

PIN Diagram of ATMega328



ATMEGA328 is an 8-bit microcontroller based on AVR RISC architecture. It is the most popular of all AVR controllers as it is used in ARDUINO boards.

ATmega328 is a high performance yet low power consumption 8-bit AVR microcontroller that's able to achieve the most single clock cycle execution of 131 powerful instructions which follows advanced RISC architecture. It can commonly be found as a processor in Arduino boards such as Arduino Fio and Arduino Uno

PIN Details of ATMega 328

Pin No.	Pin name	Description	Secondary Function
1	PC6 (RESET)	Pin6 of PORTC	Pin by default is used as RESET pin. PC6 can only be used as I/O pin when RSTDISBL Fuse is programmed.
2	PD0 (RXD)	Pin0 of PORTD	RXD (Data Input Pin for USART) USART Serial Communication Interface [Can be used for programming]
3	PD1 (TXD)	Pin1 of PORTD	TXD (Data Output Pin for USART) USART Serial Communication Interface [Can be used for programming] INT2(External Interrupt 2 Input)
4	PD2 (INT0)	Pin2 of PORTD	External Interrupt source 0
5	PD3 (INT1/OC2B)	Pin3 of PORTD	External Interrupt source1 OC2B(PWM - Timer/Counter2 Output Compare Match B Output)
6	PD4 (XCK/T0)	Pin4 of PORTD	T0(Timer0 External Counter Input) XCK (USART External Clock I/O)
7	VCC		Connected to positive voltage
8	GND		Connected to ground
9	PB6 (XTAL1/TOSC1)	Pin6 of PORTB	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
10	PB7 (XTAL2/TOSC2)	Pin7 of PORTB	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
11	PD5 (T1/OC0B)	Pin5 of PORTD	T1(Timer1 External Counter Input) OC0B(PWM - Timer/Counter0 Output Compare Match B Output)
12	PD6 (AIN0/OC0A)	Pin6 of PORTD	AIN0(Analog Comparator Positive I/P) OC0A(PWM - Timer/Counter0 Output Compare Match A Output)
13	PD7 (AIN1)	Pin7 of PORTD	AIN1(Analog Comparator Negative I/P)
14	PB0 (ICP1/CLKO)	Pin0 of PORTB	ICP1(Timer/Counter1 Input Capture Pin) CLKO (Divided System Clock. The divided system clock can be output on the PB0 pin)
15	PB1 (OC1A)	Pin1 of PORTB	OC1A (Timer/Counter1 Output Compare Match A Output)

16	PB2 (SS/OC1B)	Pin2 of PORTB	SS (SPI Slave Select Input). This pin is low when controller acts as slave. [Serial Peripheral Interface (SPI) for programming] OC1B (Timer/Counter1 Output Compare Match B Output)
17	PB3 (MOSI/OC2A)	Pin3 of PORTB	MOSI (Master Output Slave Input). When controller acts as slave, the data is received by this pin. [Serial Peripheral Interface (SPI) for programming] OC2 (Timer/Counter2 Output Compare Match Output)
18	PB4 (MISO)	Pin4 of PORTB	MISO (Master Input Slave Output). When controller acts as slave, the data is sent to master by this controller through this pin. [Serial Peripheral Interface (SPI) for programming]
19	PB5 (SCK)	Pin5 of PORTB	SCK (SPI Bus Serial Clock). This is the clock shared between this controller and other system for accurate data transfer. [Serial Peripheral Interface (SPI) for programming]
20	AVCC		Power for Internal ADC Converter
21	AREF		Analog Reference Pin for ADC

22	GND		GROUND
23	PC0 (ADC0)	Pin0 of PORTC	ADC0 (ADC Input Channel 0)
24	PC1 (ADC1)	Pin1 of PORTC	ADC1 (ADC Input Channel 1)
25	PC2 (ADC2)	Pin2 of PORTC	ADC2 (ADC Input Channel 2)
26	PC3 (ADC3)	Pin3 of PORTC	ADC3 (ADC Input Channel 3)
27	PC4 (ADC4/SDA)	Pin4 of PORTC	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/output Line)
28	PC5 (ADC5/SCL)	Pin5 of PORTC	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)

Features of ATMega328

- 8-Bit AVR Microcontroller
- Modified Harvard RISC Architecture
- 32KB Flash Memory
- 1KB EEPROM
- 2KB SRAM
- Two 8-bit Timer/Counters
- One 16-bit Timer/Counter
- Six PWM Channels
- Eight 10-Bit ADC Channels in Atmega32
- Six 10-Bit ADC Channels in Atmega328
- USART, SPI and I2C Interfaces
- Watchdog Timer, Pin Change Interrupt and Wake-up
- Power-on Reset, Internal and External Interrupts
- Operating Voltage:1.8V to 5.5V for 0 – 4MHz, 2.7V to 5.5V for 0 – 10MHz and 4.5V to 5.5V for 0 – 20MHz speed grades.
- Active Mode Power Consumption of 0.2mA at 1.8V and 1MHz
- Power Down Mode Consumption of 0.1µA at 1.8V and 1MHz

Software's used:
Simul IDE → https://www.simulide.com/p/downloads.html?m=1 (use this version only 0.4.15 - SR10)

Arduino IDE --> https://www.arduino.cc/en/software (use 1.8.19 version)

MICROCHIP -> https://www.microchip.com/en-us/tools-resources/develop/microchip-studio (download the offline installer)