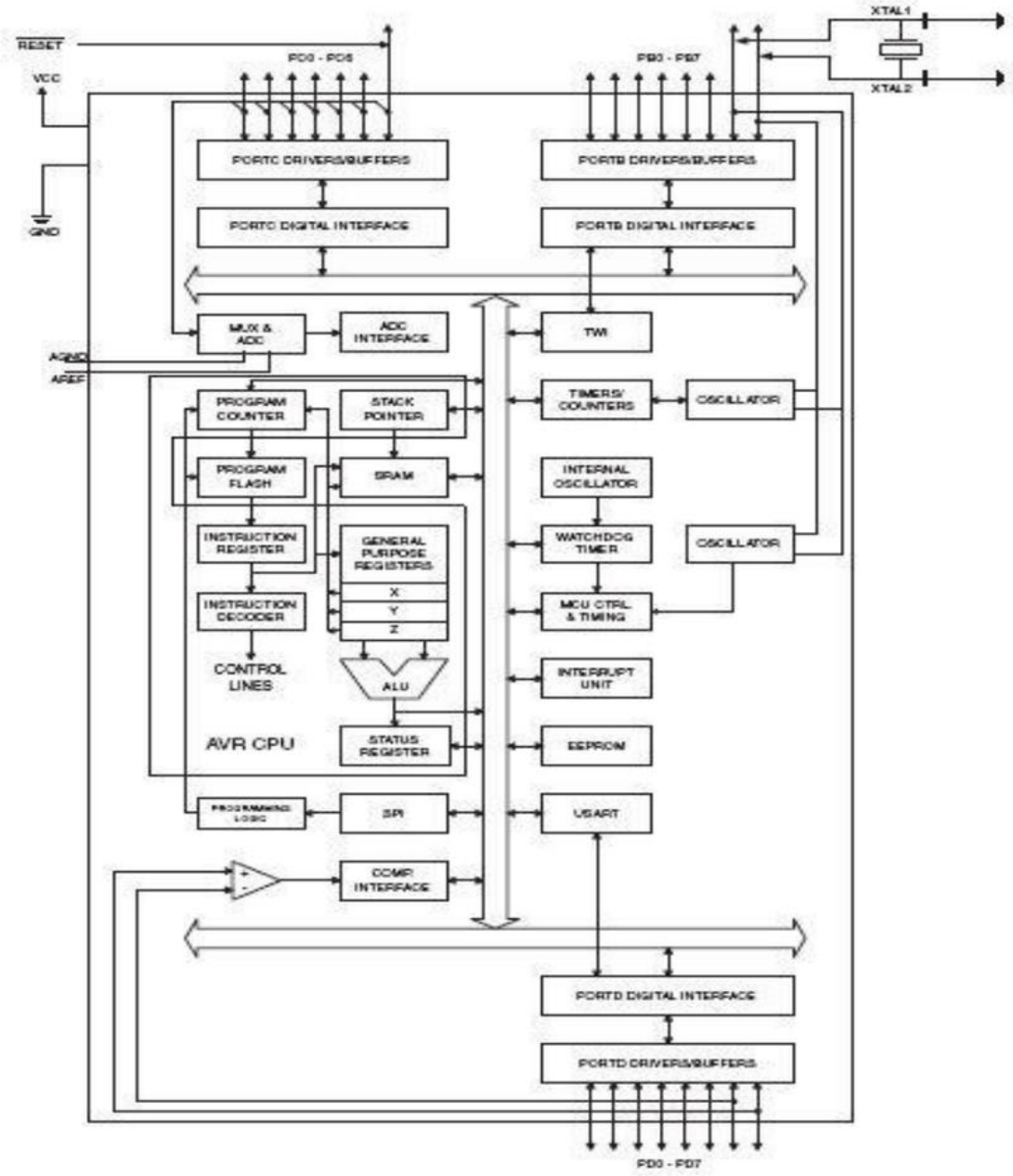
EE2016 Microprocessor Laboratory

Introduction to AVR Microprocessors

The **AVR** is a modified Harvard architecture 8-bit RISC single-chip microcontroller, which was developed by Atmel in 1996. The AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

Block Diagram of Atmega8 AVR processor



Features

AVRs offer a wide range of features:

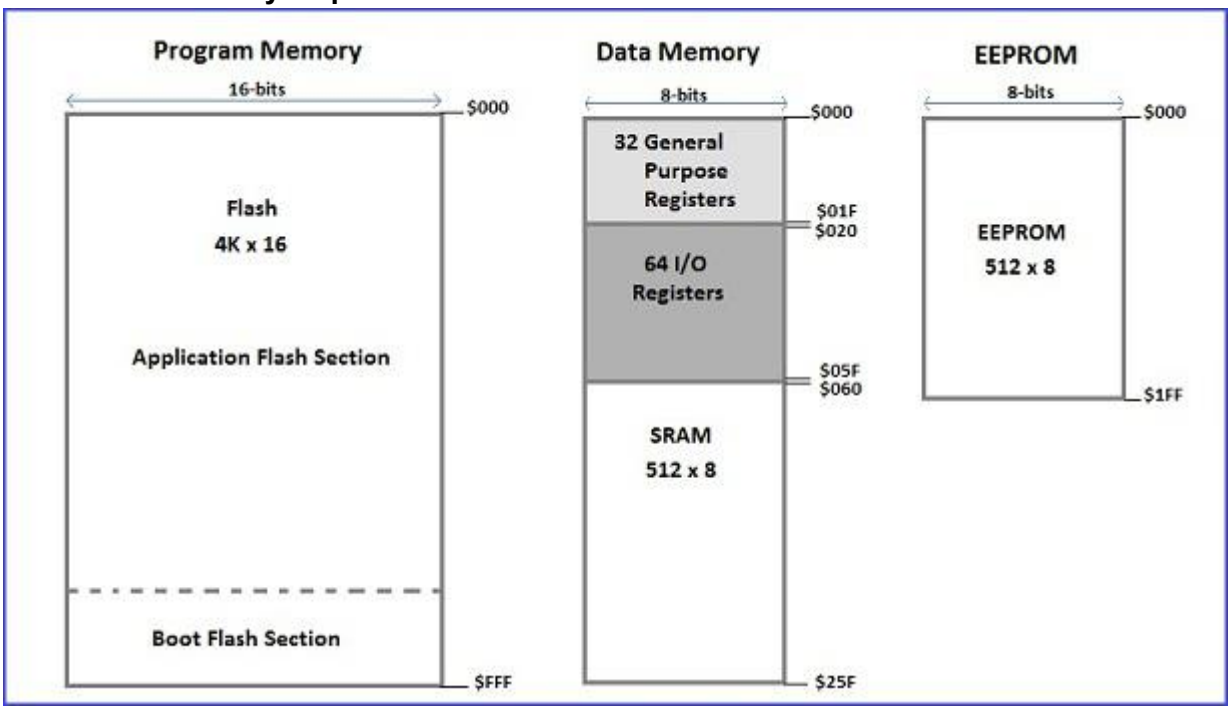
- Multifunction, bi-directional general-purpose I/O ports with configurable, built-in pull-up resistors
- Multiple internal oscillators, including RC oscillator without external parts
- Internal, self-programmable instruction flash memory up to 256 KB (384 KB on XMega)
    - In-system programmable using serial/parallel low-voltage proprietary interfaces or JTAG
    - Optional boot code section with independent lock bits for protection
- Internal data EEPROM up to 4 KB
- Internal SRAM up to 16 KB (32 KB on XMega)
- External 64 KB little endian data space on certain models
- 8-bit and 16-bit timers
    - PWM output
    - Input capture that record a time stamp triggered by a signal edge
- Analog comparator
- 10 or 12-bit A/D converters, with multiplex of up to 16 channels
- 12-bit D/A converters
- A variety of serial interfaces, including
    - I²C compatible Two-Wire Interface (TWI)
    - Synchronous/asynchronous serial peripherals (UART/USART) (used with RS-232, RS-485, and more)
    - Serial Peripheral Interface Bus (SPI)
    - Universal Serial Interface (USI): a multi-purpose hardware communication module that can be used to implement an SPI,[10] I²C[11][12] or UART[13] interface.
- Brownout detection
- Watchdog timer (WDT)
- Multiple power-saving sleep modes
- Lighting and motor control (PWM-specific) controller models
- CAN controller support
- USB controller support
    - Proper full-speed (12 Mbit/s) hardware & Hub controller with embedded AVR.
    - Also freely available low-speed (1.5 Mbit/s) (HID) bitbanging software emulations
- Ethernet controller support
- LCD controller support
- Low-voltage devices operating down to 1.8 V (to 0.7 V for parts with built-in DC–DC upconverter)
- picoPower devices
- DMA controllers and "event system" peripheral communication.
- Fast cryptography support for AES and DES

Device architecture

Flash, EEPROM, and SRAM are all integrated onto a single chip, removing the need for external memory in most applications.

Program instructions are stored in non-volatile flash memory. Although the MCUs are 8-bit, each instruction takes one or two 16-bit words.
There is no provision for off-chip program memory; all code executed by the AVR core must reside in the on-chip flash

AVR Memory Map



.

Internal data memory

The data address space consists of the register file, I/O registers, and SRAM.
The AVRs have 32 single-byte registers and are classified as 8-bit RISC devices.

GPIO ports

Each GPIO port on a tiny or mega AVR drives up to eight pins and is controlled by three 8-bit registers: DDR*x*, PORT*x* and PIN*x*, where *x* is the port identifier.

- DDR*x*: Data Direction Register, configures the pins as either inputs or outputs.
- PORT*x*: Output port register. Sets the output value on pins configured as outputs. Enables or disables the pull-up resistor on pins configured as inputs.
- PIN*x*: Input register, used to read an input signal.

EEPROM

Almost all AVR microcontrollers have internal EEPROM for semi-permanent data storage. Like flash memory, EEPROM can maintain its contents when electrical power is removed.

Program execution

Atmel's AVRs have a two-stage, single-level pipeline design. This means the next machine instruction is fetched as the current one is executing. Most instructions take just one or two clock cycles, making AVRs relatively fast among eight-bit microcontrollers.
The AVR processors were designed with the efficient execution of compiled C code in mind and have several built-in pointers for the task.

Instruction set

- Pointer registers X, Y, and Z have addressing capabilities that are different from each other.
- Register locations R0 to R15 have different addressing capabilities than register locations R16 to R31.
- I/O ports 0 to 31 have different addressing capabilities than I/O ports 32 to 63.
- Accessing read-only data stored in the program memory (flash) requires special LPM instructions; the flash bus is otherwise reserved for instruction memory.

MCU speed

The AVR line can normally support clock speeds from 0 to 20 MHz, with some devices reaching 32 MHz. AVRs feature an on-chip oscillator, removing the need for external clocks or resonator circuitry.
Since all operations (excluding multiplication and 16-bit add/subtract) on registers R0–R31 are single-cycle, the AVR can achieve up to 1 MIPS per MHz, i.e. an 8 MHz processor can achieve up to 8 MIPS. Loads and stores to/from memory take two cycles, branching takes two cycles.

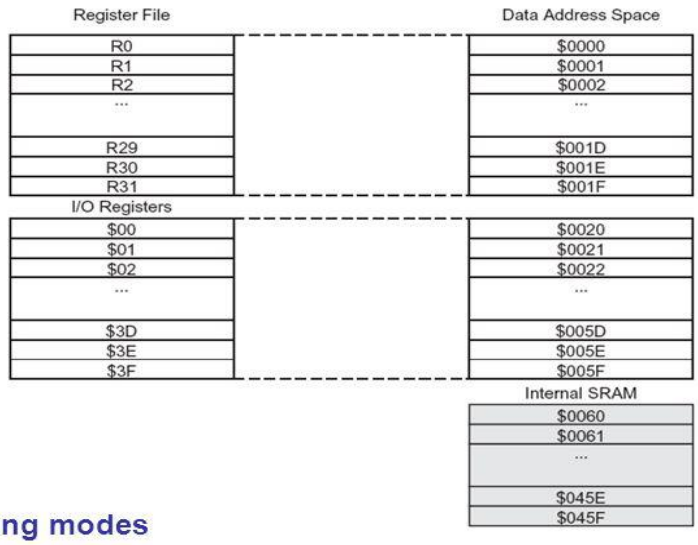
AVR Memory organization:

- **SRAM Data Memory:**

  32 GPR's and
  64 SFR's mapped
  to SRAM memory space

  SFR's accessed
  via in / out instructions
  (I/O-registers)

  1 Kbytes of internal
  SRAM can be accessed
  from address 0x060
  to address 0x45f



5 Direct and indirect addressing modes