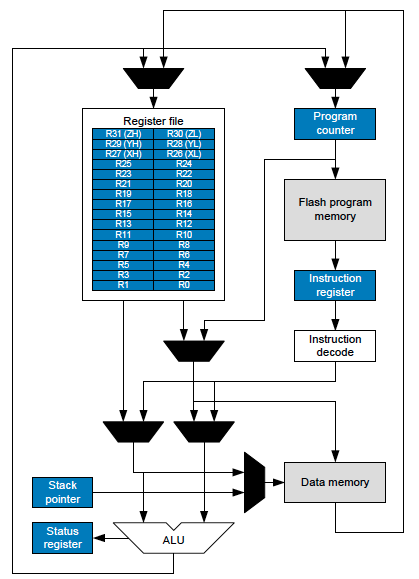
The main function of the AVR® Central Processing Unit (CPU) core is to ensure correct program execution. The CPU must, therefore, be able to access memories, perform calculations, control peripherals, and handle interrupts.



AVR Core

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 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](https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8-bit-avr/structure/instruction-timing/). The program memory is In-System Reprogrammable Flash memory.

Registers

The fast-access Register file contains 32 x 8-bit [General Purpose Working registers](https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8-bit-avr/structure/gpr/) with a single clock cycle access time. 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 lookup tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-registers.

Arithmetic Logic Unit (ALU)

The [ALU](https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8-bit-avr/structure/alu/) supports arithmetic and logic operations between registers or between a constant and a register. Single clock cycle access time allows single-cycle ALU operations. In a typical ALU operation, two operands are output from the Register file, the operation is executed and the result is stored back in the Register file in one clock cycle. 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. Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Memory

The memory spaces in the AVR architecture are all linear and regular memory maps.

Program Flash memory space is divided into two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The Store Program Memory (SPM) instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the [stack](https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8-bit-avr/structure/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.

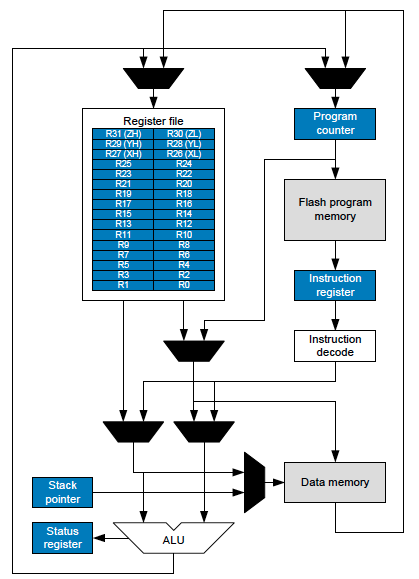
All user programs must initialize the Stack Pointer (SP) in the Reset routine (before subroutines or interrupts are executed). The SP is read/write  
accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control registers, Serial Peripheral Interface (SPI), and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register file, 0x20 - 0x5F. In addition, this device has extended I/O space from 0x60 - 0xFF in SRAM.

Interrupts

A flexible [Interrupt module](https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8-bit-avr/structure/interrupts/) has its Control registers in the I/O space with an additional Global Interrupt Enable bit in the [Status register](https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8-bit-avr/structure/status/). 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.

*La función principal del núcleo de la Unidad Central de Procesamiento (CPU) AVR® es garantizar la correcta ejecución del programa. Por lo tanto, la CPU debe poder acceder a memorias, realizar cálculos, controlar periféricos y manejar interrupciones.*



Núcleo AVR

**Núcleo AVR**

*Para maximizar el rendimiento y el paralelismo, el AVR utiliza una arquitectura Harvard con memorias y buses separados para programas y datos. Las instrucciones en la memoria del programa se ejecutan con canalización de un solo nivel. Mientras se ejecuta una instrucción, la siguiente instrucción se recupera previamente de la memoria del programa. Este concepto permite ejecutar instrucciones en cada ciclo de reloj. La memoria del programa es una memoria Flash reprogramable en el sistema.*

**Registros**

*El archivo de registro de acceso rápido contiene 32 registros de trabajo de uso general de 8 bits con un tiempo de acceso a un único ciclo de reloj. Seis de los 32 registros se pueden utilizar como tres punteros de registro de direcciones indirectas de 16 bits para el direccionamiento del espacio de datos, lo que permite cálculos de direcciones eficientes. Uno de estos punteros de dirección también se puede utilizar como puntero de dirección para tablas de búsqueda en la memoria del programa Flash. Estos registros de función agregados son los registros X, Y y Z de 16 bits.*

**Unidad Aritmética Lógica (ALU)**

*La ALU admite operaciones aritméticas y lógicas entre registros o entre una constante y un registro. El tiempo de acceso al ciclo de reloj único permite operaciones de ALU de ciclo único. En una operación típica de ALU, se generan dos operandos desde el archivo de registro, la operación se ejecuta y el resultado se almacena nuevamente en el archivo de registro en un ciclo de reloj. Las operaciones de registro único también se pueden ejecutar en la ALU. Después de una operación aritmética, el registro de estado se actualiza para reflejar información sobre el resultado de la operación. El flujo del programa lo proporcionan instrucciones de salto y llamada condicionales e incondicionales, capaces de abordar directamente todo el espacio de direcciones. La mayoría de las instrucciones AVR tienen un único formato de palabra de 16 bits. Cada dirección de memoria de programa contiene una instrucción de 16 o 32 bits.*

**Memoria**

*Los espacios de memoria en la arquitectura AVR son todos mapas de memoria lineales y regulares.*

*El espacio de la memoria Flash del programa se divide en dos secciones: la sección del programa de inicio y la sección del programa de aplicación. Ambas secciones tienen bits de bloqueo dedicados para protección contra escritura y lectura/escritura. La instrucción Store Program Memory (SPM) que escribe en la sección de memoria Flash de la aplicación debe residir en la sección Boot Program.*

*Durante las interrupciones y llamadas de subrutinas, la dirección de retorno del Contador de programa (PC) se almacena en la pila. La pila se asigna efectivamente en la SRAM de datos generales y, en consecuencia, el tamaño de la pila solo está limitado por el tamaño total de la SRAM y el uso de la SRAM.*

*Todos los programas de usuario deben inicializar el puntero de pila (SP) en la rutina de reinicio (antes de que se ejecuten subrutinas o interrupciones). El SP es de lectura/escritura.*

*accesible en el espacio de E/S. Se puede acceder fácilmente a la SRAM de datos a través de los cinco modos de direccionamiento diferentes admitidos en la arquitectura AVR.*

*El espacio de memoria de E/S contiene 64 direcciones para funciones periféricas de la CPU como registros de control, interfaz periférica serie (SPI) y otras funciones de E/S. Se puede acceder a la memoria de E/S directamente o como ubicaciones de espacio de datos después de las del archivo de registro, 0x20 - 0x5F. Además, este dispositivo tiene espacio de E/S ampliado de 0x60 - 0xFF en SRAM.*

interrupción

*Un módulo de interrupción flexible tiene sus registros de control en el espacio de E/S con un bit de habilitación de interrupción global adicional en el registro de estado. Todas las interrupciones tienen un vector de interrupción separado en la tabla de vectores de interrupción. Las interrupciones tienen prioridad de acuerdo con su posición del vector de interrupción. Cuanto menor sea la dirección del vector de interrupción, mayor será la prioridad.*