

Optimización del código C en AVR de 8 bits

Antes de optimizar el software de los sistemas integrados, es necesario comprender bien cómo está estructurado el núcleo del microcontrolador (MCU) **AVR**® y qué estrategias utiliza AVR GNU Compiler Collection (GCC) para generar código eficiente para el procesador.

Descripción general de la optimización

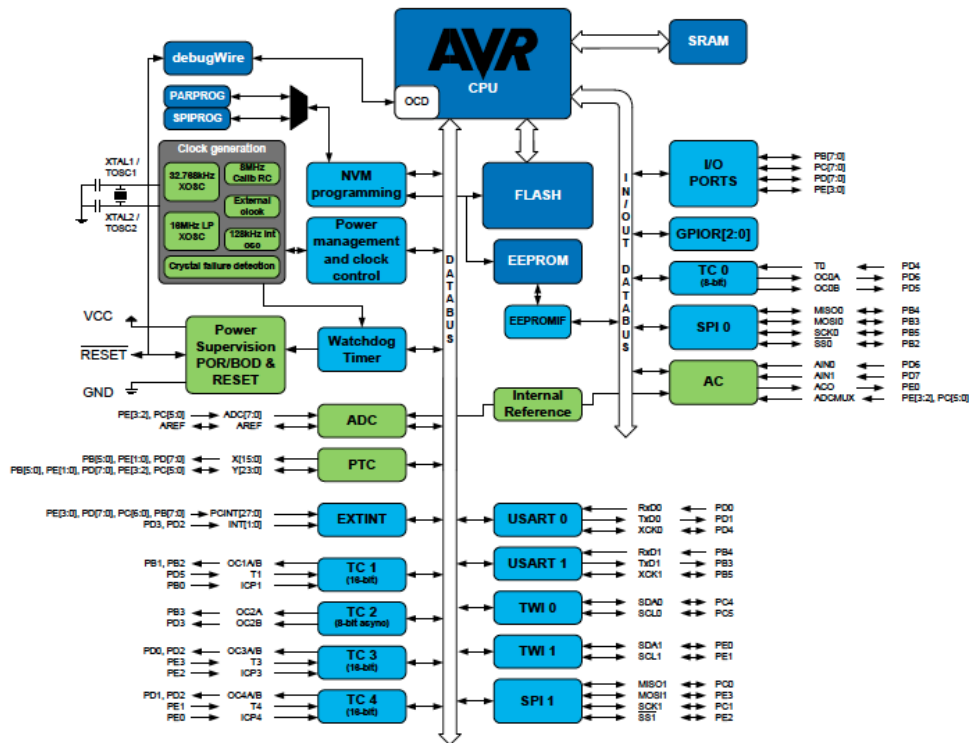
AVR® Insights - Episode 10 - Optimization of C Code on AVR MC...



Arquitectura Atmel AVR de 8 bits

AVR utiliza la arquitectura Harvard, con memorias y buses separados para programas y datos. Tiene un archivo de registro de acceso rápido de 32 x 8 registros de trabajo de propósito general con un solo tiempo de acceso de ciclo de reloj. Los 32 registros de trabajo son una de las claves para una codificación C eficiente. Estos registros tienen la misma función que el acumulador tradicional, excepto que hay 32 de ellos. Las

instrucciones aritméticas y lógicas del AVR funcionan en estos registros, por lo que ocupan menos espacio de instrucción. En un ciclo de reloj, AVR puede enviar dos registros arbitrarios desde el archivo de registro a la ALU, realizar una operación y volver a escribir el resultado en el archivo de registro.



(/local--files/8avr:memory/avrbus.png)

Las instrucciones en la memoria del programa se ejecutan con una canalización de un solo nivel. Mientras se ejecuta una instrucción, la siguiente instrucción se obtiene previamente de la memoria del programa. Este concepto permite ejecutar instrucciones en cada ciclo de reloj. La mayoría de las instrucciones AVR tienen un solo formato de palabra de 16 bits. Cada dirección de memoria de programa contiene una instrucción de 16 o 32 bits.

AVR CCG

AVR GCC proporciona varios niveles de optimización. Son -O0, -O1, -O2, -O3 y -Os. En cada nivel, hay diferentes opciones de optimización habilitadas, excepto -O0 que significa que no hay optimización. Además de las opciones habilitadas en los niveles de optimización, también puede habilitar opciones de optimización separadas para obtener una optimización específica. El manual de la colección de compiladores GNU (<http://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html>) tiene una lista completa de opciones y niveles de optimización.

Aparte del AVR GCC, se necesitan muchas otras herramientas trabajando juntas para producir la aplicación ejecutable final para el microcontrolador AVR. El grupo de herramientas se denomina cadena de herramientas. Dentro de la cadena de herramientas AVR, AVR Libc Library, que proporciona muchas de las mismas funciones

que se encuentran en una biblioteca C estándar y muchas funciones de biblioteca adicionales específicas para un AVR. Además, la biblioteca proporciona el código de inicio básico que necesitan la mayoría de las aplicaciones. Consulte el Manual de referencia de AVR Libc (<https://www.microchip.com/webdoc/AVRLibcReferenceManual/>) para obtener más detalles.

Información Adicional

- Nota de aplicación AVR4027: consejos y trucos para optimizar su código C para microcontroladores AVR de 8 bits
(<http://ww1.microchip.com/downloads/en/AppNotes/doc8453.pdf>)
- Nota de aplicación AVR035 - Codificación C eficiente para AVR
(<http://ww1.microchip.com/downloads/en/AppNotes/doc1497.pdf>)