

HAL/GNU: Using Register in Inline Assembly

The HAL/GNU C Compiler allows us to use inline assembly in the C source code. In order to avoid clobbering registers used in particular function by the C compiler, we can declare the registers as global or local registers. This paper will provide sample codes for using registers in inline assembly code.

/* The compiler can be prevented to use any registers by declaring them as global. */

```
register int dummy0 asm("r2");
register int dummy1 asm("r3");
register int dummy2 asm("r4");
```

```
func()
{
    int i;
    asm("mov.w    #0,r2");
    asm("mov.w    #0,r3");
    asm("mov.w    #0,r4");
    for (i = 0; i < 10; i++)
    {
        foo();
        asm("adds  #1,r2");
        asm("add.w  r2,r3");
        asm("add.w  r3,r4");
    }
}
```

/* The compiler can be prevented to use any registers by declaring them as local. If they are not used, the compiler will optimize them out and reuse the registers. */

```
func()
{
    int i;
    register int r2_var asm("r2");
    register int r3_var asm("r3");
    register int r4_var asm("r4");

    asm("mov.w    #0,%0" : "=r" (r2_var) );
    asm("mov.w    #0,%0" : "=r" (r3_var) );
    asm("mov.w    #0,%0" : "=r" (r4_var) );
    for (i = 0; i < 10; i++)
    {
        foo();
        asm("adds  #1,%0" : "=r" (r2_var));
        asm("add.w  %1,%0" : "=r" (r3_var) : "r" (r2_var));
        asm("add.w  %1,%0" : "=r" (r4_var) : "r" (r3_var));
    }
    bar(r2_var, r3_var, r4_var);
}
```

The information in this document has been carefully checked; however, the contents of this document may be changed and modified without notice. Hitachi America, Ltd. shall assume no responsibility for inaccuracies, or any problem involving a patent infringement caused when applying the descriptions in this document. This material is protected by copyright laws. © Copyright 1995, Hitachi America, Ltd. All rights reserved. Printed in U.S.A.