

HAL/GNU: Storing Variable Value into Register

Using the HAL/GNU H8/300 C Compiler, we can store a variable value into a register. This paper will provide a sample program in storing the variable value into a register. There are comments in the C source code for explanation.

The following is the C source code called ValToReg.c (ValueIntoRegister):

```
#define VALUE 9
void (*(functiontable[]))();
register int *table asm("r3");
int j;

p0()
{
    /* Using global register variables */
    j = table[9];
}

p1()
{
    /* Using extended asm */
    void (*address)() = functiontable[9];

    /* the asm statement takes operand 0, which is VALUE, and uses it
       as an integer 'i' */
    asm("mov.w    %0,r0" : : "i" (VALUE));

    /* this asm statement takes operand 0, which is address, and
       uses it like a register 'r' */
    asm("mov.w    %0,r2" : : "r" (address));

    /* jump to r2 */
    asm("jmp      @r2" );
}

p2()
{
    /* using explicit registers for variables and
       extended asm */
    register void *address asm("r2");
    register int arg1 asm("r0");

    /* load functiontable+9 into address, which is stored in r2 */
    address = functiontable[9];

    /* load VALUE into arg1, which is stored in r0 */
    arg1 = VALUE;

    /* jump through operand 0 - which is address, and use it like a
       register. Note that arg1 is also provided, so that the optimizer
       doesn't delete the the assignment of VALUE to arg1 */
}
```

```

asm("jmp @%0" : : "r" (address) , "r" (arg1));
}
p3()
{
int stackloc[10];

/* move the the address of stackloc into r0 */
asm("mov.w    %0,r0" : : "r" (&stackloc));

/* call a function */
foo();
}

```

We can compile and assemble the ValToReg.c file to produce the assembly listing. The following commands will compile and assemble ValToReg:

GCC -g -S ValToReg.c

AS -al ValToReg.s > ValToReg.lst

where

GCC	name of the compiler
-g	compiler switch to generate debugging information
-S	compiler switch to generate assembly source file
AS	name of the assembler
-al	assembler switch to produce assembly listing on standard output
>	redirect the output listing to ValToReg.lst

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.