

### HAL/GNU: Storing Variable Address into Register

Using the HAL/GNU H8/300 C Compiler, we can store variable addresses into registers. This paper will provide a sample program in storing the variable addresses into registers.

The basic format list is:

	nothing print naturally
'w'	print the low byte of 32bit value
'x'	print the 2nd byte of 32bit value
'y'	print the 3rd byte of 32bit value
'z'	print the 4th byte of 32bit value
'f'	print the low word of a 32 bit value
'e'	print the high word of a 32 bit value

The basic constraint list is:

'r'	use as a register
'i'	use as an immediate value
'o'	use as an offsetable memory location

The following is the sample C source code called AddToReg.c (AddressIntoRegister):

```
int a[10];
int *foo;
m()
{
    register long tmp0 asm("r0");

    asm("mov.w    %1,%0" : "=r" (tmp0) : "i" (&a[4]));
    tmp0 += 9;

    asm("mov.b    %w1,%0" : "=o" (foo[1]) : "r" (tmp0));
    asm("mov.b    %x1,%0" : "=o" (foo[2]) : "r" (tmp0));
    asm("mov.b    %y1,%0" : "=o" (foo[3]) : "r" (tmp0));
    asm("mov.b    %z1,%0" : "=o" (foo[4]) : "r" (tmp0));
}

        .global _m
_m:
; #APP
        mov.w    #_a+8,r0
; #NO_APP
        add      #9,r1l
        addx     #0,r1h
        addx     #0,r0l
        addx     #0,r0h
        mov.w    @_foo,r4
; #APP
        mov.b    r1l,@(2,r4)
        mov.b    r1h,@(4,r4)
```

```
    mov.b    r0l,@(6,r4)
    mov.b    r0h,@(8,r4)

; #NO_APP
    rts
    .comm _a,20
    .comm _foo,2
    .end
```

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.