The SuperH-3, part 9: Constants

devblogs.microsoft.com/oldnewthing/20190815-00

August 15, 2019

Raymond Chen

Loading constants on the SH-3 is a bit of a pain. We saw that the `MOV` instruction supports an 8-bit signed immediate, but what if you need to load something outside that range?

The assembler allows you to write this:

```
MOV     #value, Rn      ; load constant into Rn
```

If the value fits in an 8-bit signed immediate, then it uses that. Otherwise, it chooses a PC-relative `MOV.W` or `MOV.L` depending on the size of the value, and it generates the constant into the code at a point it believes that the code is unreachable, such as two instructions after a `bra` or `rts`. If no such point can be found, the assembler raises an error. You can use the `.nopool` directive to prevent constants from being generated at a particular point, or `.pool` to force them to be generated.

If the compiler can generate the constant in two instructions, typically by combining an immediate with a shift, then the compiler will tend to prefer the two-instruction version instead of using a constants pool, especially if it can put the second half of the calculation into an otherwise-wasted branch delay slot. (Yes, we haven’t learned about branch delay slots yet. Be patient.)

```
; for -256 ≤ value < 256, multiples of 2
MOV    #value / 2, Rn
SHLL   Rn

; for -512 ≤ value < 512, multiples of 4
MOV    #value / 4, Rn
SHLL2  Rn

; for -65536 ≤ value < 65536, multiples of 256
MOV    #value / 256, Rn
SHLL8  Rn

; for -16777216 ≤ value < 16777216, multiples of 65536
MOV    #value / 65536, Rn
SHLL16 Rn
```

Other instructions that could be useful for building constants are logical right shift and rotate. I’m not going to write them out, though. Use your imagination.

Now, it may seem cumbersome to have to use two instructions to generate a constant, but remember that these instructions are only 16 bits in size, so you can fit two of them in the space of a single MIPS, PowerPC, or Alpha AXP instruction. And if you can schedule the instructions properly, the fact that the SH-3 is dual-issue means that each of the instructions executes in a half-cycle, so the pair of them takes a single cycle, assuming you can schedule another instruction between them.

Next up are the control transfer instructions, and the return of the confusing branch delay slot, but the SH-3 adds more wrinkles to make them even more confusing.

Raymond Chen

Follow