10.14 Instructions unique to SuperH

(Original section[1])

Register 0 plays a special role in SuperH address modes. It can be added to another register to form an address in indirect indexed addressing and PC-relative addressing. R0 is used to load constants to give a larger addressing range than can easily be fit into the 16-bit instructions of the SuperH. R0 is also the only register that can be an operand for immediate versions of AND, CMP, OR, and XOR. Below is a list of the remaining unique details of the SuperH architecture:

- *Decrement and test*—`DT` decrements a register and sets the T bit to 1 if the result is 0.
- *Optional delayed branch*—Although the other embedded RISC machines generally do not use delayed branches (see COD Appendix A (The Basics of Logic Design)), SuperH offers optional delayed branch execution for `BT` and `BF`.
- *Many multiplies*—Depending on whether the operation is signed or unsigned, if the operands are 16 bits or 32 bits, or if the product is 32 bits or 64 bits, the proper multiply instruction is `MULS`, `MULU`, `DMULS`, `DMULU`, or `MUL`. The product is found in the MACL and MACH registers.
- *Zero and sign extension*—Byte or halfwords are either zero-extended (`EXTU`) or sign-extended (`EXTS`) within a 32-bit register.
- *One-bit shift amounts*—Perhaps in an attempt to make them fit within the 16-bit instructions, shift instructions only shift a single bit at a time.
- *Dynamic shift amount*—These variable shifts test the sign of the amount in a register to determine whether they shift left (positive) or shift right (negative). Both logical (`SHLD`) and arithmetic (`SHAD`) instructions are supported. These instructions help offset the 1-bit constant shift amounts of standard shifts.
- *Rotate*—SuperH offers rotations by 1 bit left (`ROTL`) and right (`ROTR`), which set the T bit with the value rotated, and also have variations that include the T bit in the rotations (`ROTCL` and `ROTCR`).
- *SWAP*—This instruction swaps either the high and low bytes of a 32-bit word or the two bytes of the rightmost 16 bits.
- *Extract word* (`XTRCT`)—The middle 32 bits from a pair of 32-bit registers are placed in another register.
- *Negate with carry*—Like `SUBC` (COD Figure D.6.6 (Arithmetic/logical instructions not found in MIPS core …)), except the first operand is 0.
- *Cache prefetch*—Like many of the desktop RISCs (COD Figure D.6.1 (Data transfer instructions not found in MIPS core …), COD Figure D.6.2 (Arithmetic/logical instructions not found in MIPS core …), COD Figure D.6.3 (Control instructions not found in MIPS core …), COD Figure D.6.4 (Floating-point instructions not found in MIPS core …)), SuperH has an instruction (`PREF`) to prefetch data into the cache.
- *Test-and-set*—SuperH uses the older test-and-set (`TAS`) instruction to perform atomic locks or semaphores (see COD Chapter 2 (Instructions: Language of the Computer)). `TAS` first loads a byte from memory. It then sets the T bit to 1 if the byte is 0 or to 0 if the byte is not 0. Finally, it sets the most significant bit of the byte to 1 and writes the result back to memory.

(*1) This section is in original form.