C compilers and development tools simplify DSP assembly-language programming

N Manjunath - January 21, 1999
1. That for large applications with millions of lines of code and applications without a fixed standard, such as 3G Cellular in DSP5600 assembly-tool development, recommends the following procedures:

   - Optimize ASM.
   - Possibility 2:
     - Verify assembly in system.

   For more information, visit TI's DSP third-party Web site (www.ti.com/sc/docs/dsps/develop/3party.htm).

   The C compiler generates efficient code that is optimized for both code density and execution time. The compiler incorporates four levels of state-of-the-art generic and target-specific optimizations. The level of optimization used depends on code generation and target architecture.

   Texas Instruments TMS320C6x DSP features two datapaths—each with 16 32-bit registers—shared by eight functional units. The C6x is a very-long-instruction-word (VLIW) processor; a composite instruction (line), formed by algebraic or mnemonics assemblers, is optimized for time-critical loops. You can also use pretested math, DSP, and C runtime-library routines to shorten your development time.

   With the TMS320C54x DSPs, you can generate code with either an algebraic or a mnemonics/assembler. Tatge says no inherent advantages of either approach, either in the implementation of the algorithm in the assembler or in the programmer.

   Programming DSPs becomes easy in assembler forms, but assembler forms come with a skill factor: the programmer can optimize the code to the fullest extent and achieve maximum performance (Figure 1).

   Figure 1.

   Figure 2.

   The skill factor is ever-present—here is where you need to know a lot about the hardware, how to represent scalar fractional data, redundant assignment elimination, loop-induction-variable optimizations/strength reduction, loop rotation, loop invariant code motion, inline expansion of function calls, register tracking/targeting, and cost-based register allocation.

   Further, the need to track various functional units in a VLIW architecture can rapidly overwhelm a DSP programmer. Furthermore, the need to track various functional units in a VLIW architecture can rapidly overwhelm a DSP programmer.

   In digital-communication applications, FIR or IIR filters lend themselves well to C with intrinsic addressing. In more intensive computations, such as butterflies and convolutions, always suggest themselves as logical candidates for coding in assembly. In digital-communication applications, FIR or IIR filters lend themselves well to C with intrinsic addressing. In more intensive computations, such as butterflies and convolutions, always suggest themselves as logical candidates for coding in assembly.

   The IDE includes access to the SHARC DSP C compiler, C runtime library, assembler, linker, loader, simulator, and profiler (also known as Simulator). These tools work together to efficiently compile and assemble DSP software.

   You can write assembly-language routines as a linear sequence of instructions. You can also combine instructions that are not valid in the linear sequence but are valid at the assembly level. These instructions can be placed in groups. For example, several instructions can be combined with a single mnemonic (Figure 2).