Code optimization

Industrial

Optimize the code of an Action Video game: Action video games are multimedia applications. They require graphics, sound, possibly a network connection, and user input. In a PC, the hardware devices that support these features are completely external to the CPU. In fact, the algorithmic issues of artificial intelligence, collision detection, keeping score, and keeping track of time are ordinarily a very low performance hit on a good CPU. Designing the interaction with the hardware devices will be much more important in terms of the quality of your game, in more ways than just speed. Optimizing the code in such a scenario is very important.

Research

Optimize code for embedded processors: The building blocks of today's and future embedded systems are complex intellectual property components, or cores, many of which are programmable processors. Traditionally, these embedded processors mostly have been programmed in assembly languages due to efficiency reasons. This implies time consuming programming, extensive debugging, and low code portability. The requirements of short time-to-market and dependability of embedded systems are obviously much better met by using high-level language (e.g. C compilers) instead of assembly. However, the use of C compilers frequently incurs a code quality overhead as compared to manually written assembly programs. Due to the need for efficient embedded systems, this overhead must be very low in order to make compilers useful in practice. In turn, this requires code optimization techniques that take the specific constraints in embedded system design into account. An example are the specialized architectures of recent DSP and multimedia processors, which are not yet sufficiently exploited by existing compilers.