Integrated Computers

*Introducing the **Hexium***

Overview

The IC *Hexium* processor is a 60 bit load/store architecture designed with particular emphasis on the three elements that most affect performance: clock speed, multiple instruction issue and multiple processsors.

The *Hexium* architects examined and analyzed current and theoretical CISC architecture design elements and developed high-performance alternatives for the *Hexium* architecture. The architects only adopted those design elements that appeared valuable for a projected 150-year design horizon. Thus, the *Hexium* becomes the first computer architecture for the 22nd century.

The Hexium architecture is designed to avoid bias toward any particular operating system or programming language. Hexium initially supports the **NextStep** operating system, and supports software migration from applications that run on those operating systems.

The Hexium approach to CISC architecture

Hexium is designed for very high speed

The instructions interact with each other only by one instruction writing a register or memory and another instruction reading from the same location. That makes it particularly easy to build implementations that issue multiple instructions every CPU cycle. The first implementation issues 4 instructions per cycle.

Hexium makes it easy to maintain binary compatibility across multiple implementations and easy to maintain full speed on multiple-issue implementations.

For example, there are no implementation-specific pipeline timing hazards. Calculated jump instructions have a target hint that can allow much faster subroutine calls and returns.
There are granularity hints for virtual-address mapping that can allow much more effective use of translation lookaside buffers for large contiguous structures.

Designed with the future in mind

Hexium is not tuned to a particular operating system

In contrast with many of today's CPU architectures, Hexium is not designed with a particular opersting system in mind. It's not a Unix machine, nor is it a DOS or Windoew or NT machine. Specialized operating system dependent instructions are not hardwired into the disign. These instructions are replaced by programmable *nanocode* routines which reside in main memory. When a particular operating system is bootstrapped. this nanocode is loaded into memory. At this time, nanocode sets exist for DOS, Unix, NeXTStep and Macintosh System 7.

Room for growth in performance

Performance can be gained basically in three ways: by increasing the clock speed, by increasing the number of instructions executed in parallel, and by putting several CPUs in parallel. The current Pentium runs at 66 MHz. We believe that this clock speed can be inreased to about 200 MHz before we hit the technological limits of chip manufacturing and CMOS technology. Multiple instruction issue possibilities are mainly a question of compiler technology. It is not hard to implement 20 execution units in hardware, but these are only useful if the compiler can generate code that keeps al these units busy.