Cahners
MICROPROCESSOR REPORT
www.MPRonline.com
THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

INTEL EMBRACES MULTITHREADING

Fall 2001 IDF: Coming-out Party for Jackson Technology

By Kevin Krewell {9/17/01-01}

At the Intel Developer Conference, Fall 2001, Intel announced it will embrace simultaneous multithreading (SMT) in its future processors. The XeonMP (code-named FosterMP) server processor will introduce SMT. Not content to use the industry-accepted term SMT, Intel is calling its version "Hyper-Threading." However, Intel has even grander plans for this technology, beyond servers: the company is planning to extend Hyper-Threading to desktop processors in 2003. Software support for Intel's SMT is currently available in WindowsXP Professional and in Linux Kernel 2.4.1 and later, but the hardware (XeonMP) won't arrive until 2002.

SMT is fast becoming an essential addition to server processor design. Compaq planned to introduce SMT in the now-canceled Alpha EV8 (see *MPR 12/6/99-01*, "Compaq Chooses SMT for Alpha" for a description of the EV8 and of SMT); Sun recently revealed that UltraSPARC V will offer SMT (see *MPR 9/4/01-02*, "900MHz UltraSPARC III Ready to Ship"); and the Power4 processor from IBM will have two complete processors on one die (chip multiprocessing). Each implementation of SMT will be different, but the basic strategy is the same: increase processor resource utilization by executing more than one thread per processor.

The origin of SMT is related to the limitations of instruction-level parallelism. The addition of more functional units to a superscalar design eventually reaches a point of diminishing returns as compilers and schedulers struggle to reorder instructions to simultaneously use the increased processor functional units. Without complex instruction reordering and speculative execution, a significant amount of processor resources can be left idle.

Many years of research into multiprocessing designs have produced reasonable software solutions for distributing tasks over multiple CPUs. Each processor in a multiprocessing system is issued a routine (thread) to execute. SMT leverages the multiprocessing model but takes a middle approach

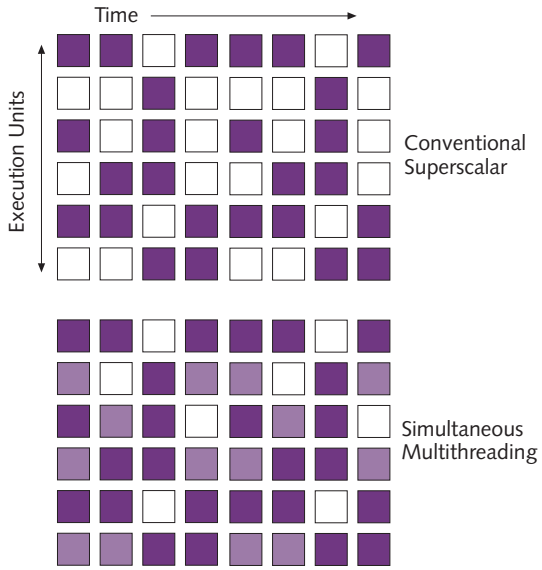

Figure 1. In a wide superscalar design, significant resources can be left idle. (Idle resources in the out-of-order portion of the Pentium 4 are shown in white.) With SMT, the second thread (shown in light purple) can utilize the unused functional units and make more efficient use of resources.

between an individual processor and a multiprocessor system. Instead of duplicating all the processor resources (as in IBM's Power4), SMT duplicates only the information to hold the unique state of each thread. Processor resources are shared with the intention that each thread use different resources during each clock cycle. The goal is that the additional thread will fill voids in the pipeline and take over processor resources when one thread is stalled, such as by waiting on a cache miss (see Figure 1).

Intel's implementation is limited to two threads, each with a unique set of state information, including APIC. The trace cache (decoded instruction cache) has an additional tag bit for each instruction, to represent the thread number. If one thread takes an exception, the instructions from only that thread are flushed. The L1 data cache, the L2 cache, and the L3 cache are unaware of the thread execution and allow data to be shared between the threads (without duplication). The L2 and L3 caches were increased to eight-way to increase cache efficiency, as multiple threads will stress the cache. Intel did not indicate that any additional execution units, additional rename registers, wider issue rate, or larger L1 and L2 caches will be in FosterMP beyond the Willamette microarchitecture.

Each thread is initialized by the operating system as if the thread is running on a separate processor. Active threads are interleaved at the instruction fetch and issue stages of the pipeline. Intel has apparently not enhanced the instruction decoder beyond the present one x86 instruction per cycle and can decode or issue only one thread per clock cycle. The scheduler and out-of-order section of the microarchitecture can intermix threads in each cycle (see *MPR 08/28/00-01*, "Pentium 4 [Partially] Previewed"). Each thread has equal priority, which can be a problem when a low-priority task is running concurrently with a high-priority, time-critical task. Intel might consider refining Hyper-Threading at some time in the future with some sort of priority scheme or the ability to dynamically turn SMT on and off. To refrain from wasting resources with idle loops, operating systems need to use the Halt instruction when a thread goes idle. When using spin-lock code, which aggressively captures processor resources, adding a pause will allow the other thread fair access to processor resources. Exceptions, such as SMI (system management interrupt), will also require action from both threads.

Intel indicated that early testing has shown significant increases in software productivity (work/time) when Hyper-Threading is used. Microsoft's IIS Web-serving software showed the greatest improvement, at 30%. Microsoft's SQL-Server had a 22% improvement, and Exchange server increased performance by 23%. These applications are multithreaded but not specifically optimized for Hyper-Threading. Hyper-Threading may compromise some highly optimized or hand-tuned software, such as that used in games and streaming media, as the highly optimized code will be disrupted by the alternate thread. The interleaved thread will contend for resources and destroy the effectiveness of the tight code optimizations. Both threads may also come into conflict over the precious space in the trace cache. The limited trace cache size has already changed recommended programming optimizations for the Pentium 4: programmers now avoid loop unrolling, as it increases the amount of code created and increases trace cache misses. Keeping tight code loops in the trace cache is key to effective performance on the Pentium 4. A trace cache miss forces the execution to use the x86 instruction decoder, which can decode only one x86 instruction per cycle.

Hyper-Threading, as it increases utilization of processor resources, will also increase the average power dissipation of the processor. Intel has prepared for OEMs a maximum-power test program (often called "Max Power") specifically designed to test the thermal limits of processors with Hyper-Threading. Bus and memory utilization should also increase.

Some minor chip-set changes were also required. Each stop-clock signal (STP_CLK) to the processor requires two stop-grant (STP_GNT) responses from the processor before the clock can be stopped safely. Intel also increased the I/O queue depth to handle additional I/O requests from the processor.

Is Hyper-Threading the Next MMX?

SMT has the potential to be the next must-have technology for server, workstation, and, soon, desktop processors. Although it will take until 2003 for SMT to reach the desktop (and even then it may have limited use), Intel can create market demand for the new technology, putting AMD in a difficult position—unless it has SMT in the Hammer family. The challenge is whether AMD can or will react to Intel's initiative quickly enough. Intel claims that Hyper-Threading has added less than 5% to the die area, but it still takes considerable research and development to know what 5% to add—besides obvious additional state information. We expect Intel to reveal additional design details of Hyper-Threading at the **2001 Microprocessor Forum**.