Rachet Up Reliability for Mission-Critical Applications

Intel® Instruction Replay Technology

The Intel® Itanium® processor family provides the foundation for mainframe-class computing systems designed for today's most demanding, mission-critical environments. The latest Intel Itanium processor 9500² series (code-named Poulson) extends the capabilities of the previous-generation Intel Itanium processor 9300 series to deliver substantial improvements in mission-critical capability.

This white paper focuses on a particularly important new RAS feature in Intel Itanium processor 9500 series: Intel Instruction Replay Technology. This new technology provides enhanced support for automatically detecting and correcting errors in the instruction pipeline of the processor. It not only identifies a wider range of potential errors, but also corrects them almost instantly, so the delay is transparent to the software running on the server platform. Along with the many other RAS improvements in this new processor family, Intel Instruction Replay Technology provides an even stronger foundation for supporting the world's most mission-critical computing requirements.
Improve Uptime for Your Most Critical Applications

In addition to providing major performance gains, this new processor family delivers substantial improvements in reliability, availability, and serviceability (RAS) to support even higher levels of data integrity and system uptime. Hardware-based error prevention, detection, and correction are enhanced and extended throughout the platform. Improvements in firmware add to these advantages, providing expanded coverage of potential error events, along with improved logging for higher availability, faster recovery, and better support for predictive failure. These capabilities work in conjunction with Intel Itanium processors’ complete machine check architecture, which coordinates error handling across hardware, firmware, and operating systems to enable extremely high availability and data integrity.

Advanced Error Correction throughout the Platform

All silicon-based computer chips are vulnerable to ordinary background radiation. An alpha particle can change the value of data in a register or array. Electrical noise and variations in power supplies can have similar impacts (although they rarely do). The longer the data is held, the greater the chance that it will be modified by one of these transient events, resulting in a “soft error.” There are many possible design strategies for dealing with soft errors. The best hardware designs automatically detect and correct for common classes of soft errors to improve data integrity and system availability without requiring firmware, operating system (OS), or application intervention.

The Intel Itanium processor family incorporates extensive features for automatically detecting and correcting soft errors at the hardware level. For example:

• Errors in large caches and arrays are automatically detected and corrected using error correcting code (ECC).
• Errors in smaller caches and various buffers and arrays are detected using parity bits. These transient errors can then be corrected using various forms of “trying again,” which simply means returning to a state prior to the error event and then proceeding as if the error had not occurred.

Next-Generation RAS with Intel® Instruction Replay Technology

The next-generation Intel Itanium processor family provides enhanced support for soft error detection and correction throughout the platform. One of the most important new RAS features is Intel Instruction Replay Technology. This technology provides exceptionally fast recovery from soft errors in one of the most performance-critical areas of the processor: the instruction pipeline. In order to understand how Intel Instruction Replay functions, it is first necessary to understand how the pipeline itself works.

Understanding normal (error-free) pipeline execution

Intel Itanium processors have a memory hierarchy of caches, buffers, and registers that hold the data waiting to be processed and the program instructions waiting to be executed. Software programs held in main memory are executed by bringing the needed portions of the program into the processor’s caches. From there, the instructions are moved into buffers and sent down pipelines to be executed. Data moves in a similar fashion, from main memory, to caches, to buffers, and finally to registers, at which point specific instructions act on specific data.
A microprocessor pipeline is like the conveyor belt of an assembly line (Figure 1). It is divided into a number of stages. At each stage, a different step of the process is performed. Once in the pipeline, the instructions from the software program can act directly on data or other input. They can also direct data to be moved among the memory, cache, and register hierarchy so the right data will be available at the right time to enable efficient processing.

Each core in an Intel Itanium processor includes three main sections:

- The instruction fetch pipeline accesses main memory and brings needed instructions into the processor’s caches. It also identifies the next set of instructions to execute and feeds them into the instruction buffer.
- The instruction execution pipeline reads and executes instructions from the instruction buffer to perform operations on data, such as an addition or multiplication.
- The data memory access pipeline brings needed data into the processor caches from main memory and writes processed data back into memory as appropriate, through the local data cache.

Replays—resolving resource hazards and correcting many soft errors

In some cases, an instruction moving through the pipeline encounters a “resource hazard” that prevents immediate execution. This happens when execution requires results or resources that are not yet available. Resource hazards are detected in the pipeline and can be resolved by a replay.

In a replay, the instruction that encountered the resource hazard is removed from the pipeline, along with all the instructions that come after it. The instruction is then read again out of the instruction buffer and restarted at the beginning of the pipeline. All the instructions that follow it are also reread from the buffer. To ensure a replay can be initiated for any instruction in the pipeline that encounters a resource hazard, a copy of each instruction is maintained.

Figure 1 Poulson Replay Pipeline. Intel Itanium processor 9500 series feature multiple replay paths for hazard avoidance and error correction.
in the instruction buffer until the instruction has successfully traversed the pipeline and is no longer needed. If necessary, an instruction can replay multiple times.

Intel Instruction Replay Technology combines the replay process described above with error detection mechanisms to enable fast, automated correction of soft errors in the pipeline—with very low performance overhead (Figure 1).

Soft error detectors are located in several stages of the instruction execution pipeline to check parity, residues, and ECC. If a transient error occurs while the instruction is flowing down the pipeline, a simple replay of the affected instruction will correct the error. The instruction is simply reread correctly from the instruction buffer and restarted through the pipeline as if the error had never occurred. The instruction executes properly, gets the correct result, and finishes normally.

Transient errors can also occur when data is read from cache into the pipeline; or they may be present in cache even before the data is read. Normally, data are accessed from the single-cycle first level data cache. When a parity calculation identifies a soft error during a read from cache, the erroneous cache entry is removed from cache and a replay of the affected instruction is performed. As the instruction comes down the pipeline again, it accesses the second-level data cache instead of the first level data cache for the needed data value. The error is thereby corrected and the instruction completes normally.

These instruction replay mechanisms provide very quick recovery for soft errors. They delay instruction execution by only seven core clock cycles, which is the length of the instruction execution pipeline. This delay is too short to be visible to software.

**Refetch—identifying and fixing most other soft errors**

Some soft errors that are detected in the instruction execution pipeline cannot be cured by a replay. For example, a soft error can occur in the instruction buffer itself. When the affected instruction is read from the buffer into the pipeline during a replay, the erroneous bit will be included in the instruction and will continue to prevent proper execution.

The next-generation Intel Itanium processor solves this challenge by implementing a “refetch” of the affected instruction instead of a replay. A refetch removes all instructions in the instruction execution pipeline, the instruction buffer, and the instruction fetch pipeline (Figure 2). It then re-reads the instruction, and the instructions after it, from cache. This cleans out the soft error, after which program execution continues without incident. A refetch takes only about twice as long as a replay and serves the same function. The hardware detects the problem and corrects it without any impact on the software.

A refetch can also be used for handling most soft errors that occur in the instruction cache. Parity errors in the first-level instruction cache can be cured with a refetch, as can single-bit or double-bit ECC errors in the mid-level instruction cache. The erroneous cached instruction is removed from cache during the refetch and then fetched from the next higher level of cache or from main memory. Since these locations won’t have the transient error, the soft error is fixed without software interruption and with only a short delay of execution.
Resteer—correcting the rare errors that cannot be fixed in hardware

There is another class of soft errors that are not directly correctable in hardware, but are detected and can be corrected by processor firmware. These errors are handled by a “resteer” operation. During a resteer, the OS or application is interrupted and processor firmware is executed to correct the error. The firmware takes appropriate actions to remove the error from hardware and then restarts the OS or application where it left off.

Resteers are used to correct soft errors in translation lookaside buffers (TLBs) and in general purpose and floating point register files. However, these error scenarios are relatively rare. The vast majority of soft errors that occur in a next-generation Intel Itanium processor will be corrected by the hardware itself, without any need for software intervention.

Conclusion

Intel Itanium processor 9500 series builds on the mainframe-class capabilities delivered by previous-generation Intel Itanium processor 9300 series. In addition to major improvements in performance and throughput, it includes new, enhanced, and extended RAS features to enable improved data integrity and higher levels of system resiliency in mission-critical computing environments.

A prime example of this enhanced RAS support is Intel Instruction Replay Technology, which automatically detects and corrects soft errors in the instruction pipeline. With this technology, soft errors can be identified and corrected in as few as seven clock cycles, which is fast enough to be invisible to the software running on the platform. Through this and many other improvements throughout the platform, Intel Itanium processor 9500 series marks another significant step forward in capability and value for mission-critical computing.