# Next Generation Itanium™ Processor Overview

Intel Developer Forum
Fall 2001


Gary Hammond

Principal Architect

Enterprise Platform Group

Intel Corporation

Sam Naffziger

Lead Circuit Architect

Microprocessor Technology Lab

HP Corporation


August 27-30, 2001

# Agenda

- Program Goals
- Processor Enhancements
- Processor Structures
  - Pipelines
  - Front-End
  - Functional Units
  - Caches
- System Bus
- Reliability Features
- Summary

# McKinley Program Goals

- Builds on Itanium™ processor's EPIC design philosophy
- 100% Itanium Processor Family binary compatibility
- World class performance
  - high performance for commercial servers
  - high integer and floating-point performance
- Support for mission critical applications
  - Robust error recovery and containment

# Architecture Philosophy



- **Builds on Itanium™ processor's EPIC features to achieve higher instruction level parallelism**
  - **Prefetch/branch/cache hints, speculation, predication, register stacking = same as Itanium processor.**
  - **Provide performance scaling + binary compatibility with Itanium-based applications/OSes**
    - **Full hardware score-board design to resolve register hazards between instruction groups**

**Same programming model as Itanium™ processor**
**No code changes required**

# McKinley Optimizations



- Improved dynamic properties
  - Target production frequency is 1 GHz
  - Reduced L1, L2, L3 latencies
  - L3 cache has been incorporated on die
  - Improved L2 cache capacity
  - Improved FSB bandwidth
  - Lower branch prediction penalties

**McKinley provides significant speed ups on existing Itanium™ processor binaries**

# McKinley Optimizations

- **Reduced execution paths**
  - **More parallelism/resources**
    - More integer, multi-media units and memory ports
  - **Short latencies**
    - Fully bypassed functional units
    - Very Low L1D/L2/L3 Cache Latencies
    - Low latency FP execution
  - **Many more ways to issue/execute 6 insts/clk**

**McKinley provides performance headroom for re-optimized binaries**

Processor Enhancements

# Micro-Architectural Enhancements

Intel Developer Forum Fall 2001

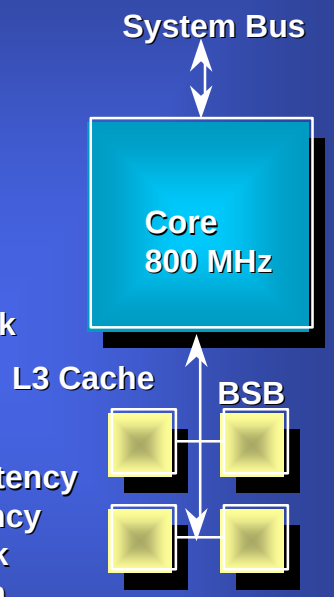
## Itanium™ Processor

**System Bus**
64 bits wide
133MHz/266 MT/s
2.1 GB/s

**Width**
2 bundles per clock
4 integer units
2 load or stores per clock
9 issue ports

**Caches**
L1 – 2X16KB - 2 clock latency
L2 – 96K – 12 clock latency
L3 - 4MB external –20 clk
11.7 GB/s bandwidth

**Addressing**
44 bit physical addressing
50 bit virtual addressing
Maximum page size of 256MB



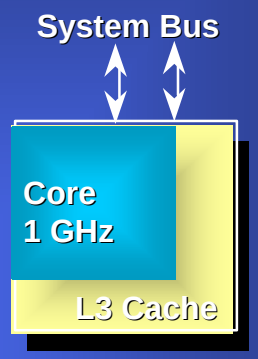

## McKinley

**System Bus**
128 bits wide
200MHz/400 MT/s
6.4 GB/s

**Width**
2 bundles per clock
6 integer units
2 loads and 2 stores per clock
11 issue ports

**Caches**
L1 – 2X16KB - 1 clock latency
L2 – 256K – 5 clock latency
L3 - 3MB – 12 clk
32 GB/s bandwidth

**Addressing**
50 bit physical addressing
64 bit virtual addressing
Maximum page size of 4GB



# Architectural Changes



- **Beneficial to compilers**
  - Improved data/control speculation support
    - ALAT - fully associative = minimize thrashing
    - processor directly vectors to recovery code for reduced speculation costs
  - 64-bit Long Branch Instruction
- **Beneficial to OS and System designs**
  - Full 64-bit virtual addressing
  - Full 2**24 virtual address spaces
  - 4GB virtual pages = reduced TLB pressure
  - 50-bit Physical addressing = very large memory/IO spaces

**Changes provide more flexibility to compiler, OS and system designs**

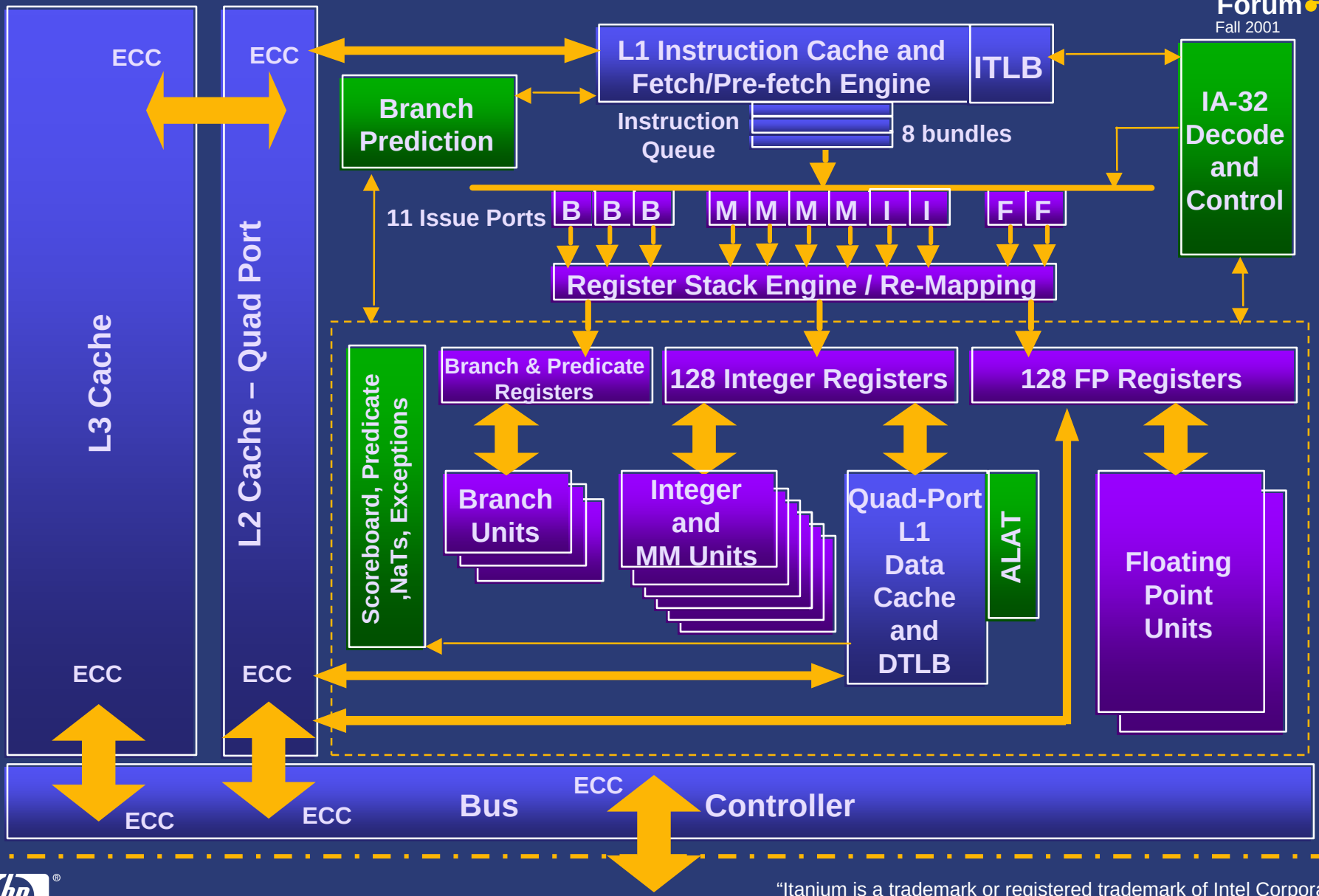
# McKinley Microarchitecture

Intel Developer Forum Fall 2001

- Full Chip Block Diagram
- Pipelines
- Front-end and Branch prediction
- Functional Units
- Caches
- System Bus

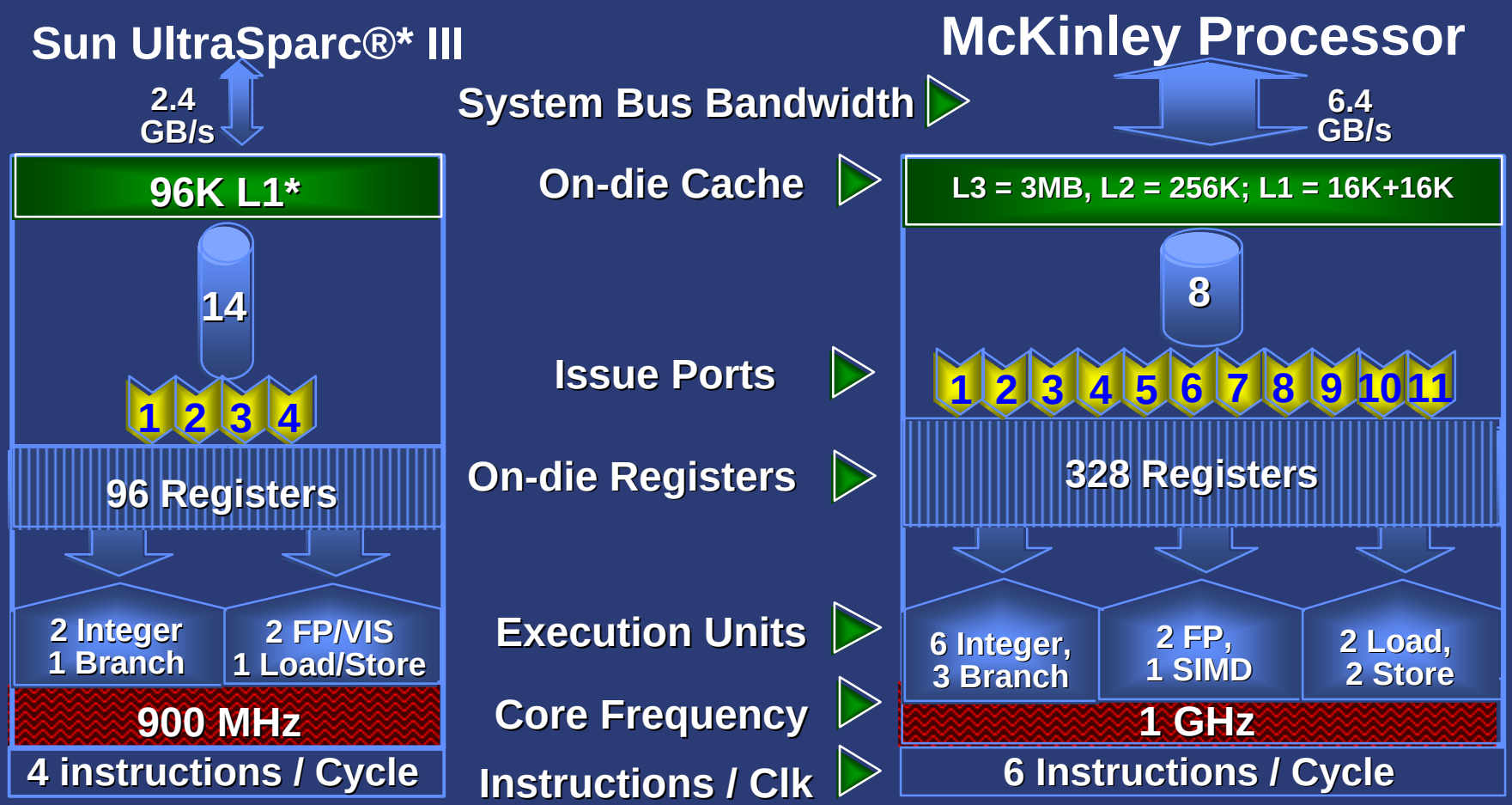
# McKinley Block Diagram





Intel Developer Forum Fall 2001

# Micro-Architecture Comparison




**RISC Architecture** *8 MB L2 External Cache

**EPIC Architecture**

Source: www.sun.com. The SPARC Architecture Manual (Prentice Hall)

# McKinley Pipelines




| | | | | |
|---|---|---|---|---|
| IPG | IP Generate, L1I Cache (6 inst) and TLB access | | EXE | ALU Execute(6), L1D Cache and TLB access + L2 Cache Tag Access(4) |
| ROT | Instruction Rotate and Buffer (6 inst) | | DET | Exception Detect, Branch Correction |
| EXP | Expand, Port Assignment and Routing | | WB | Writeback, Integer Register update |
| REN | Integer and FP Register Rename (6 inst) | | FP1-WB | FP FMAC pipeline (2) + reg write |
| REG | Integer and FP Register File read (6) | | L2N-L2I | L2 Queue Nominate/Issue (4) |
| | | | L2A-W | L2 Access, Rotate, Correct, Write (4) |

- **Short 8-stage in-order main pipeline**
  - In-order issue, out-of-order completion
  - Reduced branch misprediction penalties
  - Fully interlocked, no way-prediction or flush/replay mechanism

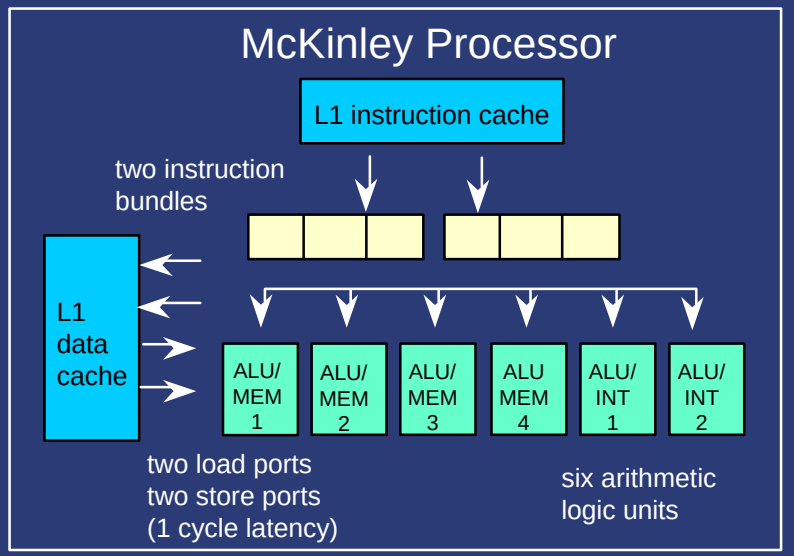
**Pipelines are designed for very low latency**

# McKinley Issue Ports



## Issue ports

- 4 Mem/ALU/Multi-Media
- 2 Integer/ALU/Multi-Media
- 2 FMAC
- 3 branch




## –4 memory ports

- Integer: allow 2 load AND 2 store per clk
- FP: 2 FP load pairs AND 2 store per clk to feed 2 FMACs

**Substantial performance headroom for FP and integer kernels**

# McKinley Dispersal Matrix



| | MII | MLI | MMI | MFI | MMF | MIB | MBB | BBB | MBB | MFM |
|---|---|---|---|---|---|---|---|---|---|---|
| MII | | | | | | | | | | |
| MLI | | | | | | | | | | |
| MMI | | | | | | | | | | |
| MFI | | | | | | | | | | |
| MMF | | | | | | | | | | |
| MIB* | | | | | | | | | | |
| MBB | | | | | | | | | | |
| BBB | | | | | | | | | | |
| MMB* | | | | | | | | | | |
| MFB* | | | | | | | | | | |

* hint in first bundle

Possible McKinley full issue

Possible Itanium™ processor and McKinley full issue

**McKinley allows more compiler dispersal options**

# McKinley Unit Latencies



| | Consuming Class Instruction | | | |
|---|---|---|---|---|
| **Producing Class Instruction** | **Integer** | **Multi-media** | **Load Address** | **Store Data** |
| Mem/integer ports ALU | 1 | 2 | 1 | 1 |
| Integer only ports ALU | 1 | 2 | 1 | 1 |
| Multimedia | 3 | 2 | 3 | 3 |
| Integer Loads (L1D hit) | 1 | 2 | 2 | 1 |

**Short latencies and full bypasses, improve performance for re-optimized code**

# Floating Point Latencies



| Operation | Latency |
|---|---|
| FP Load (4)<br>(L2 Cache hit) | 6 |
| FMAC,FMISC (2) | 4 |
| FP -> Int (getf) | 5 |
| Int -> FP (setf) | 6 |
| Fcmp to branch<br>Fcmp to qual pred | 2<br>2 |

**Short latencies = performance upside for re-optimized FP code**

# Branch Prediction

- Zero clock branch prediction
  - 2 level branch prediction hierarchy
    - L1IBR – Level 1 Branch Cache
      - Part of the L1 I-cache
      - 1K trigger predictions+0.5K target addresses
    - L2B - Level 2 Branch Cache (12K histories)
    - PHT - Pattern History Table (16K counters)
- Reduced prediction penalties
  - IP-relative branch w/correct prediction - 0 cycle
  - IP-relative branch w/wrong target - 1 cycle
  - Return branch w/correct prediction - 1 cycle
  - Last branch in counted loop prediction - 0 cycle
  - Branch Misprediction 6 cycle

**Reduced branch penalties speed up existing code**

# Instruction Prefetching



- **Streaming prefetching**
  - **Initiated by br.many (hint on branch inst)**
    - CPU prefetches ahead the sequential execution stream
    - Streaming prefetch is cancelled by:
      - a predicted-taken branch in the front-end
      - a branch misprediction occurs on the back-end
      - Software cancels the prefetch with a brp instruction
- **Branch Prefetching Hints**
  - **Initiated by brp.few, brp.many or mov_to_br**
    - One time prefetch for the target
    - Two hint prefetches can be initiated per cycle

**Software initiated instruction prefetching improves performance by lower instruction fetch penalties**

# McKinley Caches



| | L1I | L1D | L2 | L3 |
|---|---|---|---|---|
| Size | 16K | 16K | 256K | 3M on die |
| Line Size | 64B | 64B | 128B | 128B |
| Ways | 4 | 4 | 8 | 12 |
| Replacement | LRU | NRU | NRU | NRU |
| Latency (load to use) | I-Fetch:1 | INT:1<br>FP: NA | INT: 5<br>FP: 6 | 12 |
| Write Policy | - | WT (RA) | WB (WA +RA) | WB (WA) |
| Bandwidth | R: 32 GBs | R: 16 GBs<br>W: 16 GBs | R: 32 GBs<br>W: 32 GBs | R: 32 GBs<br>W: 32 GBs |

**All caches are physically indexed, pipelined, and non-blocking: score boarded registers allow continued execution until load use**

# L1D (1 clock Integer Cache)



- High Performance 16GBs, 2 ld AND 2 st ports
  - Write Through – all stores are pushed to the L2
  - FP loads force miss, FP stores invalidate
  - True dual-ported read access – no load conflicts
  - pseudo-dual store port write access
    - 2 store coalescing buffers/port hold data until L1D update
  - Store to load forwarding

**One clock data cache provides a significant performance benefit**

# L2 and L3 Cache



- **L2 256KB, 32GBs, 5 clk**
  - Data array is pseudo-4 ported - 16 banks of 16KB each
- **Non-blocking/out-of-order**
  - L2 queue (32 entries) - holds all in-flight load/stores
  - out-of-order service - smoothes over load/store/bank conflicts, fills
  - Can issue/retire 4 stores/loads per clock
  - Can bypass L2 queue (5,7,9 clk bypass) if
    - no address or bank conflicts in same issue group
    - no prior ops in L2 queue want access to L2 data arrays
- **Large L3 3MB, 32GBs, 12 clk cache on die!!**
  - Single ported – full cache line transfers

**Large on die L2 and L3 cache provides significant performance potential**

# TLBs



- 2-level TLB hierarchy
  - DTC/ITC (32/32 entry, fully associative, .5 clk)
    - Small fast translation caches tied to L1D/L1I
      - Key to achieving very fast 1clk L1D, L1I cache accesses
  - DTLB/ITLB (128/128 entry, fully associative, 1 clk)
    - All architected page sizes (4K to 4GB)
    - Supports up to 64/64 ITR/DTRs
    - TLB miss starts hardware page walker

**Small fast TLBs enable low latency caches**

# System Bus Enhancements

- Extension of the Itanium™ processor bus
  - Same protocol with minor extensions
  - Increased to 6.4GBs bandwidth
    - frequency 200MHz, 400MHz data, 128-bit data bus
  - Bus is non-blocking and out of order
    - Most transactions can be deferred for later service
    - Buffering
      - 18 bus requests/CPU are allowed to be outstanding
      - 16 Read Line + 6 Write Line + two 128 byte WC buffers

**McKinley significantly extends the system bus performance level**

# New Bus Transactions



- **L3 cast-outs (Normally silent L3 replacement (E->I, S->I))**
  - Reduces snoop traffic in Directory based systems
  - Backward inquiry for L2, L1 coherency
- **Memory read current**
  - non-destructive (non-coherent) snoop of CPU lines
  - Used in high bandwidth graphic based systems
- **Cache Cleanse – writes all modified lines to memory**
  - M->E, Used in fault tolerant systems – invoked via PAL

**McKinley provides several new bus transactions to improve performance/reliability**

# Error Features

- **Error detection on all major arrays**
  - **Parity coverage on L1D, L1I, and TLBs**
  - **ECC on L2 and L3**
    - double bit detection single bit correction - Out of path repair
    - all errors are fully contained
  - **Bus is covered with parity/ECC**
    - double bit detection single bit correction on transmission
  - **Error Isolation (end-to-end error detection)**
    - From memory: unique FSB 2xECC syndrome encoding can tolerate additional single bit errors in transmission
    - Error not reported until referenced by a consuming process

**McKinley provides extensive error detection/correction/containment**

# Thermal Management



- **Programmable fail-safe thermal trip**
  - **McKinley will reduce power consumption**
    - Reduce power consumption to ~60% of peak
    - Execution rate dropped to 1 inst per clock
    - Correct Machine Check notification posted to OS
    - Full speed execution resumes when temperature drops
  - **never invoked in properly designed and operating cooling systems**
    - even on worse case power code

**McKinley provides a thermal fail-safe mechanism in the event of a cooling failure**

# Summary

Intel Developer Forum Fall 2001

- McKinley builds on and extends the Itanium™ processor family to meet the needs of the most demanding enterprise and technical computing environments
  - Enhanced McKinley features are a result of extensible Itanium™ architecture
  - McKinley is compatible with Itanium™ processor software
- Major enhancements include:
  - Increased frequency
  - Enhanced micro-architecture – more execution units, issue ports
  - Efficient data handling; higher bandwidth and reduced latencies
- Intel estimates McKinley based systems will deliver ~1.5X – 2X performance improvement over today's Itanium™ based systems