21264 vs NetBurst

Two Different Processors - Both Nonexistent

CSE 240C - Rushi Chakrabarti - WI09
Common Boasts

• Out of Order Execution
• Speculative Execution
• High Performance Memory System
• Industry Leading Clock Rates
Bit ‘o History

• It all started with the 21064
• Clock rate was ~100MHz
• 750nm process
• 1.6 million xtors
21064

• Dual Issue
• 7 stage int/10 stage FP
• 22 in-flight instruction
• 8KB each L1 I$ D$
21164

- 500MHz
- 500nm process
- 9.7 million xtors
21164

- 4 Issue (2 int/2 FP)
- 7 stage int/10 stage FP
- Same L1 caches
- Now with more L2! (96KB)
21264

- 600 MHz
- 350nm process (initially)
- 15.2 million xtors
Stage 0

- Instruction Fetch
- 4 instructions per cycle
- I$ 64K 2-way set associative (huge)
- Remember 21164 only had 8K DM
Stage 0

- On fetch it would set Line and Set Prediction bits
- Line prediction was good for loops and dynamically linked libraries
- Set prediction said which “way” in the cache. Gave it DM like performance.
Stage 0

- Both global and local branch prediction
- 7 cycle penalty
- Uses a tournament predictor
- Can speculate up to 20 branches ahead
Branch Predictor

- Local table: 10 bits history for 1024 branches.
- Global table: 4096 entry table with 2 bits (indexed by history of last 12 branches)
Stage 1

- Instruction assignment to int or FP queues
Stage 2

- Register Renaming

- Gets 4 instructions every cycle, renames, and queues via scoreboard.

- It can issue up to 6 instructions per cycle (4 Int, 2 FP)

- Renamed based on write-reference to register (gets rid of WAW and WAR). Results committed in order.
Stage 2

- 64 arch registers (+ 41 Int and 41 FP physical ones)
- 80 instruction in-flight window
- 21164 had only 20, P6 had 40
- Memory can do an additional 32 in flight loads and 32 in flight stores
Stage 3

• Issue Stage. This is where reordering gets done.

• Selected as data becomes ready from respective (int or FP) queues via register scoreboards. Oldest instructions first.

• Int queue can hold 20, FP can hold 15 instructions.

• Queues are collapsing (ie entry becomes available after issue or squash)
Stage 4

- Register Read
Stage 5

- EX stage
- Int RF are cloned
- Adds 1 cycle of latency to copy values over.
- FP has 1 cluster
Stage 5

- New in this version:
  - fully pipelined integer multiply
  - floating point square root
  - leading/trailing zero counter
Stage 6

- MEM stage.
- 2 memops per cycle.
- D$ is also 64K 2 way.
- 2 memops => twice the frequency of processor.
- 3 cycles for integer load. 4 for FP.
- I+D L2. DM 1-16MB. 12 cycles latency.
Bonus round

- Introduced cache prefetching instructions:
- Normal Prefetch: get 64 bytes into L1/L2 data
- Modify intent: load into cache with writable state
- Evict Intent: fetch with the intention of evicting next access
- Write-hint: Write to 64-byte block without reading first (use to zero out mem)
- Evict: Boot from cache.
Bonus round 2

• Has the ability to do write-invalidate cache coherence for shared memory multiprocessing.

• It does MOESI (modified-owned-exclusive-shared-invalid).
Trivia

- Their External bus used DDR, and also had time-multiplexed control lines. They licensed this to AMD, which went into their Athlon processors as the “EV6 bus”. (wiki)
Trivia

• IBM was able to boost it to around 1.33 GHz using a smaller process.

• Samsung announced a 180nm version at 1.5 GHz, but never made it.
Future

- 21364 came out. It was the EV68 core with a few extra doodads.
- 21464 was cancelled. It was going to double the Int and FP units, and add SMT. 250 million xtors.
Intel

• 8086 -- First x86 processor;

• 80186 -- Included a DMA controller, interrupt controller, timers, and chip select logic.

• 286 -- First x86 processor with protected mode

• i386 -- First 32-bit x86 processor

• i486 -- Intel's 2nd gen 32-bit x86 processors, included built in FP unit
Intel

• P5 -- Original Pentium microprocessors
• P6 -- Used in Pentium Pro, Pentium II, Pentium II Xeon, Pentium III, and Pentium III Xeon microprocessor
• [NetBurst] -- Used in Pentium 4, Pentium D, and some Xeon microprocessors.
• Our Focus today
Intel

- Pentium M -- Updated version of P6 designed for mobile computing
- Enhanced Pentium M -- Updated, dual core version. Core Duo, etc. (Yonah)
- Core -- New microarchitecture, based on the P6 architecture, used in Core 2 and Xeon microprocessors (65nm process).
- Penryn -- 45nm shrink of the Core microarchitecture with larger cache, faster FSB and clock speeds, and SSE4.1 instructions.
Intel

- Nehalem -- 45nm process and used in the Core i7 and Core i5 microprocessors.
- Westmere -- 32nm shrink of the Nehalem
- Sandy Bridge -- Expected around 2010, based on a 32nm process.
- Ivy Bridge -- 22nm shrink of the Sandy Bridge microarchitecture, expected around 2011.
- Haswell -- around 2012, 22nm process.
Unconventional stuff:

- **Atom** -- Low-power, in-order x86-64 processor for use in Mobile Internet Devices.

- **Larrabee** -- Multi-core in-order x86-64 processor with wide SIMD vector units and texture sampling hardware for use in graphics.
Pipelining

• Pentium Pro had 14 pipelining stages.
• PIII went down to 10.
• Pentium M was 12-14
• As we will see Netburst started with 20
  • Last iteration had 31 stages.
More History

• P5:
  • 800 nm process.
  • 3.1 million xtors
  • 60 MHz
  • 8K each I$+D$
  • MMX
P6

- PPro:
  - 600nm/350nm
  - 5.5 million xtors
  - 150-200MHz
  - 8K each I$
  - 256K L2
  - No MMX
P6

- Pentium II
- 350 nm
- 7.5 million xtors
- 233 MHz
- 16K each
- 512K L2
- MMX
P6

- Pentium III
- 250nm process
- 9.5 million xtors
- 450 MHz
- 16K each. 512K L2 on die
- MMX + SSE
- Started the OOO/Spec Exec trend w/ Intel
P6

- It did OOO with
  - Reservation Stations
  - Reorder Buffers
  - 3 instructions/cycle
  - Essentially: Instruction Window!
- Register renaming vital. x86 only has 8 regs
P6 pipeline

• 12 stages. Important ones:
  • BTB access and IF (3-4 stages)
  • Decode (2-3 stages)
  • Register Rename
  • Write to RS
  • Read from RS
  • EX
  • Retire (2 cycles)
PM (just for kicks)

• 130 nm process
• 77 million x tors
• 600MHz - 1.6 GHz
• 32K each. 1 MB L2.
NetBurst

• It was all marketing. GHz race started with Pentium III. High numbers sell. So, they made huge sacrifices for the numbers.

• Deepening the pipeline was the key to getting the numbers high. Not a performance driven improvement =(. 
NetBurst

- Internally called P68 (P7 was IA-64)
  - 180 nm process
  - 1.5 GHz
  - 42 million xtors
  - 16K caches each
  - HT added in 2002
NetBurst (near end)

- 90 nm process
- 125 million xtors
- 2.8GHz-3.4 GHz
- 16K cache each. 1MB L2.
- 31 Stages :(
NetBurst Pipeline

- First to include “drive” stages.
- These shuttle signals across chip wires.
- Keep signal propagation times from limiting the clock speed of the chip.
- No useful work, but we lose 1 more on pipeline flush.
- However, no decode stages (in a bit)
Pipeline Overview

- Stages 1-2: Trace Cache next Inst. Pointer
- Stages 3-4: Trace Cache Fetch
- Stage 5: Drive
- Stage 6-8: Allocate resources and Rename
- Stage 9: Queue by memory or arithm uop
- Stage 10-12: Schedule (i.e. reorder here)
Pipeline Overview

- Stages 13-14: Dispatch. 6 uops/cycle
- Stages 15-16: Register File
- Stage 17: EX
- Stage 18: Flags.
- Stage 19: Branch Check. Should we squash?
- Stage 20: Drive
On to the Paper
Clock Rates

• Trade offs they note in 2000:

• Dependent on:
  • complicated circuit design
  • silicon process technology
  • power/thermal constraints
  • clock skew/jitter
Trace Cache

• Specialized L1 I$
• Stores uops instead of x86 instructions
• This takes decode out of the pipeline
• Gets 3 uops/cycle
• 6 uops/trace line.
Front End

- Trace cache has own BP for subset of program in trace at the time.
- 33% better than P6 when used with the global predictor.
- ROM used for complex IA-32 instructions
  - More than 4 uops
  - ex. String Move is 1000s uops
Branch Predictor

- In addition to TBTB:
  - 4K entries on the front end
  - Otherwise static (back-taken, forward-not)
OOO Execution

- NetBurst can have up to:
  - 126 instructions in flight
  - 48 loads in flight
  - 24 stores
- Register Renaming:
  - 128 registers in file (vs 8 architectural)
Execution Units

- **Exec Port 0**
  - ALU (Double Speed)
  - FP Move
- **Exec Port 1**
  - ALU (Double Speed)
  - Integer Operation
  - FP execute
- **Load Port**
  - Memory Load
- **Store Port**
  - Memory Store

**Add/Sub Logic**
- Add/Sub
- Shift/rotate
- FP/SSE-Add
- FP/SSE-Mul
- FP/SSE-Div
- MMX

**Store Data Branches**
- Store
- Data
- Branches
- FP/SSE Store
- FXCH

**Load Port**
- All loads
- LEA
- SW prefetch

**Store Port**
- Store Address
Hannibal

- Jon Stokes writes for Ars Technica
- Some of the Intel overview was from him
- He is awesome, read him if you already don’t