Inside the
Pentium® 4 Processor
Micro-architecture
Next Generation IA-32 Micro-architecture

Doug Carmean
Principal Architect
Intel Architecture Group

August 24, 2000
Agenda

- IA-32 Processor Roadmap
- Design Goals
- Frequency
- Instructions Per Cycle
- Summary
Intel® Pentium® 4 Processor Design Goals

- Deliver world class performance across both existing and emerging applications
- Deliver performance headroom and scalability for the future

Micro-architecture that will Drive Performance Leadership for the Next Several Years
Intel® NetBurst™ Micro-architecture

- 400 MHz System Bus
- Advanced Dynamic Execution
- Rapid Execution Engine
- Execution Trace Cache
- Enhanced Floating Point / Multi-Media
- Advanced Transfer Cache
- Hyper Pipelined Technology
- Streaming SIMD Extensions 2
- Intel PDX
Delivered Performance = Frequency * Instructions Per Cycle
Frequency

- What limits frequency?
  - Process technology
  - Microarchitecture

- On a given process technology
  - Fewer gates per pipeline stage will deliver higher frequency

Frequency is driven by Micro-architecture
Netburst™ Micro-architecture
Pipeline vs P6

Basic P6 Pipeline

1. Fetch
2. Decode
3. Decode
4. Rename
5. ROB Rd
6. Rdy/Sch
7. Exec

Basic Pentium® 4 Processor Pipeline

1. TC Nxt IP
2. TC Fetch
3. Drive
4. Alloc
5. Rename
6. Que
7. Sch
8. Sch
9. Sch
10. Sch
11. Disp
12. Disp
13. Disp
14. Disp

Hyper pipelined Technology enables industry leading performance and clock rate

Intro at 733MHz .18µ
Intro at ≥ 1.4GHz .18µ

Hyper pipelined Technology enables industry leading performance and clock rate
Hyper Pipelined Technology

- **Introduction**
  - 60MHz
  - 166MHz
  - 233MHz

- **Time**
  - P5 Micro-Architecture
  - P6 Micro-Architecture
  - Netburst™ Micro-Architecture

- **Frequency**
  - ≥1.4GHz
  - 1.13GHz
  - 1.4GHz

Copyright © 2000 Intel Corporation.
Hyper pipelined Technology

TC Nxt IP: Trace cache next instruction pointer
Pointer from the BTB, indicating location of next instruction.
Hyper Pipelined Technology

TC Fetch: Trace cache fetch
Read the decoded instructions (uOPs) out of the Execution Trace Cache
Hyper Pipelined Technology

Drive: Wire delay
Drive the uOPs to the allocator
## Hyper Pipelined Technology

<table>
<thead>
<tr>
<th></th>
<th>TC Nxt IP</th>
<th>TC Fetch</th>
<th>Drive</th>
<th>Alloc</th>
<th>Rename</th>
<th>Que</th>
<th>Sch</th>
<th>Sch</th>
<th>Disp</th>
<th>RF</th>
<th>RF</th>
<th>Ex</th>
<th>Flgs</th>
<th>Br Ck</th>
<th>Drive</th>
</tr>
</thead>
</table>

### Alloc: Allocate

Allocate resources required for execution. The resources include Load buffers, Store buffers, etc.
Hyper Pipelined Technology

|   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  | 19  | 20  |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TC Nxt IP | TC Fetch | Drive | Alloc | Rename | Que | Sch | Sch | Disp | Disp | RF | RF | Ex | Flgs | Br Ck | Drive |

**Rename: Register renaming**

Rename the logical registers (EAX) to the physical register space (128 are implemented).

![Diagram of Hyper Pipelined Technology]

---

Copyright © 2000 Intel Corporation.
Hyper Pipelined Technology

Que: Write into the uOP Queue

uOPs are placed into the queues, where they are held until there is room in the schedulers.
Sch: Schedule

Write into the schedulers and compute dependencies. Watch for dependency to resolve.
Hyper Pipelined Technology

Disp: Dispatch
Send the uOPs to the appropriate execution unit.
Hyper Pipelined Technology

RF: Register File

Read the register file. These are the source(s) for the pending operation (ALU or other).
Hyper Pipelined Technology

Ex: Execute

Execute the uOPs on the appropriate execution port.
Flgs: Flags

Compute flags (zero, negative, etc.). These are typically the input to a branch instruction.
Br Ck: Branch Check

The branch operation compares the result of the actual branch direction with the prediction.
Hyper Pipelined Technology

Drive: Wire delay
Drive the result of the branch check to the front end of the machine.
Delivered Performance = Frequency * Instructions Per Cycle
Improving Instructions Per Cycle

- Improve efficiency
  - Do more things in a clock
  - Branch prediction

- Reduce time it takes to do something
  - Reducing latency
Improving Instructions Per Cycle

- Improve efficiency
  - Branch prediction
  - Do more things in a clock

- Reduce time it takes to do something
  - Reducing latency
Branch Prediction

- Accurate branch prediction is key to enabling longer pipelines
- Dramatic improvement over P6 branch predictor:
  - 8x the size (4K)
  - Eliminated 1/3 of the mispredictions
- Proven to be better than all other publicly disclosed predictors
  - (g-share, hybrid, etc)
Execution Trace Cache

- Advanced L1 instruction cache
  - Caches “decoded” IA-32 instructions (uops)
- Removes decoder pipeline latency
- Capacity is ~12K uOps
- Integrates branches into single line
  - Follows predicted path of program execution

Execution Trace Cache feeds fast engine
### Execution Trace Cache

<table>
<thead>
<tr>
<th>Location</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>3 sub</td>
</tr>
<tr>
<td>T2</td>
<td>5 mov, 6 sub</td>
</tr>
<tr>
<td>T3</td>
<td>8 add, 9 sub</td>
</tr>
<tr>
<td>T4</td>
<td>10 mul, 11 cmp, 12 br</td>
</tr>
</tbody>
</table>

---

### Trace Cache Delivery

<table>
<thead>
<tr>
<th>Location</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cmp</td>
</tr>
<tr>
<td>2</td>
<td>br T1</td>
</tr>
<tr>
<td>3</td>
<td>T1: sub</td>
</tr>
<tr>
<td>4</td>
<td>br T2</td>
</tr>
<tr>
<td>5</td>
<td>mov</td>
</tr>
<tr>
<td>6</td>
<td>sub</td>
</tr>
<tr>
<td>7</td>
<td>br T3</td>
</tr>
<tr>
<td>8</td>
<td>T3: add</td>
</tr>
<tr>
<td>9</td>
<td>sub</td>
</tr>
<tr>
<td>10</td>
<td>mul</td>
</tr>
<tr>
<td>11</td>
<td>cmp</td>
</tr>
<tr>
<td>12</td>
<td>br T4</td>
</tr>
</tbody>
</table>
Advanced Dynamic Execution

- Extends basic features found in P6 core
- Very deep speculative execution
  - 126 instructions in flight (3x P6)
  - 48 loads (3x P6) and 24 stores (2x P6)
- Provides larger window of visibility
  - Better use of execution resources

Deep Speculation Improves Parallelism
Improving Instructions Per Cycle

• Improve efficiency
  – Do more things in a clock
  – Branch prediction

• Reduce time it takes to do something
  – Reducing latency
Rapid Execution Engine

- Dramatically lower ALU latency
- P6:
  - 1 clock @ 1GHz
- Intel® NetBurst™ micro-architecture:
  - ½ clock @ >1.4GHz
  - <0.35ns
High Performance
L1 Data Cache

- 8KB, 4-way set associative, 64-byte lines
- Very high bandwidth
  - 1 Ld and 1 St per clock
- New access algorithms
- Very low latency
  - 2 clock read

New algorithm enables faster cache
Data Speculation

- Observation: Almost all memory accesses hit in the cache
- Optimize for the common case
  - Assume that the access will hit the cache
  - Use a low cost mechanism to fix the rare cases that miss
- Benefit:
  - Reduces latency
  - Significantly higher performance
Replay

- Repairs incorrect speculation
  - Re-execute until correct
- Replay is uOP specific
  - Replay the uOP that mis-speculated
  - Replay dependent uOPs
  - Independent uOPs are not replayed
L1 Cache is >2x Faster

- **P6:**
  - 3 clocks @ 1GHz
- **Intel® NetBurst™ micro-architecture:**
  - 2 clocks @ ≥1.4GHz

Lower Latency is Higher Performance
Example with higher IPC and Faster Clock!

**Code Sequence**

- Ld
- Add
- Add
- Ld
- Add
- Add

**P6 @1GHz**

- 10 clocks
- 10ns
- IPC = 0.6

**Intel® NetBurst™ Micro-architecture @1.4GHz**

- 6 clocks
- 4.3ns
- IPC = 1.0
L2 Advanced Transfer Cache

L2 Cache and Control

- BTB
- Decoder
- Trace Cache
- Rename/Alloc
- uop Queues
- Schedulers
- Integer RF
- FP RF
- uCode ROM
- L1 D-Cache and D-TLB

3.2 GB/s System Interface

Store AGU
Load AGU
ALU
ALU
ALU
FP move
FP store
FMul
FAdd
MMX
SSE
L2 ATC Organization

- 256KB, 8-way set associative
  - 128-byte lines
  - Two 64-byte pieces per line
- Holds both data and instructions
- High bandwidth: 45 GB/Sec @ 1.4GHz
  - 2.8x P6 @1GHz
Aggregate Cache Latency

- Function of all caches in a processor
- Overall Effective Latency
  - L1 latency +
  - L1 Miss Rate * L2 latency +
  - L2 Miss Rate * DRAM Latency

Average cache speed is >1.8x better than the Pentium® III Processor

Average on desktop applications, Intel® Pentium® III processor @ 1GHz, Intel® Pentium® 4 processor @ 1.4GHz
# Comparison

<table>
<thead>
<tr>
<th></th>
<th>Pentium® III Processor</th>
<th>Pentium 4 Processor</th>
<th>Relative Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>1 GHz</td>
<td>≥ 1.4 GHz</td>
<td>≥ 1.4</td>
</tr>
<tr>
<td>Adder Speed</td>
<td>1 ns</td>
<td>&lt; .36 ns</td>
<td>≥ 2.8</td>
</tr>
<tr>
<td>Adder Bandwidth</td>
<td>2 billion/sec</td>
<td>≥ 5.6 billion/sec</td>
<td>&gt; 2.8</td>
</tr>
<tr>
<td>L1 Cache Speed</td>
<td>3 ns</td>
<td>&lt; 1.42 ns</td>
<td>≥ 2.1</td>
</tr>
<tr>
<td>L1 Cache Size</td>
<td>16 KB</td>
<td>8 KB</td>
<td>0.5</td>
</tr>
<tr>
<td>L1 Cache Bandwidth</td>
<td>16 GB/sec</td>
<td>≥ 44.8 GB/sec</td>
<td>≥ 2.8</td>
</tr>
<tr>
<td>L2 Cache Bandwidth</td>
<td>16 GB/sec</td>
<td>≥ 44.8 GB/sec</td>
<td>≥ 2.8</td>
</tr>
<tr>
<td>Instructions in flight</td>
<td>40</td>
<td>126</td>
<td>3.15</td>
</tr>
<tr>
<td>Loads in flight</td>
<td>16</td>
<td>48</td>
<td>3</td>
</tr>
<tr>
<td>Stores in flight</td>
<td>12</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>Branch targets</td>
<td>512</td>
<td>4092</td>
<td>8</td>
</tr>
<tr>
<td>Uop Fetch Bandwidth</td>
<td>3 billion/sec</td>
<td>≥ 4.2 billion/sec</td>
<td>≥ 1.4</td>
</tr>
</tbody>
</table>
Intel® Pentium® 4 Processor

Summary

- Revolutionary, new micro-architecture from Intel designed for the evolving Internet
- Design features for balanced, high performance platform scalability and headroom
- The world’s highest performance desktop processor