AMD's Bulldozer Architecture

Chris Ziemba
Jonathan Lunt
Overview

• AMD's Roadmap
• Instruction Set
• Architecture
• Performance
• Later Iterations
  o Piledriver
  o Steamroller
  o Excavator
1. Changed this section, bulldozer is covered in architecture so it makes sense to not reiterate with later slides

Chris Ziemba, ☞ o
AMD's Roadmap

- **October 2011**
  - First iteration, Bulldozer released

- **June 2013**
  - Piledriver, implemented in 2nd gen FX-CPUs

- **2013**
  - Steamroller, implemented in 3rd gen FX-CPUs

- **2014**
  - Excavator, implemented in 4th gen Fusion APUs

- **2015**
  - Revised Excavator adopted in 2015 for FX-CPUs and beyond
Instruction Set: Overview

- Type: CISC
- Instruction Set: x86-64 (AMD64)
  - Includes Old x86 Registers
  - Extends Registers and adds new ones
  - Two Operating Modes: Long Mode & Legacy Mode
- Integer Size: 64 bits
- Virtual Address Space: 64 bits
  - 16 EB of Address Space (17,179,869,184 GB)
- Physical Address Space: 48 bits (Current Versions)
  - Saves space/transistors/etc
  - 256TB of Address Space
# Instruction Set: ISA

## Registers

**General-Purpose Registers (GPRs)**

<table>
<thead>
<tr>
<th>Register</th>
<th>64-Bit Media and Floating-Point Registers</th>
<th>128-Bit Media Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAX</td>
<td>MMX0/FPR0</td>
<td>XMM0</td>
</tr>
<tr>
<td>RBX</td>
<td>MMX1/FPR1</td>
<td>XMM1</td>
</tr>
<tr>
<td>RCX</td>
<td>MMX2/FPR2</td>
<td>XMM2</td>
</tr>
<tr>
<td>RDX</td>
<td>MMX3/FPR3</td>
<td>XMM3</td>
</tr>
<tr>
<td>RBP</td>
<td>MMX4/FPR4</td>
<td>XMM4</td>
</tr>
<tr>
<td>RSI</td>
<td>MMX5/FPR5</td>
<td>XMM5</td>
</tr>
<tr>
<td>RDI</td>
<td>MMX6/FPR6</td>
<td>XMM6</td>
</tr>
<tr>
<td>RSP</td>
<td>MMX7/FPR7</td>
<td>XMM7</td>
</tr>
<tr>
<td>R8</td>
<td>63</td>
<td>127</td>
</tr>
<tr>
<td>R9</td>
<td>63</td>
<td>0</td>
</tr>
<tr>
<td>R10</td>
<td>63</td>
<td>0</td>
</tr>
<tr>
<td>R11</td>
<td>EFLAGS</td>
<td></td>
</tr>
<tr>
<td>R12</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>R13</td>
<td>EIP</td>
<td></td>
</tr>
<tr>
<td>R14</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>R15</td>
<td>63</td>
<td></td>
</tr>
</tbody>
</table>

- **Legacy x86 registers, supported in all modes**
- **Register extensions, supported in 64-bit mode**

Application-programming registers also include the 128-bit media control-and-status register and the x87 tag-word, control-word, and status-word registers.
## Instruction Set: Operating Modes

<table>
<thead>
<tr>
<th>Operating Mode</th>
<th>Operating System Required</th>
<th>Application Recompile Required</th>
<th>Defaults</th>
<th>Register Extensions</th>
<th>Typical GPR Width (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Long Mode</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64-Bit Mode</td>
<td>New 64-bit OS</td>
<td>yes</td>
<td>64</td>
<td>32</td>
<td>yes</td>
</tr>
<tr>
<td>Compatibility Mode</td>
<td></td>
<td>no</td>
<td>32</td>
<td></td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td><strong>Legacy Mode</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protected Mode</td>
<td>Legacy 32-bit OS</td>
<td>no</td>
<td>32</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Virtual-8086 Mode</td>
<td></td>
<td></td>
<td>16</td>
<td>16</td>
<td>no</td>
</tr>
<tr>
<td>Real Mode</td>
<td>Legacy 16-bit OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Intel x86 Extensions
  o AES-NI: Advanced Encryption Standard (AES) Instructions
  o AVX: Advanced Vector Extensions. 256 bit registers for computationally complex floating point operations such as image/video processing, simulation, etc.

• AMD x86 Extensions
  o XOP: AMD specified SSE5 Revision
  o FMA4: Fused multiply-add (MAC) instructions
Architecture: Block Diagram

[Diagram of a computer architecture showing various components and their connections, including L1 and L2 caches, FPU, L3 cache, Hyper Transport, and DDR3 interface.]
Architecture: Overview

• Multiple Modules
  o 4 Issue, Super-scalar, OOO Execution
  o Fetch/Decode
  o Dispatch
  o 2 'Cores' Per Module (Integer Clusters)
  o Floating Point Unit (Shared Between Cores)

• Cache (L1, L2, L3)
• HyperTransport™ Technology
• Integrate DRAM Controller
• Technology/Limitations
• Branch prediction critical in Bulldozer
  o Run-ahead
• Instructions Decoded To:
  o FastPath Singles (1 Macro-Op)
  o FastPath Doubles (2 Macro-Ops)
  o Microcode
• 32 byte fetch
• Decoder serves one core per cycle
• IBB decouples Fetch/Decode
Decoding: Macro Ops

• AMD Macro-op
  - Actual instructions split up into macro-ops
  - May include arithmetic and memory
  - Up to 3 µOps

• Decoders
  - 1 double instruction per cycle
  - 2 single instructions per cycle
  - Stall while micro-ops are generated (for instructions that generate more than just 2 macro-ops)
  - CMP/TEST followed by conditional jump are fused into one macro op

• Prefixes: x86 Instruction modifiers
  - 1-3 Prefixes: 1 Clock decode
  - 3-7 Prefixes: Additional 14-15 Clock decode
  - 8-11 Prefixes: 21-22 Clock decode
## Decoding: Typical Integer Decomposition

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Macro Ops</th>
<th>Clocks</th>
<th>Reciprocal Throughput</th>
<th>Pipes Usable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD, SUB (reg, reg)</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>EX0, EX1</td>
</tr>
<tr>
<td>CMP</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>EX0, EX1</td>
</tr>
<tr>
<td>IMUL (32 bit)</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>EX1</td>
</tr>
<tr>
<td>IMUL (64 bit)</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>EX1</td>
</tr>
<tr>
<td>DIV (32 bit)</td>
<td>16</td>
<td>16-43</td>
<td>16-43</td>
<td>EX0</td>
</tr>
<tr>
<td>DIV (64 bit)</td>
<td>16</td>
<td>16-75</td>
<td>16-75</td>
<td>EX0</td>
</tr>
</tbody>
</table>
Decoding: Typical Floating Point Decomposition

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Macro Ops</th>
<th>Clocks</th>
<th>Reciprocal Throughput</th>
<th>Pipes Usable</th>
</tr>
</thead>
<tbody>
<tr>
<td>FADD, FSUBB</td>
<td>1</td>
<td>5-6</td>
<td>1</td>
<td>P0, P1</td>
</tr>
<tr>
<td>FMUL</td>
<td>1</td>
<td>5-6</td>
<td>1</td>
<td>P0, P1</td>
</tr>
<tr>
<td>FDIV</td>
<td>1</td>
<td>10-42</td>
<td>5-18</td>
<td>P0, P1</td>
</tr>
<tr>
<td>FCOMI</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>P0, P1, P3</td>
</tr>
</tbody>
</table>
Decoding: Typical Integer MMX/XMM Decomposition

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Macro Ops</th>
<th>Clocks</th>
<th>Reciprocal Throughput</th>
<th>Pipes Usable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMULLD</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>P0</td>
</tr>
<tr>
<td>PAND, PANDN, POR, PXOR</td>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>P2, P3</td>
</tr>
<tr>
<td>DPPS</td>
<td>16</td>
<td>25</td>
<td>6</td>
<td>P0, P1, P2, P3</td>
</tr>
</tbody>
</table>

- PMULLD: Multiplies four sets of 32 bit integers
- PAND, PANDN, POR, PXOR: 128 bit logical operations
- DPPS: Dot product for array of structs data
Architecture: Dispatch

- **Renaming**
  - Stored in PRF

- **Scheduling**
  - Macro-ops direct out of order scheduling
  - Micro-ops executed in pipelines
  - Floating Point cluster will rely on Integer core for Load/Store
Architecture: Integer Cluster

- 2 Integer Clusters per Module

- 2 ALU: Arithmetic Logic Unit
  - EX0/ALU0: DIV, POPCNT, LZCOUNT, Division is not pipelined
  - EX1/ALU1: MUL, JMP, Handles fused jumps, Multiplies are pipelined

- 2 AGU: Address Generation Unit
  - AGU0/AGU1 Perform exact same operations
  - Can perform MOV instructions if there are few Load/Stores
  - Feeds Load/Store Unit

- LSU: Load/Store Unit

- DTLB: Data Transition Lookaside Buffer
  - L1: 32 Entry fully associative
  - L2: 1024 Entry 8-way Associative
• One floating point unit per module

• Four Pipelines:
  o Po: MAC Unit - Add, Mul, Div, Convert, Shuffle, Shift, XOP instructions
  o P1: MAC Unit - Add, Mul, Div, Shuffle, Shift
  o P2: 128 Bit Integer SIMD ALU
  o P3: 128 Bit Integer SIMD ALU, Passes data to store unit

• 256 Bit AVX Instructions
  o Split into two macro-ops
  o Since store unit is not doubled and register-to-register moves will cause a stall penalty, there is no advantage in using AVX instructions over 128 bit instructions
Architecture: Load/Store

• 2 Per Module
  o 1 Per Integer Cluster
  o FPU's Must Use Integer Load/Store Unit

• Queued
  o 40 Entry Load Queue
  o 24 Entry Store Queue

• Size
  o 256 bits per store
  o 128 bits per load

• Memory Disambiguation
  o Detects data dependencies
Architecture: Cache Level Diagram

Machine (32GB)

Socket P90 (16GB)

NUMANode P90 (8192MB)

L3 (8192KB)

L2 (2048KB)

L1i (64KB)

L1d (16KB)

Core P90

PU P49

PU P50

PU P51

NUMANode P91 (8192MB)

L3 (8192KB)

L2 (2048KB)

L1i (64KB)

L1d (16KB)

Core P91

PU P49

PU P50

PU P51

Socket P101 (16GB)

NUMANode P102 (8192MB)

L3 (8192KB)

L2 (2048KB)

L1i (64KB)

L1d (16KB)

Core P102

PU P416

PU P417

PU P418

NUMANode P103 (8192MB)

L3 (8192KB)

L2 (2048KB)

L1i (64KB)

L1d (16KB)

Core P103

PU P416

PU P417

PU P418

Indexes: physical
Date: Tue 09 Oct 2012 12:18:38 AM EDT
Architecture: Cache

• L1 Cache: Split
  o Data Cache: 2 Per Module (One per integer cluster)
    ▪ 16KB 4-Way Associative
    ▪ Write Through
    ▪ Predictive (misprediction adds stalls)
    ▪ 4 Clocks best case hit time
  o Instruction Cache: 1 Per Module
    ▪ 64KB 2-Way Associative

• L2 Cache: 8MB (Split)
  o 2MB Per Module
  o Includes L1 Data Cache
  o 16-way associative
  o 18-20 Cycle hit time

• Write Coalescing Cache
Architecture: Execution Flow Overview

Instruction Fetched
Instruction → Decode → Macro-ops
Macro-ops → Float or Integer Pipeline
Pipeline → Allocate Retirement Queue Entry
  → Physical Register File (renaming)
  → Scheduler
Scheduler → Micro-ops
  → Execute (when inputs for Micro-ops are ready)
Completed Macro-op → Signal Core for Retirement
Architecture: HyperTransport™ Technology

• AMD's Choice of BUS Technology
  o Bidirectional, Serial/Parallel, Point-to-point link
  o Low latency, high bandwidth
  o Double Data Rate: Data sent on rise and fall

• HyperTransport 3.1
  o 3.2 GHz
  o 1 16-bit link (16 receive/ 16 transmit)
  o 12.8 GB/s (Tx/Rx)
  o Maximum (not enabled)
    ▪ 4 16 bit links
    ▪ 51.2 GB/s
Architecture: DRAM Controller

- **2.2 GHz Clock**
- **Supports DDR3-1866**
  - 2 72 bit wide memory channels
- **Power Saving:**
  - Throttle activity for power reduction
  - Place in standby when no reads active
**Architecture: Power Management**

- **DVFS: Dynamic Voltage and Frequency Scaling**
  - Unless under heavy load CPU will be underclocked
  - Reducing voltage reduces transistor switching time which means frequency also needs to be reduced
  - Power Consumed in CMOS can be described by
    - $P = \alpha C_{eff} V^2 f$  ($\alpha =$ switching factor)

- **Power Gating**
  - 4 Modules: Some will turn off if necessary
  - Entire module must be turned off
    - Added complexity in turning off only parts of modules
    - Negligible gains when turning off parts
Performance: Bottlenecks

• **DVFS:**
  - Gives inconsistent performance
  - May require long stream of data to get full performance

• **Fetch/Decode**
  - Shared between two cores
  - Alternates between 2 threads so maximum of 2 instruction issues per thread per cycle (Major Bottleneck)
  - Instructions generating more than two macro-ops are handled with microcode, this blocks the decoders for several cycles

• **Integer Execution Units**
  - Single thread pure integer code can only execute 2 instructions per cycle max

• **Cache**
  - Cache bank conflicts are frequent
  - 2-Way Associative L1 Instruction serving 2 threads is too little
Performance: Power

- Power Hungry

### Power Consumption - Idle
Total System Power Consumption in Watts (Lower is Better)

<table>
<thead>
<tr>
<th>Processor Type</th>
<th>Idle Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Core i5 2500K (3.3GHz)</td>
<td>76</td>
</tr>
<tr>
<td>Intel Core i5 2400 (3.1GHz)</td>
<td>76.2</td>
</tr>
<tr>
<td>Intel Core i7 2600K (3.4GHz)</td>
<td>77.6</td>
</tr>
<tr>
<td>AMD FX-8150 (3.6GHz)</td>
<td>84.8</td>
</tr>
<tr>
<td>AMD Phenom II X6 1100T (3.3GHz)</td>
<td>109.4</td>
</tr>
<tr>
<td>AMD Phenom II X4 975 BE (3.6GHz)</td>
<td>110</td>
</tr>
</tbody>
</table>

### Power Consumption - Load (x264 HD 3.03 2nd Pass)
Total System Power Consumption in Watts (Lower is Better)

<table>
<thead>
<tr>
<th>Processor Type</th>
<th>Load Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Core i5 2400 (3.1GHz)</td>
<td>131.6</td>
</tr>
<tr>
<td>Intel Core i5 2500K (3.3GHz)</td>
<td>133.3</td>
</tr>
<tr>
<td>Intel Core i7 2600K (3.4GHz)</td>
<td>155.4</td>
</tr>
<tr>
<td>AMD Phenom II X4 975 BE (3.6GHz)</td>
<td>183.8</td>
</tr>
<tr>
<td>AMD Phenom II X6 1100T (3.3GHz)</td>
<td>200</td>
</tr>
<tr>
<td>AMD FX-8150 (3.6GHz)</td>
<td>229</td>
</tr>
</tbody>
</table>
Performance: Synthetic Benchmark

- Poor Single Threaded Performance
Performance: Cache Latency

- Poor Cache Timing

![L3 Cache Latency Bar Chart]

- Intel Core i5 2500K (3.3GHz): 7.14 ns
- Intel Core i5 2500K (3.3GHz, Turbo Off): 8.18 ns
- AMD Phenom II X4 975 BE (3.6GHz): 16.4 ns
- AMD Phenom II X6 1100T (3.3GHz): 16.7 ns
- AMD FX-8150 (3.6GHz): 16.7 ns
- AMD FX-8150 (3.6GHz, Turbo Off): 19.2 ns

![Memory Latency Bar Chart]

- Intel Core i5 2500K (3.3GHz, Turbo Off): 46.1 ns
- AMD Phenom II X6 1100T (3.3GHz): 47.6 ns
- AMD FX-8150 (3.6GHz): 48.5 ns
- AMD Phenom II X4 975 BE (3.6GHz): 50.6 ns
- AMD FX-8150 (3.6GHz, Turbo Off): 54.2 ns
Improvements/Features: Piledriver (2nd Generation)

- Importance of Branch Prediction
- FMA3 - Fused Multiply Add
- F16C - Half Precision to Single Precision
Improvements/Features: Steamroller (3rd Generation)

- Front End Improvements
  - Duplicate Decode
  - Address the bottleneck
Improvements/Features: Steamroller (3rd Generation)

- 30% Increased ops per cycle
  - Larger I-Cache
Improvements/Features: Steamroller (3rd Generation)

- High Density Library
  - 30% area and power reduction
Improvements/Features: Excavator (4th Generation)

• Remains to be seen