Fujitsu High Performance CPU for the Post-K Computer

August 21st, 2018
Toshio Yoshida
FUJITSU LIMITED
A64FX is the new Fujitsu-designed Arm processor
  - It is used in the post-K computer

A64FX is the first processor of the Armv8-A SVE architecture
  - Fujitsu, as a lead partner, collaborated closely with Arm on the development of SVE

A64FX achieves high performance in HPC and AI areas
  - Our own microarchitecture maximizes the capability of SVE
Outline

- Fujitsu Processor Development
- A64FX
  - Overview
  - Microarchitecture
  - Performance
  - Power Management
  - RAS
- Software Development
- Summary
Fujitsu Processor Development

Persistent Evolution > 60 years

Scalable Many-core Architecture
512-bit SIMD for HPC and AI
High Bandwidth Memory
Virtual Machine Architecture
Software on Chip
High-speed Interconnect
HPC-ACE
System on Chip
Hardware Barrier
Multi-core Multi-thread
L2$ on Die
Non-Blocking $ O-O-O Execution
Super-Scalar
Single-chip CPU
Store Ahead
Branch History
Prefetch

$ ECC
Register/ALU Parity
Instruction Retry
$ Dynamic Degradation
Error Checkers/History

* Post-K is underdevelopment by RIKEN and Fujitsu
DNA of Fujitsu Processors

- A64FX inherits DNA from Fujitsu technologies used in the mainframes, UNIX and HPC servers

**High reliability**
- Stability
- Integrity
- Continuity

**High speed & flexibility**
- Thread performance
- Software on Chip
- Large SMP

**High performance-per-watt**
- Execution and memory throughput
- Low power
- Massively parallel

**CPU w/ extremely high throughput**
- High performance
- Massively parallel
- Low power
- Stability and integrity

© RIKEN
A64FX Designed for HPC/AI

A64FX = CPU with extremely high throughput

1. High Performance

- HPC/AI apps. >> General purpose CPU
- Various data types
  - (FP64/32/16, INT64/32/16/8)

2. High Throughput

- Vector: 512-bit wide SIMD x 2 pipes /core
- Memory: HBM2 (extremely high B/W)
- Scalable: 48 cores, Tofu interconnect

3. High Efficiency

- Performance
  - (D|S|H)GEMM: >90%
  - Stream Triad: >80%
  - Perf-per-watt: >> General purpose CPU

4. Standard

- Binary compatibility with
  - Armv8.2-A + SVE + SBSA* level3

*Arm’s “Server Base System Architecture”
A64FX Chip Overview

**Architecture Features**
- Armv8.2-A (AArch64 only)
- SVE 512-bit wide SIMD
- 48 computing cores + 4 assistant cores*
  *All the cores are identical
- HBM2 32GiB
- Tofu 6D Mesh/Torus 28Gbps x 2 lanes x 10 ports
- PCIe Gen3 16 lanes

**7nm FinFET**
- 8,786M transistors
- 594 package signal pins

**Peak Performance (Efficiency)**
- >2.7TFLOPS (>90%@DGEMM)
- Memory B/W 1024GB/s (>80%@Stream Triad)
A64FX Features

- Collaboration with Arm to develop and optimize SVE for a wide range of applications
  - FP16 and INT16/8 dot product are introduced for AI applications

<table>
<thead>
<tr>
<th></th>
<th>A64FX (Post-K)</th>
<th>SPARC64 XIfx (PRIMEHPC FX100)</th>
<th>SPAR64 VIIIIfx (K computer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISA</td>
<td>Armv8.2-A + SVE</td>
<td>SPARC-V9 + HPC-ACE2</td>
<td>SPARC-V9 + HPC-ACE</td>
</tr>
<tr>
<td>SIMD Width</td>
<td>512-bit</td>
<td>256-bit</td>
<td>128-bit</td>
</tr>
<tr>
<td>Four-operand FMA</td>
<td>✓ Enhanced</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Gather/Scatter</td>
<td>✓ Enhanced</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predicated Operations</td>
<td>✓ Enhanced</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Math. Acceleration</td>
<td>✓ Further enhanced</td>
<td>✓ Enhanced</td>
<td>✓</td>
</tr>
<tr>
<td>Compress</td>
<td>✓ Enhanced</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First Fault Load</td>
<td>✓ New</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP16</td>
<td>✓ New</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT16/INT8 Dot Product</td>
<td>✓ New</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HW Barrier* / Sector Cache*</td>
<td>✓ Further enhanced</td>
<td>✓ Enhanced</td>
<td>✓</td>
</tr>
</tbody>
</table>

* Utilizing AArch64 implementation-defined system registers
A64FX Core Pipeline

- A64FX enhances and inherits superior features of SPARC64
  - Inherits superscalar, out-of-order, branch prediction, etc.
  - Enhances SIMD and predicate operations
    - 2x 512-bit wide SIMD FMA + Predicate Operation + 4x ALU (shared w/ 2x AGEN)
    - 2x 512-bit wide SIMD load or 512-bit wide SIMD store
Four-operand FMA with Prefix Instruction

- MOVPRFX as a prefix instruction
  - For SVE, four-operand “FMA4” requires a prefix instruction (MOVPRFX) followed by destructive 3-operand FMA3
  
  \[
  \text{MOVPRFX: } Z0 \leftarrow Z3 \\
  \text{FMA3: } Z0 \leftarrow Z0 + Z1 \times Z2 \\
  \text{“FMA4” : } Z0 \leftarrow Z3 + Z1 \times Z2
  \]

  Equivalent

- A64FX implementation for MOVPRFX
  - A64FX hides the overhead of its main pipeline by packing MOVPRFX and the following instruction into a single operation
Execution Unit

- Extremely high throughput
  - 512-bit wide SIMD x 2 Pipelines x 48 Cores
  - >90% execution efficiency in (D|S|H)GEMM and INT16/8 dot product

![Peak Performance Chart]

峰值性能（芯片级）

- SPARC64 VIIIfx (K computer) 128-bit SIMD
- SPARC64 XIfx (PRIMEHPC FX100) 256-bit SIMD
- A64FX (Post-K) 512-bit SIMD

*1 INT8 dot product

\[ C = \sum (A_i \times B_i) + C \]
Level 1 Cache

- L1 cache throughput maximizes core performance
  - Sustained throughput for 512-bit wide SIMD load
    - An unaligned SIMD load crossing cache line keeps the same throughput

- “Combined Gather” mechanism increasing gather throughput
  - Gather processing is important for real HPC applications
  - A64FX introduces “Combined Gather” mechanism enabling to return up to two consecutive elements in a “128-byte aligned block” simultaneously

< Combined Gather >

Throughput/Core [byte/cycle]

Combined Gather

Gather (Normal)

2x faster
Many-Core Architecture

- A64FX consists of four CMGs (Core Memory Group)
  - A CMG consists of 13 cores, an L2 cache and a memory controller
    - One out of 13 cores is an assistant core which handles daemon, I/O, etc.
  - Four CMGs keep cache coherency by ccNUMA with on-chip directory
  - X-bar connection in a CMG maximizes high efficiency for throughput of the L2 cache
  - Process binding in a CMG allows linear scalability up to 48 cores

- On-chip-network with a wide ring bus secures I/O performance
High Bandwidth

- Extremely high bandwidth in caches and memory
  - A64FX has out-of-order mechanisms in cores, caches and memory controllers. It maximizes the capability of each layer’s bandwidth

- Performance >2.7TFLOPS
- L1 Cache >11.0TB/s (BF ratio = 4)
- L2 Cache >3.6TB/s (BF ratio = 1.3)
- Memory 1024GB/s (BF ratio =~0.37)

- CMG
  - 12x Computing Cores + 1x Assistant Core
  - 512-bit wide SIMD 2x FMAs
  - L1D 64KiB, 4way
  - L1 Cache 8MiB, 16way
  - L2 Cache 8GiB
  - memory 1024GB/s (BF ratio =~0.37)

All Rights Reserved. Copyright © FUJITSU LIMITED 2018
A64FX boosts performance up by microarchitectural enhancements, 512-bit wide SIMD, HBM2 and process technology

- > 2.5x faster in HPC/AI benchmarks than SPARC64 XIfx (Fujitsu’s previous HPC CPU)
- The results are based on the Fujitsu compiler optimized for our microarchitecture and SVE

A64FX Benchmark Kernel Performance (Preliminary results)
Power Management

“Energy monitor” / “Energy analyzer” for activity-based power estimation

- Energy monitor (per chip) : Node power via Power API\(^*$\) (~msec)
  - Average power estimation of a node, CMG (cores, an L2 cache, a memory) etc.
- Energy analyzer (per core) : Power profiler via PAPI\(^**\) (~nsec)
  - Fine grained power analysis of a core, an L2 cache and a memory

→ Enabling chip-level power monitoring and detailed power analysis of applications

\(<A64FX\ Energy\ monitor/\ Energy\ analyzer>\)
“Power knob” for power optimization

A64FX provides power management function called “Power Knob”
- Applications can change hardware configurations for power optimization
  → Power knobs and Energy monitor/analyzer will help users to optimize power consumption of their applications

A64FX Power Knob Diagram
- Decode width: 2
- EX pipeline usage: EXA only
- Frequency reduction
- FL pipeline usage: FLA only
- HBM2 B/W adjustment (units of 10%)
Large systems require extensive RAS capability of CPU and interconnect

A64FX has a mainframe class RAS for integrity and stability. It contributes to very low CPU failure rate and high system stability

- ECC or duplication for all caches
- Parity check for execution units
- Hardware instruction retry
- Hardware lane recovery for Tofu links
- ~128,400 error checkers in total

**<A64FX RAS Mechanism>**

<table>
<thead>
<tr>
<th>Units</th>
<th>Error Detection and Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache (Tag)</td>
<td>ECC, Duplicate &amp; Parity</td>
</tr>
<tr>
<td>Cache (Data)</td>
<td>ECC, Parity</td>
</tr>
<tr>
<td>Register</td>
<td>ECC (INT), Parity(Others)</td>
</tr>
<tr>
<td>Execution Unit</td>
<td>Parity, Residue</td>
</tr>
<tr>
<td>Core</td>
<td>Hardware Instruction Retry</td>
</tr>
<tr>
<td>Tofu</td>
<td>Hardware Lane Recovery</td>
</tr>
</tbody>
</table>

**<A64FX RAS Diagram>**

- Green: 1 bit error Correctable
- Yellow: 1 bit error Detectable
- Gray: 1 bit error harmless
软件开发

- RIKEN 和 Fujitsu 正在开发后-K 系统的软件栈
  - Fujitsu 优化了编译器，针对微架构，最大限度地提高了 SVE 和 HBM2 性能
- 我们与 RIKEN / Linaro / OSS 社区 / ISVs 合作，并为 Arm HPC 生态系统做出贡献

<table>
<thead>
<tr>
<th>Post-K 应用程序</th>
</tr>
</thead>
<tbody>
<tr>
<td>管理软件</td>
</tr>
<tr>
<td>操作系统管理</td>
</tr>
<tr>
<td>系统管理，高可用性及节能</td>
</tr>
<tr>
<td>工作任务管理</td>
</tr>
<tr>
<td>高利用率及节能</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Linux OS / McKernel (Lightweight Kernel)
Summary

- A64FX is the first processor of the Armv8-A SVE architecture. It is used for the post-K computer

- Fujitsu’s proven microarchitecture achieves high performance in HPC and AI areas

- Fujitsu collaboratively works with partners and continuously contributes to Arm ecosystem

- We will continue to develop Arm processors
Abbreviations

- **A64FX**
- **RSA**: Reservation station for address generation
- **RSE**: Reservation station for execution
- **RSBR**: Reservation station for branch
- **PGPR**: Physical general-purpose register
- **PFPR**: Physical floating-point register
- **PPR**: Physical predicate register
- **CSE**: Commit stack entry
- **EAG**: Effective address generator
- **EX**: Integer execution unit
- **FL**: Floating-point execution unit
- **PRX**: Predicate execution unit
- **Tofu**: Torus-Fusion