Chapter 6
PA Event

FUJITSU LIMITED
April 2016

Contents

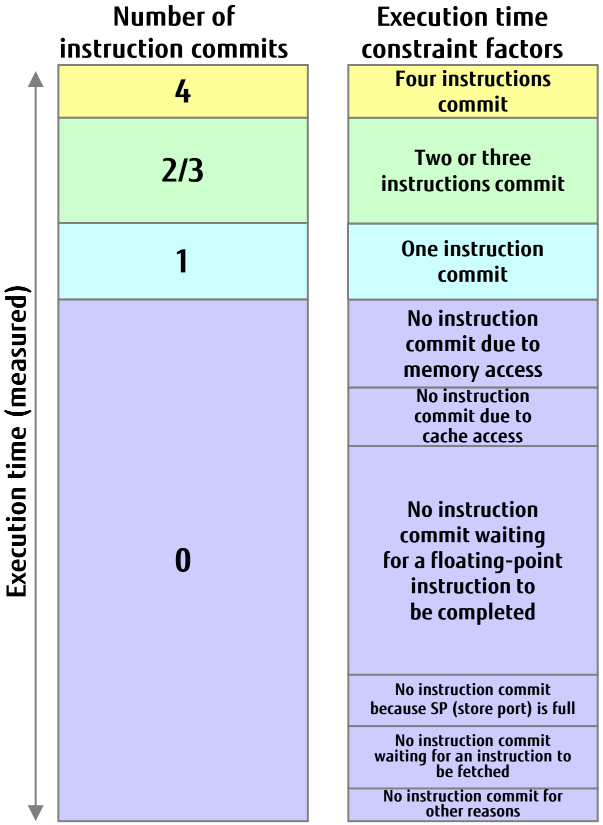
- What Is Cycle Accounting?
- Within a Node: CPU Tuning Techniques (Summary)
- Precision PA Visibility Function (Excel Format)
- PA Graph Components
  - Commit Events
  - Stall Events
- PA Information
  - PA (Performance Analysis) Information Report
  - Performance Information
  - Memory Throughput Information and Cache Throughput Information
  - SIMD Instruction Information
  - Cache Miss Information
  - Instruction count information
  - Load Balance Information
  - XFILL Flag

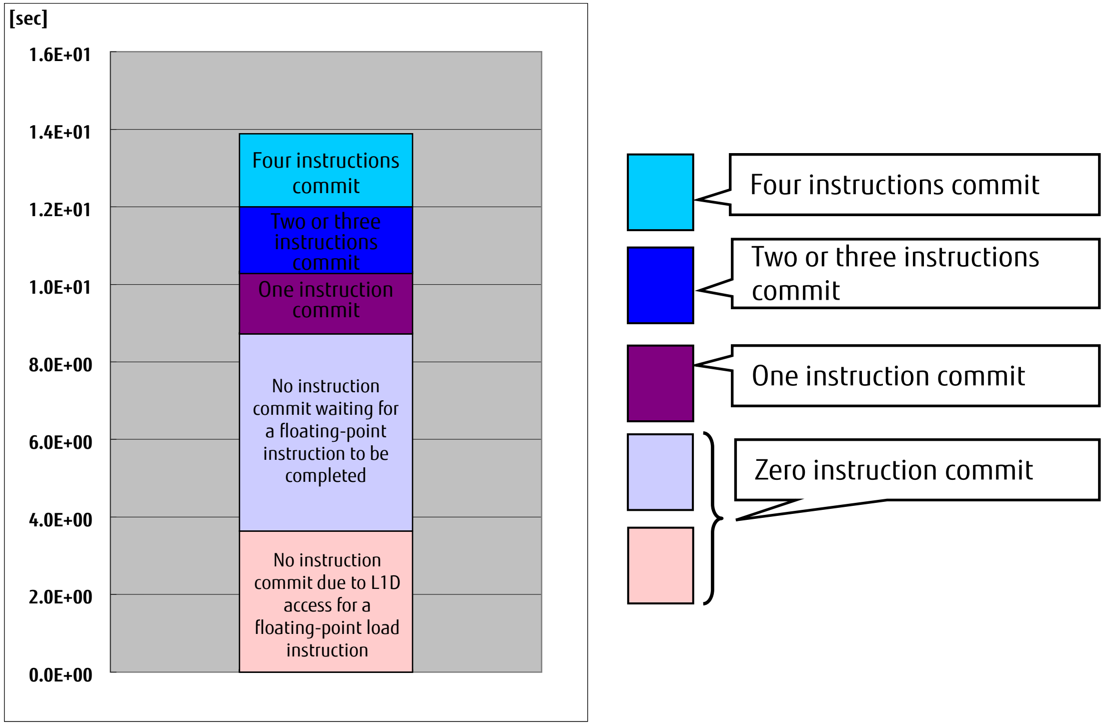
What Is Cycle Accounting?

- **Cycle accounting is an approach to analyzing performance bottleneck factors.**

  The SPARC64™ XIfx enables you to measure a wide range of PA (Performance Analysis) events to check the CPU operating status during application program execution.
  The information appears as a graph of the total time (the number of CPU cycles) taken to execute the application program, and it is broken down by CPU operating status. From this graph, you can get a grasp of bottlenecks in the CPU, and analyze and tune performance in detail.

Outline


Number of instruction commits
Execution time constraint factors
Execution time (measured)
4
Four instructions commit
2/3
Two or three instructions commit
1
One instruction commit
0
No instruction commit due to memory access
No instruction commit due to cache access
No instruction commit waiting for a floating-point instruction to be completed
No instruction commit because SP (store port) is full
No instruction commit waiting for an instruction to be fetched
No instruction commit for other reasons



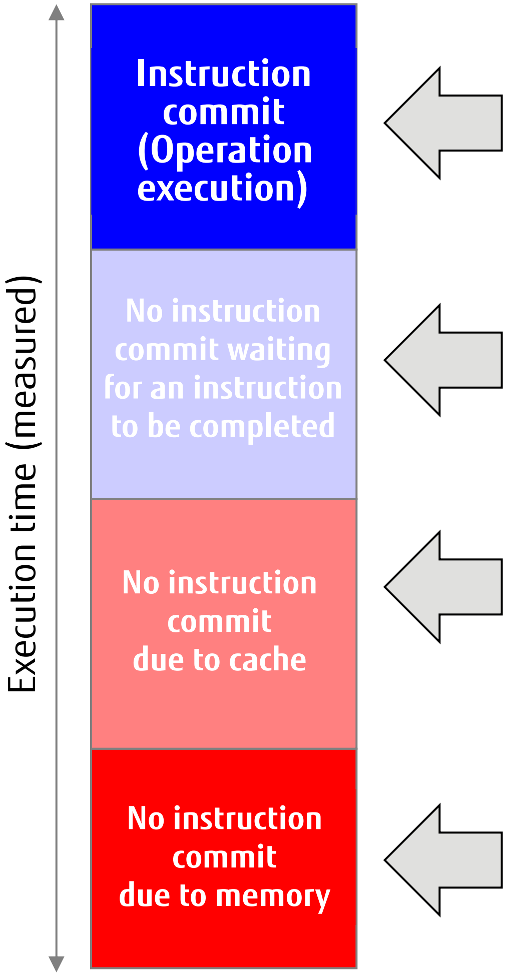
Instruction commit: Period of execution of n instructions in one machine cycle

Zero instruction commit: Period of stalled instructions due to some factor

Intra-Node: CPU Tuning Techniques (Summary)

- Select a tuning technique based on the results of cycle accounting, considering the following:

Breakdown of execution time


Execution time (measured)
Instruction commit (Operation execution)
No instruction commit waiting for an instruction to be completed
No instruction commit due to cache
No instruction commit due to memory

Major tuning techniques

- Promoting optimal reduction of the number of instructions
  - SIMD optimization
  - Elimination of common subexpressions

- Promoting instruction scheduling/software pipelining
  - Mask execution of IF statements and list vector conversion
  - Loop unrolling and loop fission
  - Loop unswitching

- Improving the efficiency of L1 cache use
  - Padding, array merging, loop fission, and loop fusion
- Hiding L2 cache latency
  - L1 prefetch (stride/list access)

- Improving the efficiency of L2 cache use
  - Loop blocking
  - Loop fusion and outer loop unrolling
- Hiding memory latency
  - L2 prefetch (stride/list access)

Precision PA Visibility Function (Excel Format) (1/2)

- The precision PA visibility function (Excel Format) is provided as a tool for obtaining the necessary PA information (including PA graphs) for performance analysis and tuning.

  * For details on how to use it, see "Precision PA Visibility Function (Excel Format) " in "Chapter 7 Tuning Tool."

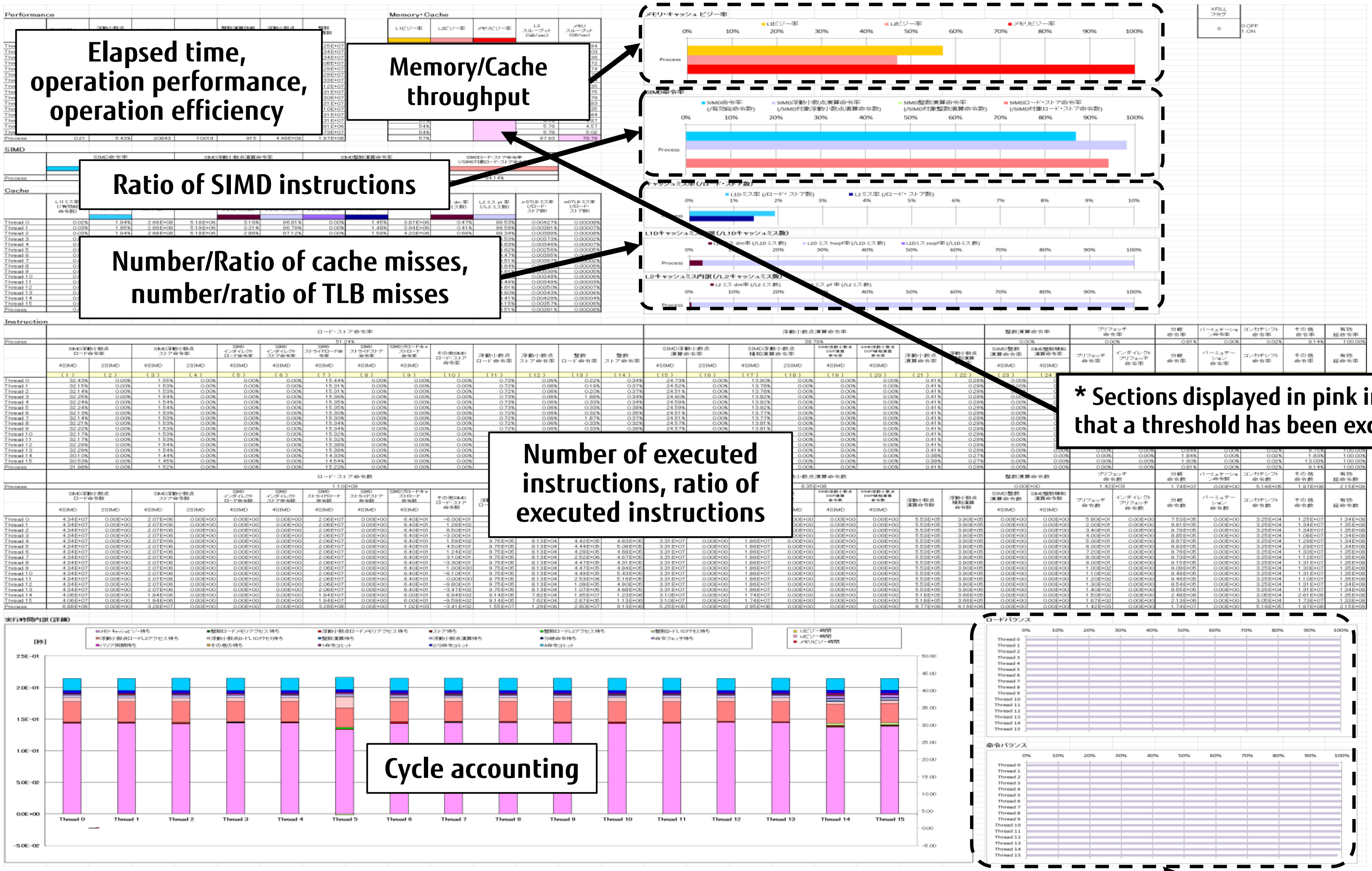
- Outline of the precision PA visibility function (Excel format)

  This tool outputs collected CSV-format PA information in the formats shown on the next page.

Precision PA Visibility Function (Excel Format) (2/2)

■ Output formats (Screenshots)

⬚ Indicator



Elapsed time, operation performance, operation efficiency

Memory/Cache throughput

Ratio of SIMD instructions

Number/Ratio of cache misses, number/ratio of TLB misses

* Sections displayed in pink indicate that a threshold has been exceeded.

Number of executed instructions, ratio of executed instructions

Cycle accounting

Load balance

*** The aggregate results are displayed in an Excel sheet (two A4 pages when printed).**

PA Graph Components

- Commit Events
- Stall Events

Commit Events

Precision PA Visibility Function (Excel Format) outputs two types of information:
PA graphs and PA information lists. This section describes the cycle events
that make up a PA graph.
The first descriptions cover the commit type of events.
Commit events represent the time (number of cycles) during which instructions are executed.

- 🟦 **Four instructions commit**

  This represents the number of cycles during which four instructions have been completed.
  (Four instructions commit: Maximum cycle efficiency)

- 🟦 **Two or three instructions commit**

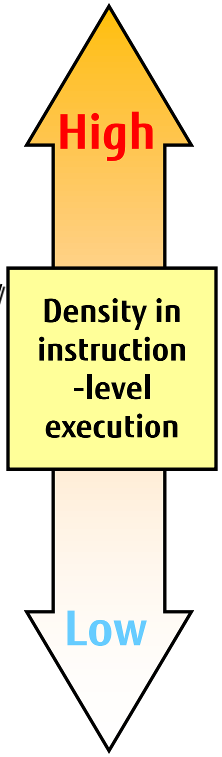
  This represents the number of cycles during which two or three instructions have been completed.

- 🟪 **One instruction commit**

  This represents the number of cycles during which one instruction has been completed.

Density in instruction-level execution tends to be higher when there is no data dependency on the loop iteration direction (return reference), no IF construct in the innermost loop, and a high ratio of operation instructions to the effective instruction.

Density in execution **High: High ratio of commit events**
Low: High ratio of stall events



Stall Events (1/2)

The next descriptions cover the stall type of events.
Stall events represent the time (number of cycles) during which no instruction is executed for whatever reason.
The color of each paragraph corresponds to a color in the PA graph.

- 🟪 **No instruction commit waiting for a floating-point instruction to be completed**
- 🟦 **No instruction commit waiting for an integer instruction to be completed**

  This represents the number of cycles during which zero instructions have been completed, because the oldest instruction being executed is executing an operation with integers or floating-point numbers.

  This state tends to occur when data has a dependency on the loop iteration direction (return reference) or the innermost loop contains an IF construct.

- 🟥 **No instruction commit due to L1D access for a floating-point load instruction**
- 🟥 **No instruction commit due to L2 access for a floating-point load instruction**
- 🟥 **No instruction commit due to memory access for a floating-point load instruction**
- 🟩 **No instruction commit due to L1D access for an integer load instruction**
- 🟩 **No instruction commit due to L2 access for an integer load instruction**
- 🟩 **No instruction commit due to memory access for an integer load instruction**

  This represents the number of cycles during which zero instructions have been completed, because the oldest instruction being executed is accessing memory or cache.

  This state tends to occur in the following situations:

  - Stride access or indirect access is in progress.
  - Even during sequential access, cache thrashing occurs frequently, or performance is limited because cache throughput or memory throughput has reached its limit.
  - Prefetching does not work for some reason, so L2 cache or memory access latency cannot be hidden.

Stall Events (2/2)

- ■ **No instruction commit because SP (store port) is full**

  This represents the number of cycles during which zero instructions have been completed, because the oldest instruction being executed is waiting for the release of any store port.

  This state tends to occur when a cache does not contain the data intended to be stored there (cache lines).

- ■ **No instruction commit due to memory and cache busy**

  This represents the number of cycles during which zero instructions have been completed, because the memory and cache are busy.

- ■ **No instruction commit waiting for a branch instruction to be completed**

  This represents the number of cycles during which zero instructions have been completed, because the oldest instruction being executed is executing a branch instruction.

- ■ **No instruction commit waiting for an instruction to be fetched**

  This represents the number of cycles during which zero instructions have been completed, because the CSE(Commit Stack Entry) is empty.

  The CSE is a buffer for retaining information on instructions being executed (issued but not yet completed). This state tends to occur when instruction cache misses occur frequently.

- ■ **Synchronous waiting time between threads**

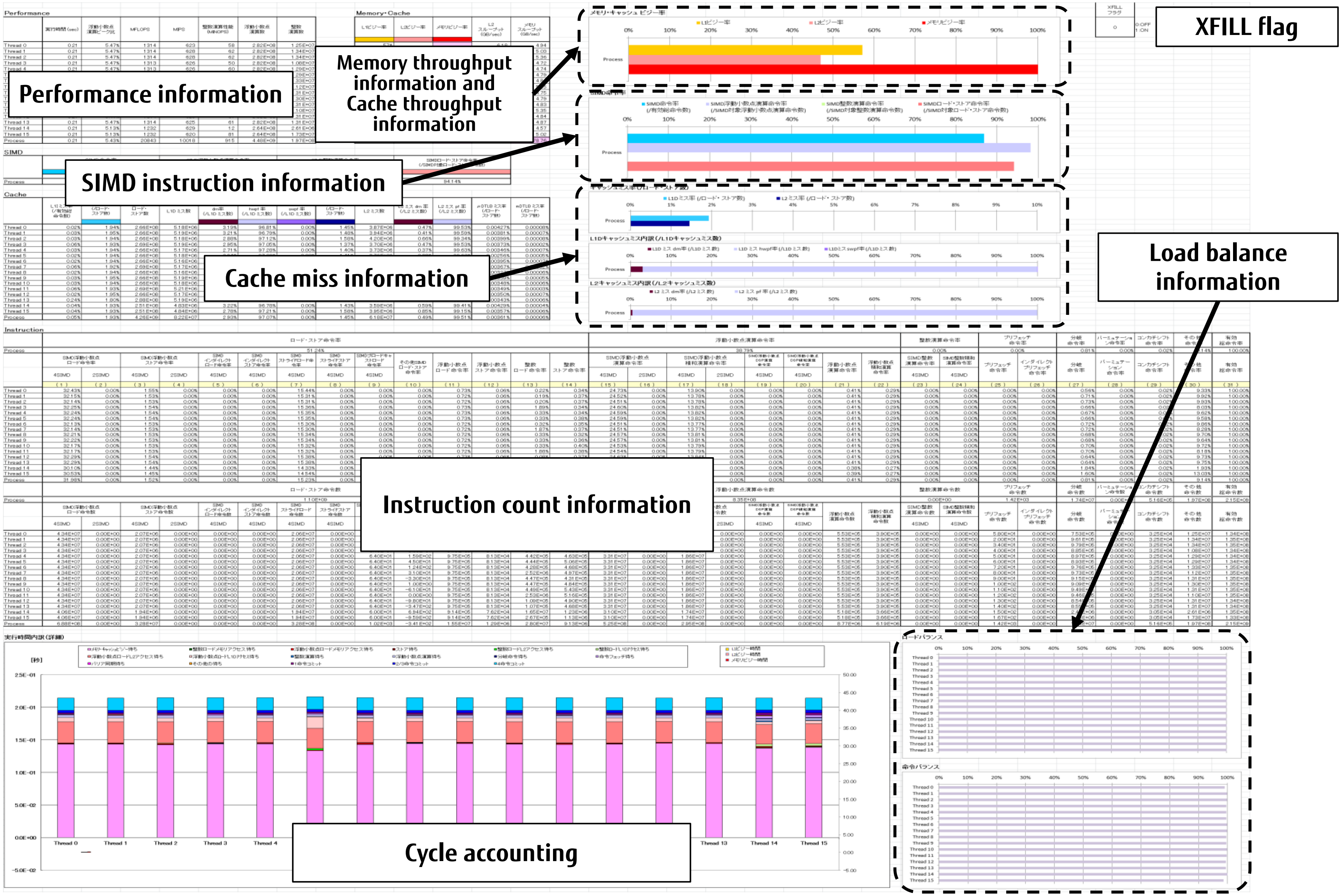
  This represents the number of cycles during which zero instructions have been completed, because the oldest instruction being executed has suspended the instruction controller with a SLEEP instruction.

PA Information

- PA (Performance Analysis) Information Report
- Performance Information
- Memory Throughput Information and Cache Throughput Information
- SIMD Instruction Information
- Cache Miss Information
- Instruction count information
- Load Balance Information
- XFILL Flag

PA (Performance Analysis) Information Report

Indicator



XFILL flag

Performance information

Memory throughput information and Cache throughput information

SIMD instruction information

Cache miss information

Load balance information

Instruction count information

Cycle accounting

Performance Information

The following information is output for performance information.

Performance

Output item	Meaning
Execution time (sec)	Shows the execution time of each thread. The Process row displays the longest execution time of all the threads.
Floating-point operation peak ratio	Shows the ratio of the actual performance value to the logical peak value of floating-point operations.
MFLOPS	Shows the floating-point operation performance.
MIPS	Shows the instruction performance.
Integer operation performance (MINOPS)	Shows the integer operation performance.
Floating-point operation	Shows the number of floating-point operations.
Integer operation	Shows the total number of integer operations. The number of integer operations includes instructions to calculate memory addresses.

Memory/Cache throughput

The following information is output for memory/cache throughput.

Memory/Cache

Output item	Meaning
L1 busy rate	Shows the value of the primary cache busy ratio. 80% to 100% is the upper limit of the reference value.
L2 busy rate	Shows the value of the secondary cache busy ratio. 80% to 100% is the upper limit of the reference value.
Memory busy rate	Shows the memory busy ratio. 90% to 100% is the upper limit of the reference value.
L2 throughput (GB/sec)	Shows the value of the secondary cache throughput.
Memory throughput (GB/sec)	Shows the value of the memory throughput.

* The L2 busy ratio is the operating ratio of a device that controls the transfer process between the primary data cache and secondary cache and the transfer process between the secondary cache and memory.

* The L1 busy ratio is the operating ratio of a device that controls the transfer process between a register and the primary data cache and the transfer process between the primary data cache and secondary cache.

SIMD Instruction Information

The following information is output as SIMD instruction information.

SIMD

Output item	Meaning
SIMD instruction rate (/Effective instruction)	Shows the ratio (%) of SIMD instructions to effective instruction.
SIMD floating point instruction rate (/SIMD target floating point instruction)	Shows the ratio (%) of SIMD instructions to the number of instructions targeted for SIMD optimization.
SIMD integer instruction rate (/SIMD target integer instruction)	Shows the ratio (%) of SIMD load instructions to the number of load instructions targeted for SIMD optimization.
SIMD load-store instruction rate (/SIMD target load-store instruction)	Shows the ratio (%) of SIMD store instructions to the number of store instructions targeted for SIMD optimization.

* A SIMD instruction is an instruction that processes multiple operands at one time.

Cache Miss Information (1/2)

The following information is output for cache miss information.

Cache

Output item	Meaning
L1I miss rate(/Effective instruction)	Shows the ratio (%) of the primary instruction cache misses to effective instruction.
L1D miss rate(/Load-store instruction)	Shows the ratio (%) of the primary data cache misses to the number of loads and stores.
Load-store instruction	Shows the total number of load and store instructions.
L1D miss	Shows the total number of primary data cache misses.
L1D miss dm rate(/L1D miss)	Shows the ratio (%) of the primary data cache misses due to load or store instructions to the number of primary data cache misses.
L1D miss hwpf rate(/L1D miss)	Shows the ratio (%) of the primary data cache misses due to hardware prefetch instructions to the number of primary data cache misses.
L1D miss swpf rate(/L1D miss)	Shows the ratio (%) of the primary data cache misses due to software prefetch instructions to the number of primary data cache misses.

*dm (demand) denotes an event due to a load or store instruction.
*hwpf (hardware prefetch) denotes an event due to a prefetch automatically analyzed and generated by hardware.
*swpf (software prefetch) denotes an event due to a prefetch instruction automatically analyzed and generated by the compiler.

Cache Miss Information (2/2)

Cache

Output item	Meaning
L2 miss rate(/Load-store instruction)	Shows the ratio (%) of secondary cache misses to the number of loads and stores.
L2 miss	Shows the total number of secondary cache misses.
L2 miss dm rate(/L2 miss)	Shows the ratio (%) of secondary cache demand misses.
L2 miss pf rate(/L2 miss)	Shows the ratio (%) of secondary cache prefetch misses.
µDTLB miss rate(/Load-store instruction)	Shows the ratio (%) of micro data TLB misses to the number of loads and stores.
mDTLB miss rate(/Load-store instruction)	Shows the ratio (%) of data main TLB misses to the number of loads and stores.

* dm (demand) denotes an event due to a load or store instruction.

* pf (prefetch) denotes an event due to a hardware or software prefetch instruction.

* µDTLB (micro data) retains address translation information, which is the basis of address translation by hardware.

* mDTLB (data main) retains address translation information to be kept in micro data. mDTLB is searched when the micro data does not have address translation information.

Instruction Count Information (1/3)

The following information is output for the instruction count information.

Instruction

Output item	Meaning
Load-store instruction rate	Shows the ratio (%) of load and store instructions. A detailed breakdown is displayed under this item. The breakdown has the following items: - Percentage of SIMD floating-point load instructions (4 SIMD, 2 SIMD) - Percentage of SIMD floating-point store instructions (4 SIMD, 2 SIMD) - Percentage of SIMD indirect load instructions (4 SIMD) - Percentage of SIMD indirect store instructions (4 SIMD) - Percentage of SIMD stride load instructions (4 SIMD) - Percentage of SIMD stride store instructions (4 SIMD) - Percentage of SIMD broadcast load instructions (4 SIMD) - Percentage of other SIMD load and store instructions - Percentage of floating-point load instructions - Percentage of floating-point store instructions - Percentage of integer load instructions - Percentage of integer store instructions

Instruction Count Information (2/3)

Instruction

Output item	Meaning
Floating-point instruction rate	Shows the ratio (%) of floating-point operation instructions. A detailed breakdown is displayed under this item. The breakdown has the following items: - Percentage of SIMD floating-point operation instructions (4 SIMD, 2 SIMD) - Percentage of SIMD floating-point multiply-add operation instructions (4 SIMD, 2 SIMD) - Percentage of SIMD floating-point DSP operation instructions (4 SIMD) - Percentage of SIMD floating-point DSP multiply-add operation instructions (4 SIMD) - Percentage of floating-point operation instructions - Percentage of floating-point multiply-add operation instructions
Integer instruction rate	Shows the ratio (%) of integer operation instructions. A detailed breakdown is displayed under this item. The breakdown has the following items: - Percentage of SIMD integer operation instructions (4 SIMD) - Percentage of SIMD integer multiply-add operation instructions (4 SIMD)

* DSP is an abbreviation for Dual Single Precision. The DSP operation instruction is an instruction that divides a double-precision register in half for handling as two single-precision floating point data.

Instruction Count Information (3/3)

Instruction

Output item	Meaning
Prefetch instruction rate	Shows the ratio (%) of prefetch instructions. A detailed breakdown is displayed under this item. The breakdown has the following items: - Percentage of prefetch instructions - Percentage of indirect prefetch instructions
Branch instruction rate	Shows the ratio (%) of branch instructions.
Permutation instruction rate	Shows the ratio (%) of permutation instructions.
Concatenate shift left instruction rate	Shows the ratio (%) of concatenation shift instructions.
Other instruction rate	Shows the ratio (%) of instructions other than those above.
Effective instruction rate	Shows the ratio (%) of the total number of the above instructions. This value is normally 100%.

* The concatenation shift instruction is an instruction to left shift by concatenating all the valid elements in a double-precision floating-point register.

* The permutation instruction is an instruction to change the order of read data within a register.

Load Balance Information

The following information is output for load balance information.

Balance

Output item	Meaning
Load balance	Shows the result of comparison of the time of “execution time - synchronous waiting time between threads" of each thread (the ratio to the thread with the longest execution time).
Instruction balance	Shows the result of comparison of effective instruction of each thread (the ratio to the thread with the most instructions).

XFILL Flag

After the output of the PA information, specify a value of 1 to the XFILL flag as necessary.

Function name	Usage and meaning
XFILL flag	Please set 1(ON) when you are using XFILL instruction. The default value for the XFILL flag is 0 (OFF).

* The XFILL instruction is an instruction to allocate a write cache line in the cache without loading data from memory.

Revision History

Version	Date	Revised section	Details
2.0	April 25, 2016	-	- First published

FUJITSU

shaping tomorrow with you