Introduction to the Fujitsu A64FX

Fujitsu A64FX: CPU of the World's "Fastest" Supercomputer

Academia

- Nano-science
- Particle physics

Government

- Long-range forecasting
- Disaster prevention

Oil and Gas

- Exploration and production
- Seismic analysis

Manufacturing

- Structural analysis
- Aerodynamics
- Computational fluid dynamics
- Crash test simulations

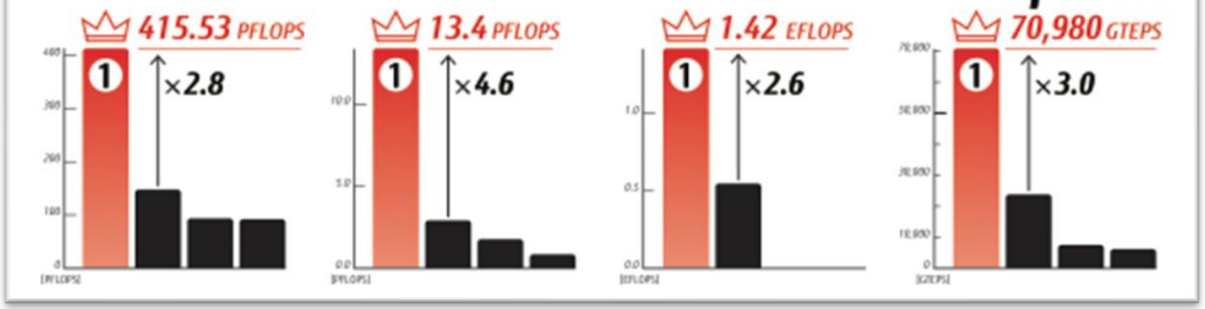
Systems Powered by A64FX

Supercomputer **Fugaku**

FUJITSU Supercomputer
PRIMEHPC FX1000
PRIMEHPC FX700

Awards

Fugaku Achieved First Place in Four Major Supercommputer Rankings (June 2020)



arm

World-leading Performance

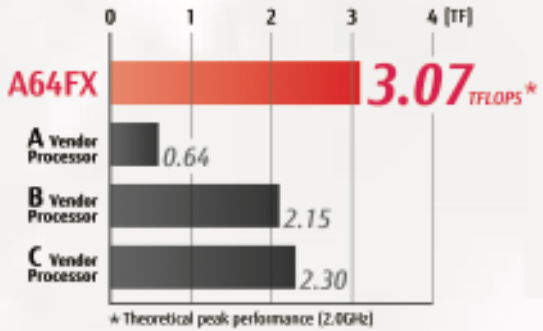
World-first combination of HBM2 and SVE 512-bit wide SIMD (No.1 of TOP500,HPCG,HPL-AI,Graph500,2020.6)

High throughput from

High memory bandwidth | **SVE 512-bit wide SIMD** | **Many core architecture**

A64FX is specifically designed for high performance in HPC

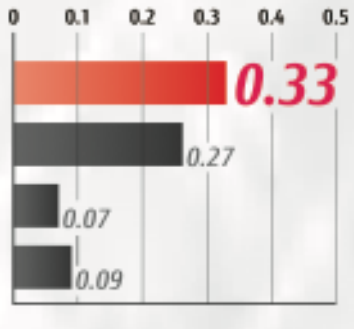
Peak Flops



* Theoretical peak performance (2.0GHz)

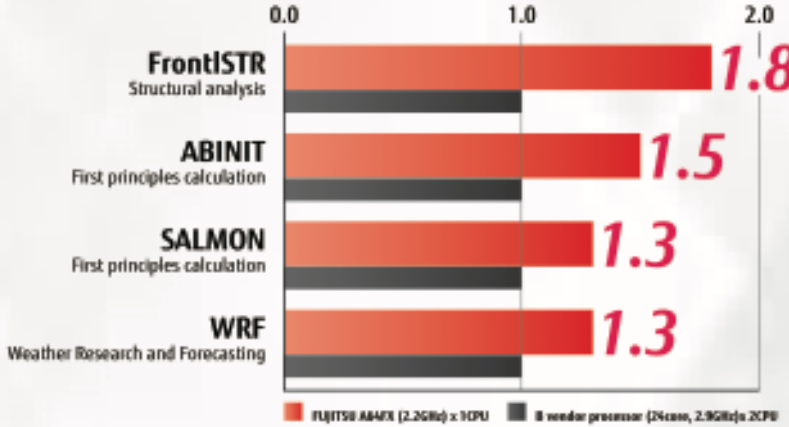
Peak Memory B/W

BF Ratio



High Performance in Real Apps

Relative speed up ratio (1 node)

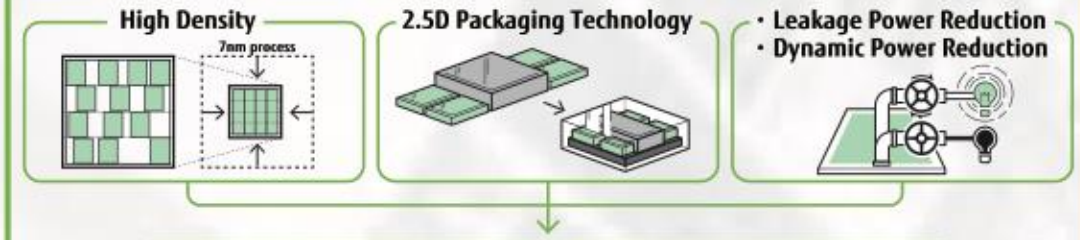


Measured on FUJITSU Supercomputer PRIMEHPC FX1000, A64FX 2.2GHz

For more information on other apps, please contact Fujitsu.

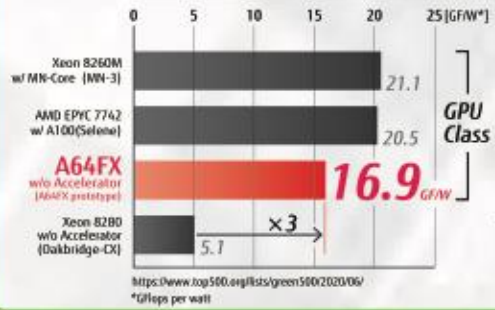
Evolved Power Efficiency

Fujitsu's circuit technology and power management



A64FX is the power efficient design for HPC

Power Performance

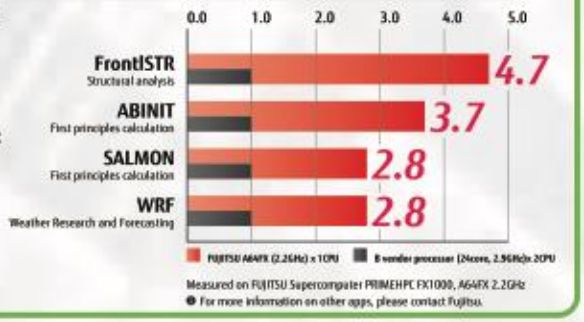


https://www.top500.org/lists/green500/2020/06/

*Gflops per watt

High Performance in Real Apps

Relative power efficiency ratio



Measured on FUJITSU Supercomputer PRIMEHPC FX1000, A64FX 2.2GHz

For more information on other apps, please contact Fujitsu.

Extensive Data Integrity

Unique 128,400 error checkers to correct or detect all 1-bit errors on a chip

~128,400 error checkers in total

https://www.fujitsu.com/downloads/SUPER/a64fx/a64fx_infographics.pdf

A Leadership CPU from start to finish

Expect excellent performance; expect to have to work to get it

Commodity HPC

- Mainline design
- Common assumptions hold
- Significant fraction of peak without tuning

Leadership HPC

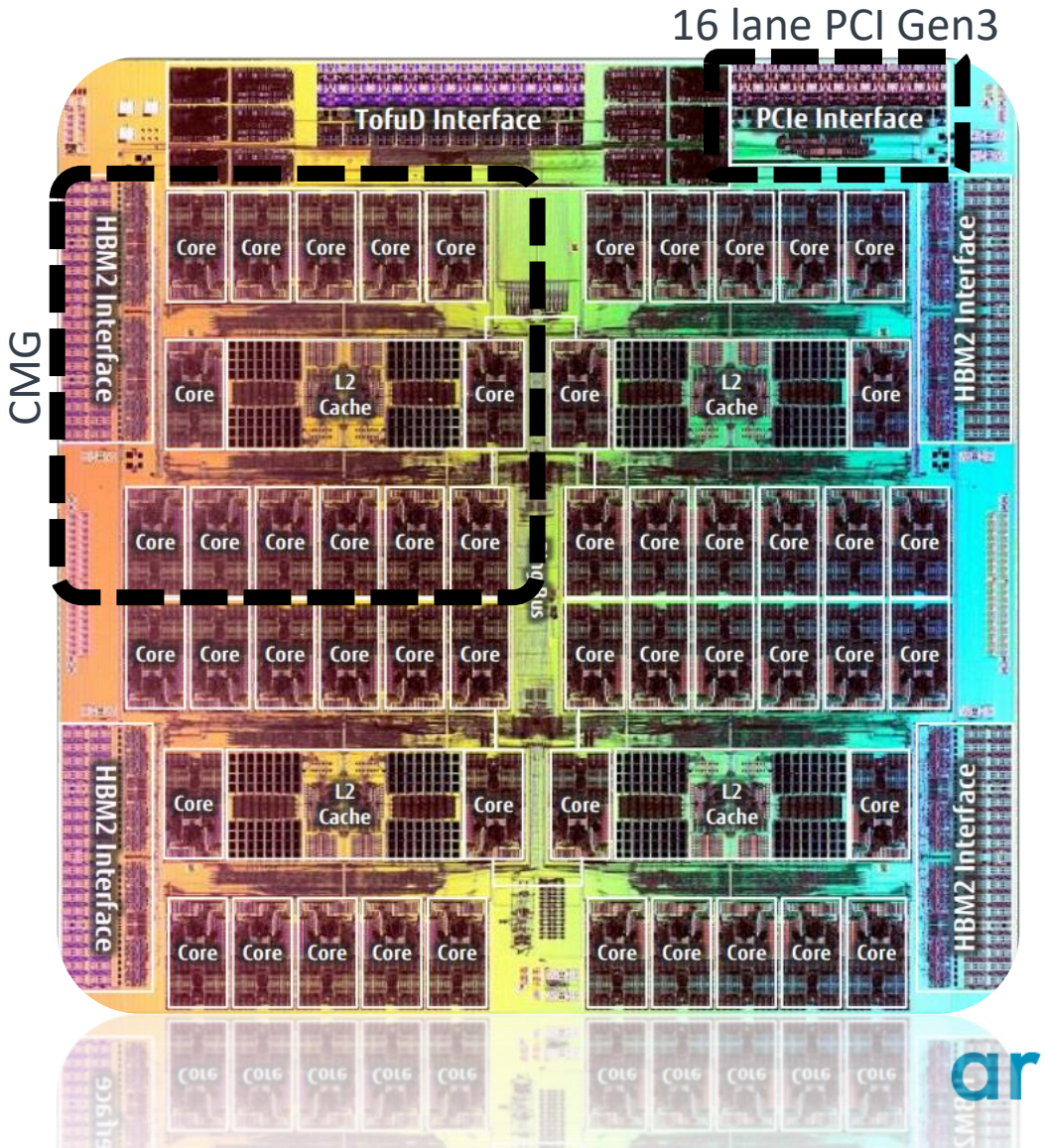
- Codesigned for specific application
- Common assumptions may hurt performance
- Significant tuning effort may be required

Confidential © 2019 Arm Limited
arm

Key Architecture Features

https://www.fujitsu.com/downloads/SUPER/a64fx/a64fx_datasheet.pdf

- Arm v8.2-A with 512-bit SVE
- Custom Fujitsu u-arch
- 7nm CMOS FinFET
- 2.2GHz, 2.0GHz, 1.8GHz
  - Constant clock: no turbo, no downclock
- **4 Core Memory Groups (CMGs)**
  - 12 cores (13 in the FX1000)
  - 64KB L1$ per core
    - 256b cache line
  - 8MB L2$ shared between all cores
    - 256b cache line
  - Zero L3$
  - 8 GB HBM at 256GB/s



Confidential © 2019 Arm Limited

arm

Hands On: FMLA

Confidential © 2019 Arm Limited

06_A64FX/01_fmla

See README.md for details

- Calculates peak per-core double precision flops
- Measures the wallclock time of a tight loop of fused multiply-add (FMLA) instructions.
- The code is written in Assembly, so the exact number of giga operations (GOP) is known.
- Performance in gigaflops (GFLOPS) is simply GFLOPS = GOP / SECONDS.

```
------------------------------------------------
./fmla_neon128.exe
256000000 Flops in 0.0194543 seconds
13.1591 GFlops
------------------------------------------------
./fmla_sve512.exe
1024000000 Flops in 0.0194653 seconds
52.6063 GFlops
------------------------------------------------
./fmla_a64fx.exe
960000000 Flops in 0.0150295 seconds
63.8742 GFlops
------------------------------------------------
```


Confidential © 2019 Arm Limited

Fujitsu A64FX Execution Pipeline

https://github.com/fujitsu/A64FX/tree/master/doc

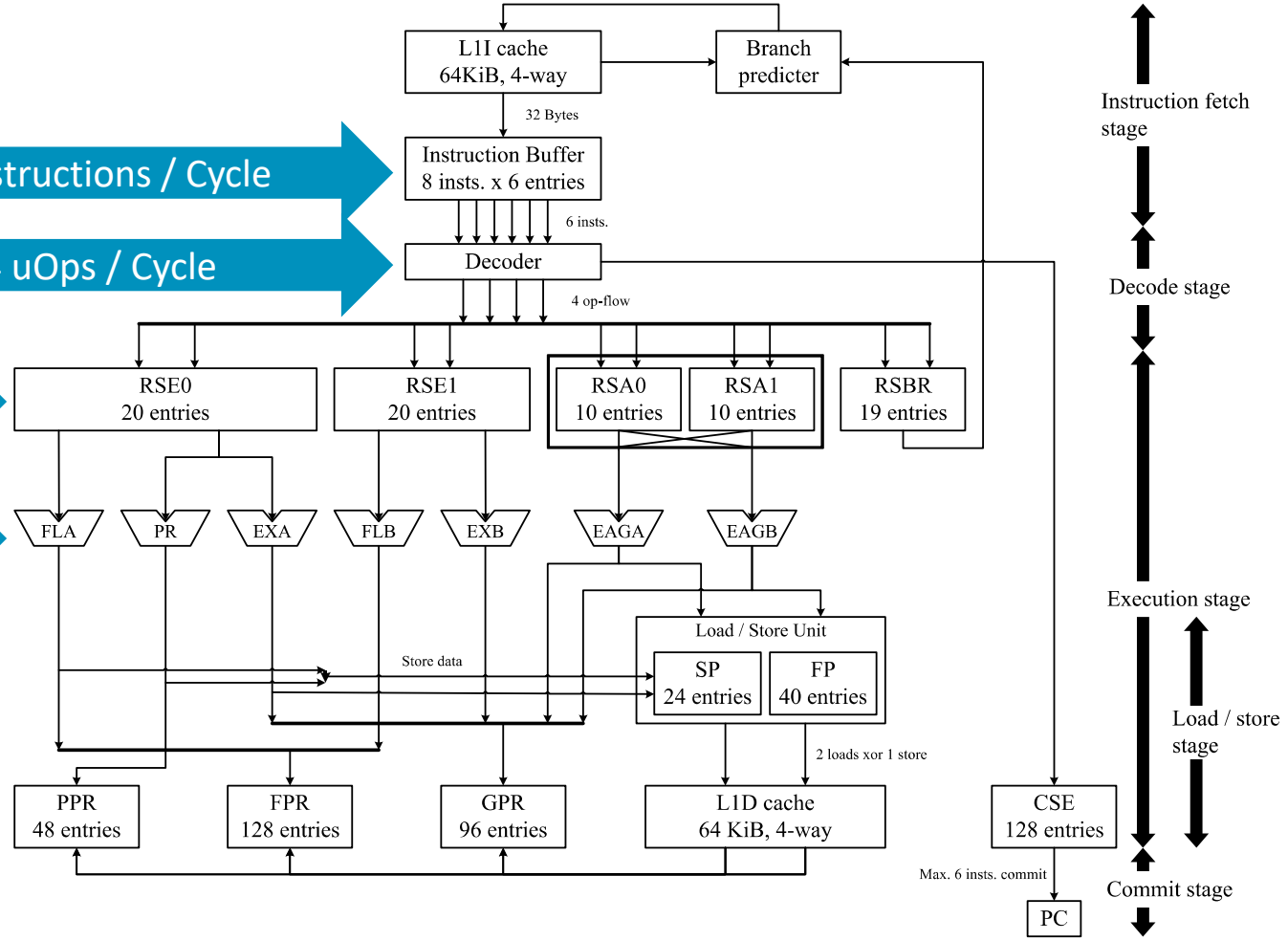
6 Instructions / Cycle

4 uOps / Cycle

2x2 Reservation Stations

Relatively high FP latencies

Optimal code generation is critical to performance



Confidential © 2019 Arm Limited

arm

Hands On: Stream

Confidential © 2019 Arm Limited

06_A64FX/02_stream/**01_stream_vanilla**

See README.md for details

- A basic, untuned, out-of-box, "vanilla" implementation
  - Performance will most likely be very poor
  - Uses only a single core and does not consider NUMA or any architectural features

GCC 11 on A64FX

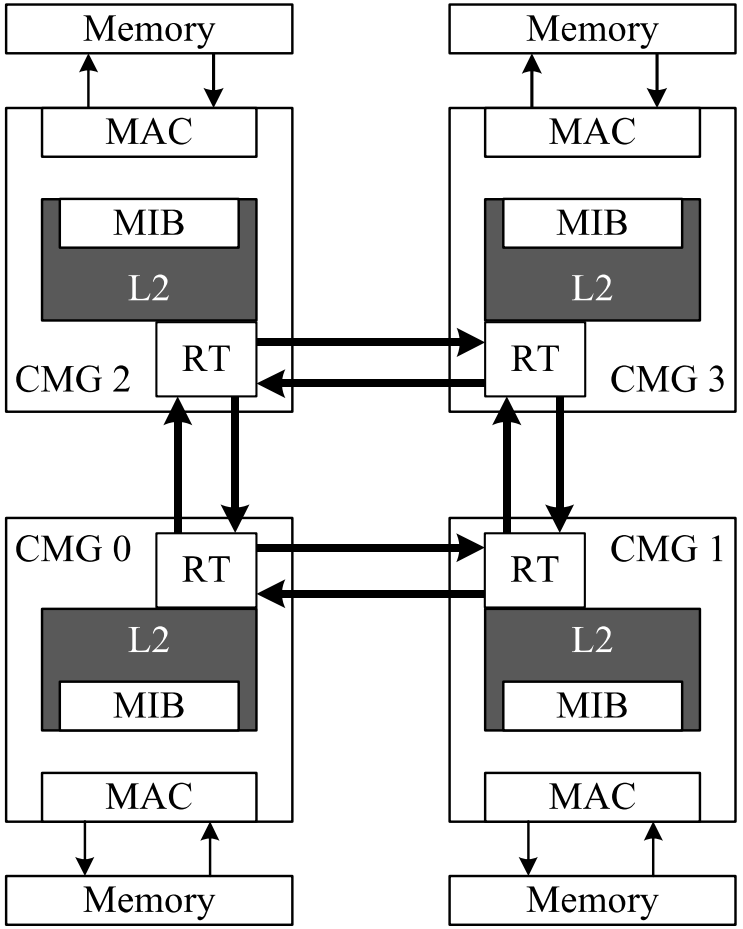
```
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           40859.3     0.003931     0.003916     0.003981
Scale:          40796.5     0.003931     0.003922     0.003949
Add:            47235.1     0.005109     0.005081     0.005188
Triad:          47253.3     0.005096     0.005079     0.005114
-------------------------------------------------------------
```


Confidential © 2019 Arm Limited

Fujitsu A64FX L1 Cache

https://github.com/fujitsu/A64FX/tree/master/doc

NEON and GP registers support L1D parallel L/S
SVE: ***either*** load 2x64b/cycle ***or*** store 1x64b/cycle



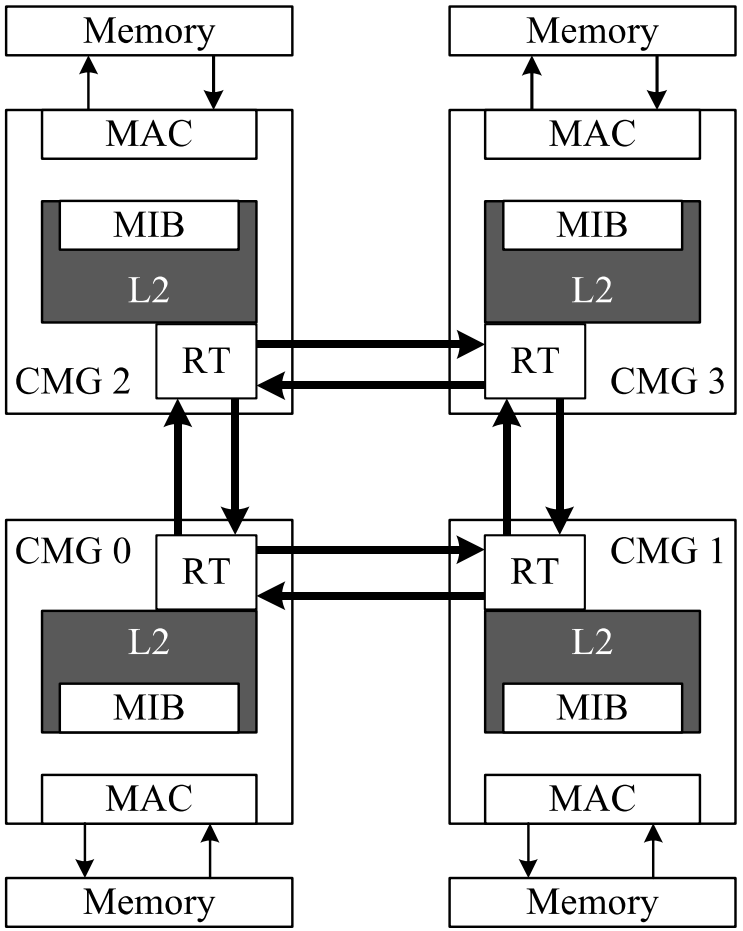
		For Instruction	For Data
L1 cache	Association method	4-way set associative	4-way set associative
	Capacity	64 KiB	64 KiB
	Hit latency (load-to-use)	4 cycles	5 cycles(integer)
			8 cycles (SIMD&FP / SVE in short mode)
			11 cycles (SIMD&FP / SVE in long mode)
	Line size	256 bytes	256 bytes
	Write method	---	Writeback
	Index tag	Virtual index and physical tag (VIPT)	Virtual index and physical tag (VIPT)
	Index formula	index_A = (A mod 16,384) / 256	index_A = (A mod 16,384) / 256
	Protocol	SI state	MESI state

Confidential © 2019 Arm Limited
arm

Fujitsu A64FX L2 Cache

https://github.com/fujitsu/A64FX/tree/master/doc

L1D -> L2 store BW is ~50% load BW
TPeak Load: 922GB/s
TPeak Store: 461GB/s



		For instruction and data (by shared)
L2 cache (shared by instruction & data)	Association method	16-way set associative
	Capacity	8 MiB
	Hit latency (load-to-use)	37 to 47 cycles
	Line size	256 bytes
	Write method	Writeback
	Index and tag	Physical index and physical tag (PIPT)
	Index formula	index <10:0> = ((PA<36:34> xor PA<32:30> xor PA<31:29> xor PA<27:25> xor PA<23:21>) << 8) xor PA<18:8>
	Protocol	MESI state

Confidential © 2019 Arm Limited
arm

06_A64FX/02_stream/**02_stream_openmp**

See README.md for details

- Uses OpenMP and numactl to improve memory/thread locality
  - On many systems, this implementation will be close to 80% of the theoretical peak bandwidth
  - Does not achieve 80% of peak on A64FX due to that system's memory architecture

GCC 11 on A64FX

```
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           537948.0     0.032011     0.031936     0.032123
Scale:          537695.1     0.032026     0.031951     0.032179
Add:            597172.3     0.043259     0.043153     0.043500
Triad:          597324.1     0.043282     0.043142     0.044186
-------------------------------------------------------------
```


Confidential © 2019 Arm Limited

06_A64FX/02_stream/**04_stream_zfill**

See README.md for details

- Uses Arm's DC ZVA instruction to zero-fill cache lines
  - Dramatically improves the performance of systems with wide L2$ lines and low L3$

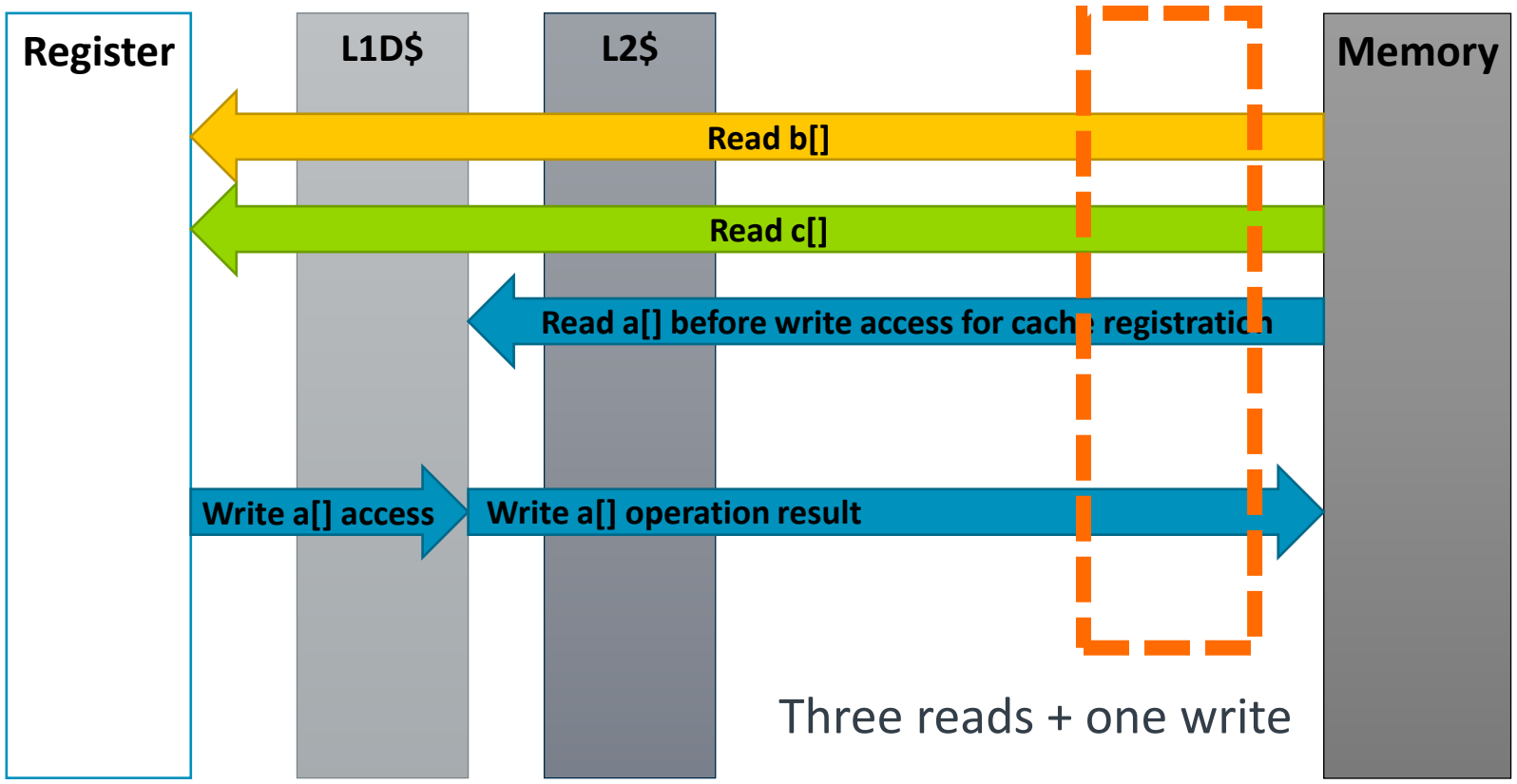
GCC 11 on A64FX

```
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           780579.1     0.022083     0.022009     0.022202
Scale:          780689.0     0.022146     0.022006     0.022576
Add:            788330.3     0.032902     0.032689     0.033698
Triad:          787974.0     0.032808     0.032704     0.033263
-------------------------------------------------------------
```


Confidential © 2019 Arm Limited

Unoptimized STREAM TRIAD

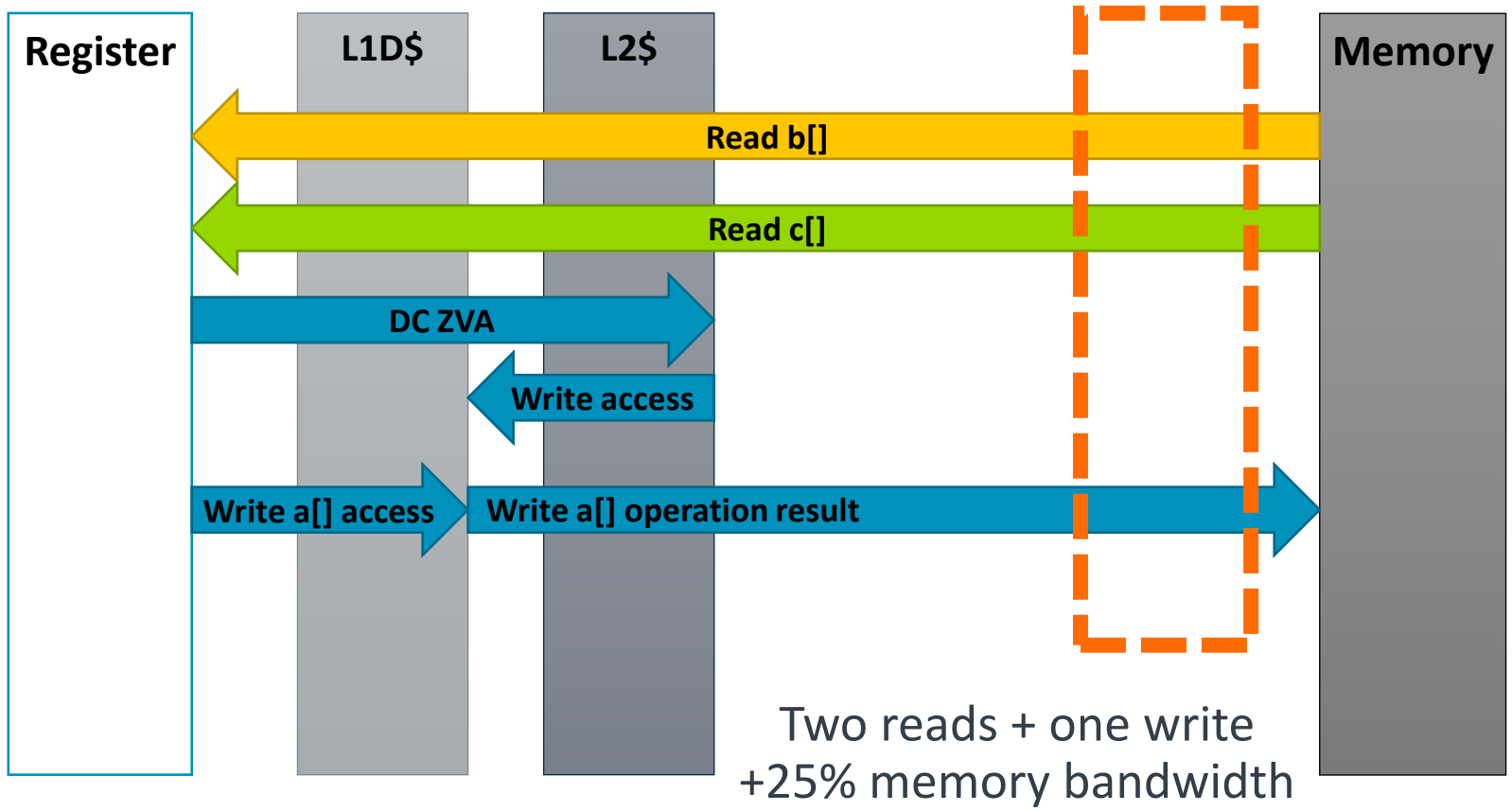
There is a hidden read before write of the result array a[] for cache registration



Confidential © 2019 Arm Limited

Use ZFILL to eliminate useless memory access

DC ZVA instruction maps cache without reading main memory



Confidential © 2019 Arm Limited

06_A64FX/02_stream/**05_stream_fujitsu**

See README.md for details

- Uses the Fujitsu compiler to maximize bandwidth on A64FX
  - No inline assembly
  - Compiler automatically inserts ZFILL instructions as needed

fcc 4.2.1 on Fujitsu A64FX

```
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           755455.3     0.022814     0.022741     0.022887
Scale:          768704.5     0.022393     0.022349     0.022464
Add:            819550.3     0.031496     0.031444     0.031545
Triad:          815555.5     0.031672     0.031598     0.031754
-------------------------------------------------------------
```


Confidential © 2019 Arm Limited

arm

Hands-on: Energy Measurement

Confidential © 2019 Arm Limited

Power measurement on FX700

- **A64FX_PMU_Events_v1.2.pdf in https://github.com/fujitsu/A64FX/blob/master/doc/**

Event number	Name	Description	Unit (estimated using Fujitsu CPU profile data)
0x01e0	**EA_CORE**	This event counts energy consumption per cycle of core.	**8.04 nJ**
0x03e0	**EA_L2**	This event counts energy consumption per cycle of L2 cache. It counts all events caused in measured CMG regardless of measured PE.	**32.8 nJ**
0x03e8	**EA_MEMORY**	This event counts energy consumption per cycle of CMG local memory. It counts all events caused in measured CMG regardless of measured PE.	**271 nJ**

- **Perf stat commands can get the PMU events counter**

```
$ perf stat -e r01e0 -e r03e0 -e r03e8 ./test.axf
928,677,262      r01e0
194,605,353      r03e0
  22,702,240      r03e8
3.322102610 seconds time elapsed
```

- These values are energy, so you want to get power, you must divide by time.
- There are no document about the unit of energy, but we estimated using the Fujitsu CPU profile data.
- Core Power is 928E6 * 8.04E-9 / 3.3s=2.25W
- Note that L2 and Memory energy is per CMG.

06_A64FX/03_energy

See README.md for details

- Uses the A64FX PMU counters to calculate energy consumption
- Runs two versions of the HACC kernel and gathers PMU data to two "CSV" files:
  - See neon.perf and sve.pef
- A Python script post-processes the raw PMU data to calculate energy consumption

ACfL 20.3 on Fujitsu FX700

```
perf stat -x\; -o neon.perf -e duration_time,r11,r1e0,r3e0,r3e8 ./hacc_arm_neon.exe 1000
Maximum OpenMP Threads: 1
Iterations: 1000
Gravity Short-Range-Force Kernel (5th Order): 12823.6 -444.108 -645.349: 1.30715 s
perf stat -x\; -o sve.perf -e duration_time,r11,r1e0,r3e0,r3e8 ./hacc_arm_sve.exe 1000
Maximum OpenMP Threads: 1
Iterations: 1000
Gravity Short-Range-Force Kernel (5th Order): 12823.6 -444.108 -645.349: 1.03654 s
./postproc_perf_energy.py neon.perf -c 8.04 -l 32.8 -m 271
Elapsed Time: 1.32164e+09 ns
Core freq: 1.78475 GHz
Per-core power: 1.74583 Watt
Per-CMG L2 power: 1.71094 Watt
Per-CMG HBM power: 1.8449 Watt
./postproc_perf_energy.py sve.perf -c 8.04 -l 32.8 -m 271
Elapsed Time: 1.04998e+09 ns
Core freq: 1.786 GHz
Per-core power: 1.62475 Watt
Per-CMG L2 power: 1.7082 Watt
Per-CMG HBM power: 1.84033 Watt
```


Confidential © 2019 Arm Limited

Resources

- Fujitsu A64FX Microarchitecture Manual
  - https://github.com/fujitsu/A64FX/tree/master/doc
- Fujitsu A64FX Performance Monitor Unit (PMU) events
  - https://github.com/fujitsu/A64FX/tree/master/doc
- Japan and Fugaku's Fight Against the COVID-19 in HPC, AHUG SC'20
  - https://www.youtube.com/watch?v=Qma7UuYifhM
- ML and HPC with Supercomputer Fugaku and its Processor A64FX, AHUG SC'20
  - https://www.youtube.com/watch?v=3TYVqodc8w4
- Cray Apollo 80 Hardware Description
  - https://pubs.cray.com/bundle/HPE_Cray_Apollo_80_Hardware_Guide_H-6220/page/Product_Description.html
- Fujitsu PRIMEHPC Documentation
  - https://www.fujitsu.com/global/products/computing/servers/supercomputer/documents/

Confidential © 2019 Arm Limited
arm