



arm

CoreLink AHB Cache (CG095)

Boost system performance for emerging use cases

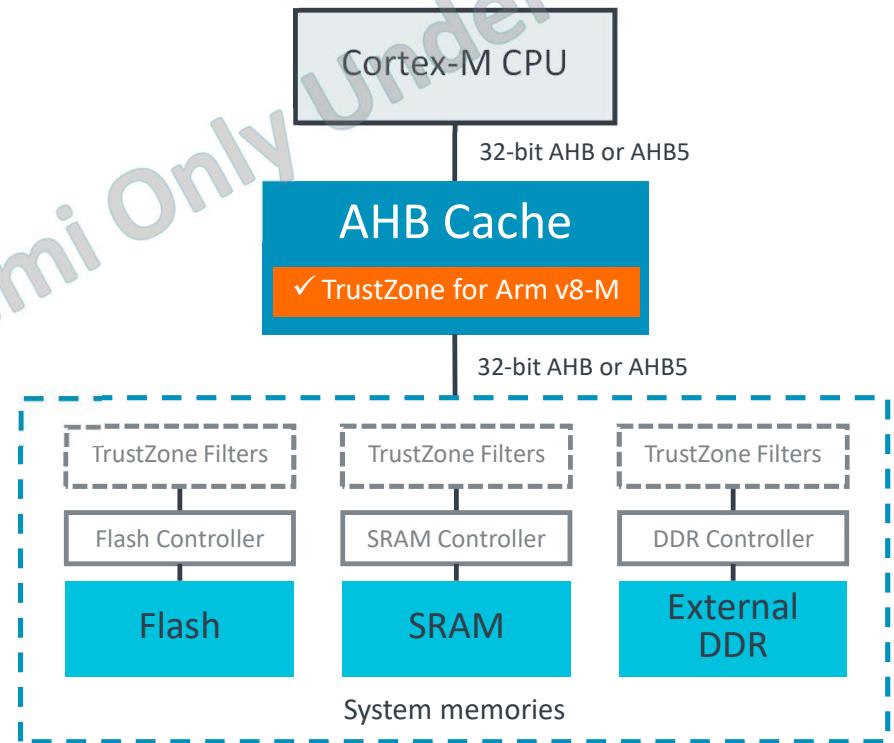
Sam Bruggen (A&I LoB)
2022

Overview of CoreLink AHB Cache (CG095)

Maturity Status: REL Release

- + Support for TrustZone for Arm v8-M
- + Compatible with Cortex-M0, Cortex-M0+, Cortex-M3, Cortex-M4, Cortex-M23, Cortex-M33
- + Versatile as a cache for data or code or both
- + Lower power due to reduced system memory access
- + Lower overall memory access latency when cache hit rate is high
- + Runs at processor frequency depending on cache configuration*
 - + Approx. 165 MHz for TSMC 40 ULP, C40, 9 track library with SVT
 - + Approx. 440 MHz for TSMC 22 ULL, C30, 7 track library with SVT

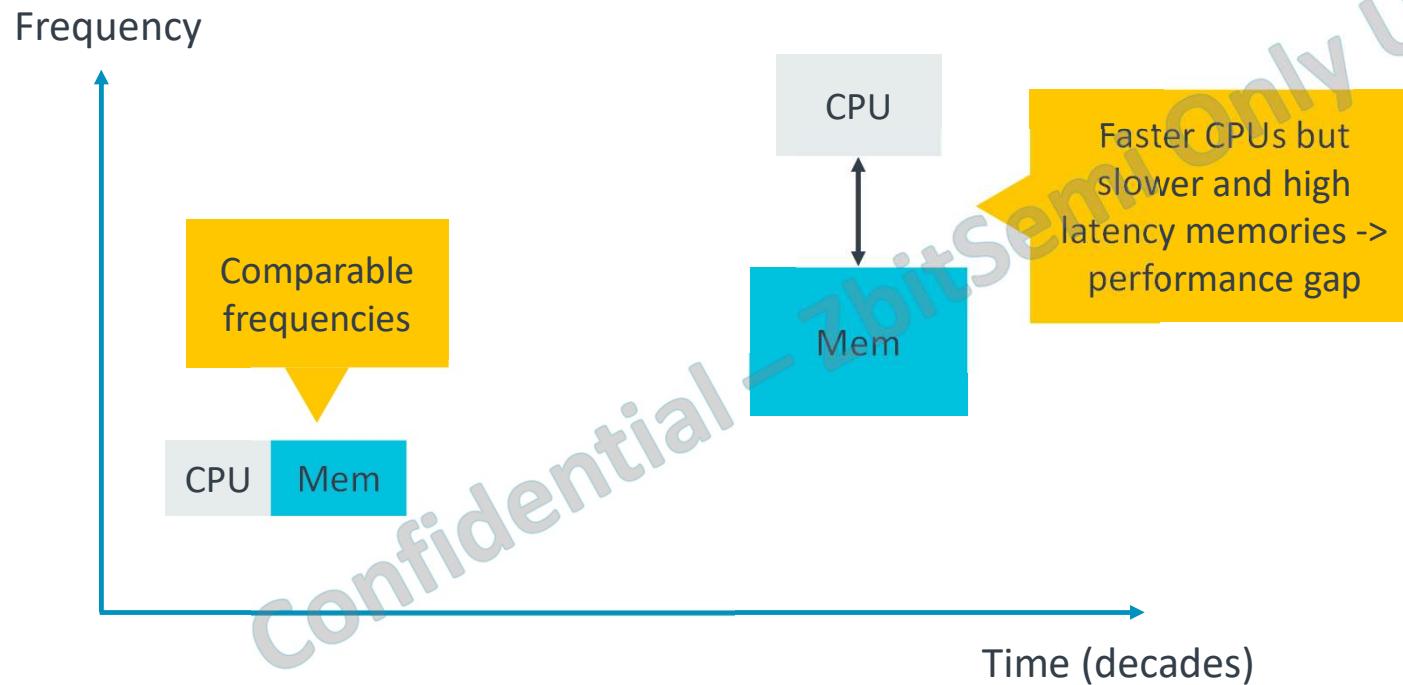
* Data based on AHB Cache EAC, subject to change



CoreLink AHB Cache (CG095) TRM: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.101807_0000_02_en/index.html

Technology Trends affecting MCU System Performance

Need for a cache

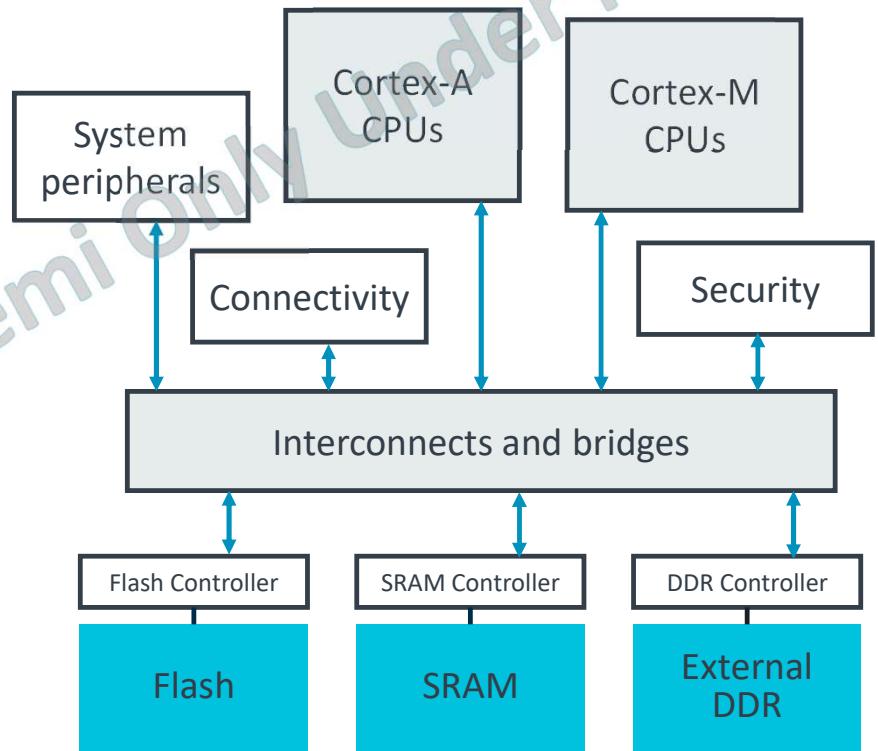


- + The problem of slower memories worsens at smaller process nodes
 - Flash, SRAM, DDR, PSRAM
- + Many wait cycles severely impact performance of AHB CPUs without caches

Heterogenous Multi-processor Systems with Cortex-M inside

Need for a cache

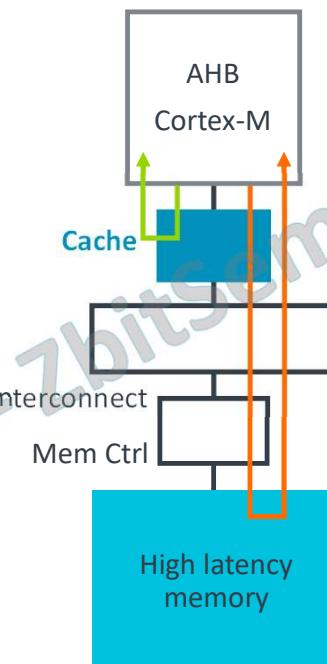
- + Example Cortex-M use-cases in such systems
 - Bluetooth stack in an IoT endpoint
 - Always-on sensor processing
 - Modem function in a cellular connectivity chip
- + Typical characteristics of rich heterogenous high-end systems
 - AXI interconnect leading to AHB-AXI bridge in the path from AHB Cortex-M to memory
 - Cortex-M is asynchronous to system memory leading to bridges in the path
 - High latency path from Cortex-M to DRAM memory



Solution: Cache Reduces Impact of High Memory Latency

A versatile cache for data and code gives optimal performance

Majority of the memory accesses are likely to be cache hit

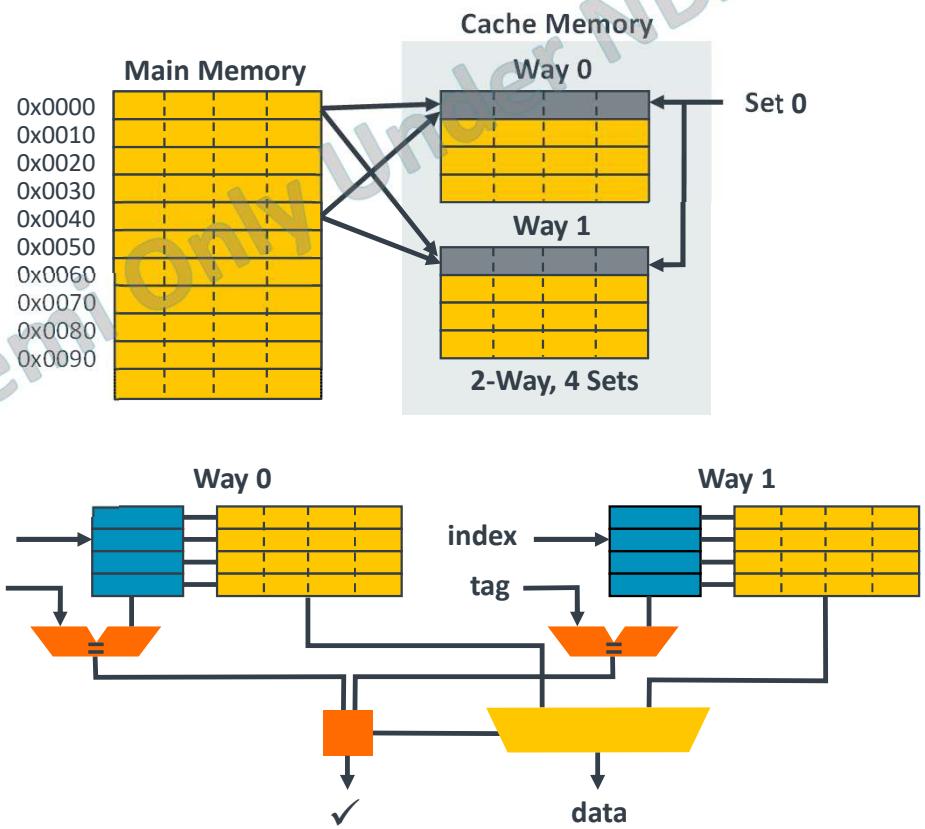


→ **Cache hit** means no wait state for Cortex-M (ideal performance)

→ **Cache miss** means wait states that depend on the (high) latency to memory

Cache Organisation and Terminology

- Cache Line, Set, Way, Tag and Index
- Cache Hit and Cache Miss
- Valid and Dirty Cache Line
- Cache Replacement Strategy, Cache Line Eviction
- Write-Through mode
 - Write updates both the Cache and the external memory system
 - Does not produce Dirty data
- Write Back mode
 - Write Cache Hit – Updates the Cache only (data is marked as Dirty)
 - Write Cache Miss – For Allocate, trigger a Linefill and mark Dirty. For No-Allocate, forward write to external memory system (via the write buffer)
 - Eviction of dirty data results in write of the Cache Line to external memory



CoreLink AHB Cache Key Features

Key differentiating features of AHB Cache

Zero bus wait state for Cache Hit, critical word first support for LineFill with data sent to CPU immediately

4-way set associative, a set is spread across four memory banks

Optional support for TrustZone for Arm v8-M

Support for software cache maintenance operations

Cache size configurable to 2KB, 4KB, 8KB, 16KB, 32KB, 64KB

Cache line size of 32 bytes (8 words per line and address is word aligned)

Configurable Write-Through or Write-Back policy

Pseudo-random replacement policy

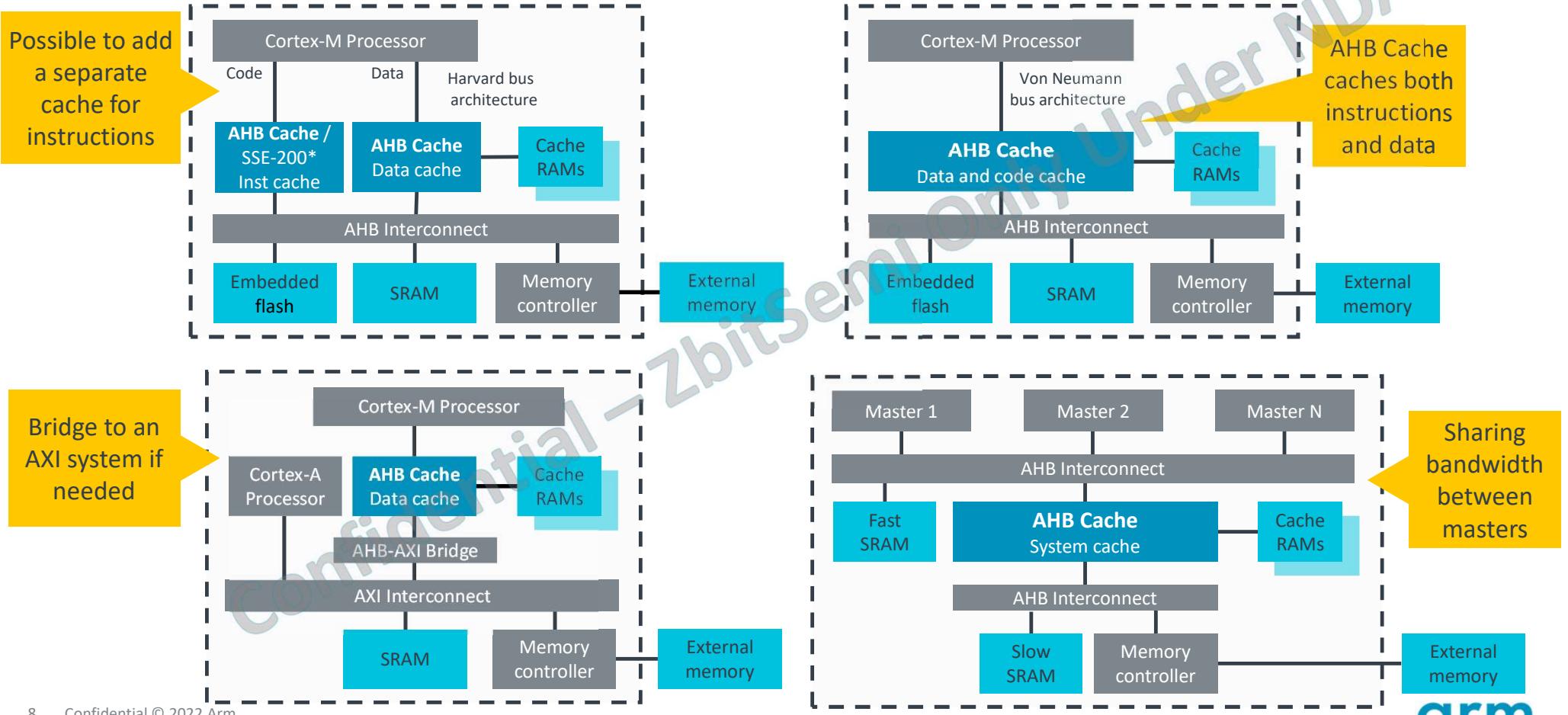
Q-Channel interfaces for clock and power control

Optional support for Execute Only Memory (XOM)

Hit and miss counters separate for secure and non-secure, configurable snapshotting functionality

Write buffers, Linefill buffer, Eviction writeback buffer

AHB Cache is Designed to address a Variety of Use-cases





Technical Details of CoreLink AHB Cache

(CG095)

AHB Cache Behavior and Timing Details

AHB Cache Behavior and Timing Details

Zero bus wait state when there is a Cache Hit

A Linefill is triggered when there is a miss for a Cacheable access that is also Allocate.

Linefill sends a wrapping burst request (8 words per burst) to the memory system starting at the critical word. The response is streamed to the CPU immediately without added latency.

No write after read penalty

No penalty on writing into a clean line

Non-cacheable read accesses bypass the cache and add no latency

Non-cacheable Bufferable write accesses get buffered and the cache responds early to CPU

Non-cacheable Non-bufferable write accesses get forwarded without delay if the write buffer is empty

AHB Cache Configuration Parameters for RTL rendering

Description	Value
Size of Data RAM	2KB to 64KB
User bus width of HAUSER, HRUSER, HWUSER	0-64
Width of Master ID (HMASTER_WIDTH)	4-8
The Master ID that the cache will generate	0- $2^{\text{HMASTER_WIDTH}-1}$
Endianness support	LE, BE8, BE32
XOM support	OFF, ON
Snapshot support	OFF, ON
Clock Q synchronizer	OFF, ON

RAM Organisation and Geometry

- 4 Tag RAMs
 - Support to join into a single RAM instance
 - Single write enable for each Tag RAM
- 4 Data RAMs
 - Write enable is byte granularity
- 1 RAM for Dirty status bits
 - Write enable is bit granularity

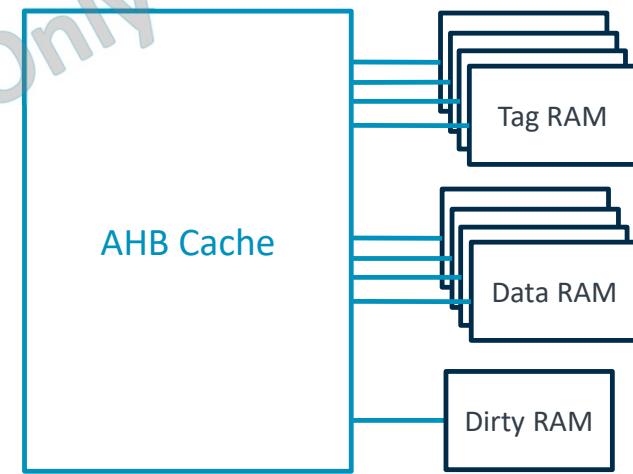
RAM Sizes and Geometry for a 64KB Cache Size

Cache Size / cache line size = 64KB / 32KB = **2KB cache lines**

Data RAM size = 64KB

Tag RAM size = 2KB*20 or 2KB*21 bits depending on XOM config.

Dirty RAM size = 2KB bits



Cache Maintenance Operations

Software Controlled and Automatic

- + Maintenance operations supported via APB interface (during run time)
 1. *Clean by address* – look up address, if look up is a hit and cache line is dirty then write back to memory
 2. *Clean all* – walk through all cache lines and write back dirty lines to memory
 3. *Invalidate by address* – look up address, if look up is a hit then cache line is invalidated (secure software only)
 4. *Invalidate all* – walk through all cache lines and invalidate them (secure software only)
 5. *Clean and invalidate by address* – combination of 1 & 3
 6. *Clean and invalidate all* – combination of 2 & 4
- + Support for automatic maintenance which can be enabled/disabled via config. ports
 1. **Powerdown** - perform *clean all* when low-power request is received
 2. **Cache enable** - perform *invalidate all* in the background (traffic is unaffected) and then enable cache
 3. **Cache disable** - stall slave interface, perform *clean all*, then disable cache and release slave interface

TrustZone Support in AHB Cache

- Secure and non-secure data is isolated from each other, this is how it is achieved:
 - The Cortex-M processor with TrustZone generates the security attribute by checking the SAU and IDAU
 - The TrustZone security attribute set by the transfer that triggers the Linefill is stored in the Tag RAM
 - Non-secure transfer is not able to read Secure data and vice versa
- User needs to manage cache maintenance if secure/non-secure partition changes when reprogramming the SAU. There are two ways to handle this:

Manual maintenance

Clean all manual operation

Disable the Cache

Reprogram the SAU

Invalidate all manual operation

Enable the Cache

Automatic maintenance (config. param)

Disable the Cache (this performs *Clean all* before disabling the cache)

Reprogram the SAU

Enable the Cache (this performs *Invalidate all* before enabling the cache)



Benchmark Data for CoreLink AHB Cache

(CG095)

arm

FPMark

Cortex-M33 and Cortex-M4

FPMark context for AHB Cache Benchmark Data

FPMark is a floating-point benchmark suite from EEMBC: <https://www.eembc.org/fpmark/>

General workload name	Specific workload name
ArcTan	atan-1k-sp
Black Scholes	blacks-sml-500v20-sp
Horner's method	horner-sml-1k-sp
Livermore loops	loops-all-tiny-sp
Neural net	nnet-data1-sp
Linear algebra	linear_alg-sml-50x50-sp
Matrix LU decomposition	lu-sml-20x2_50-sp
Inner product	inner_product-sml-sp

Workloads compiled with Arm Compiler 6.14 with below flags – Cortex-M33

Compiler flags: --target=arm-arm-none-eabi -mcpu=cortex-m33 -D__ARM_ARCH_8M_=1 -march=armv8-m.main -mcmse -mthumb -munaligned-access -O3 -munaligned-access -Omax -MMD -g -c

Linker flags: --map --ro-base=0x0 --rw-base=0x20000000 --first='boot_exectb_mcu.o(vectors)' --datacompressor=off --info=inline --entry=main --cpu=teal --lto -Omax

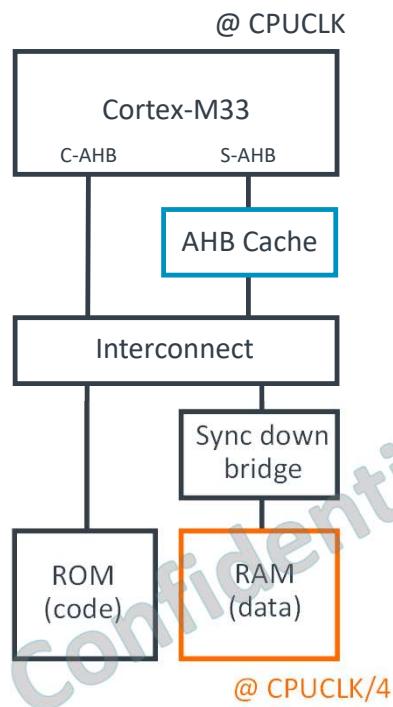
Workloads compiled with Arm Compiler 6.14 with below flags – Cortex-M4

Compiler flags: -mcpu=Cortex-m4 --target=arm-arm-none-eabi -g -Omax -mfloating-abi=hard -mfpu=fpv4-sp-d16 -mthumb -fno-common -ffunction-sections -funsigned-char -fshort-enums -fshort-wchar -gdwarf-3 -ffp-mode=fast -mcpu=Cortex-m4 -O3 -munaligned-access -g -c

Linker flags: --ro-base=0x0 --rw-base=0x20100000 --first='startup_CMSDK_CM4.s.o(RESET)' --info=inline --datacompressor=off --debug --map -Omax --cpu=Cortex-M4 --fpu=FPV4-SP --load_addr_map_info --xref --callgraph --symbols

Cortex-M33 & SRAM Performance Uplift with AHB Cache (1)

Cortex-M33 CPU clock to system memory clock is 1:4



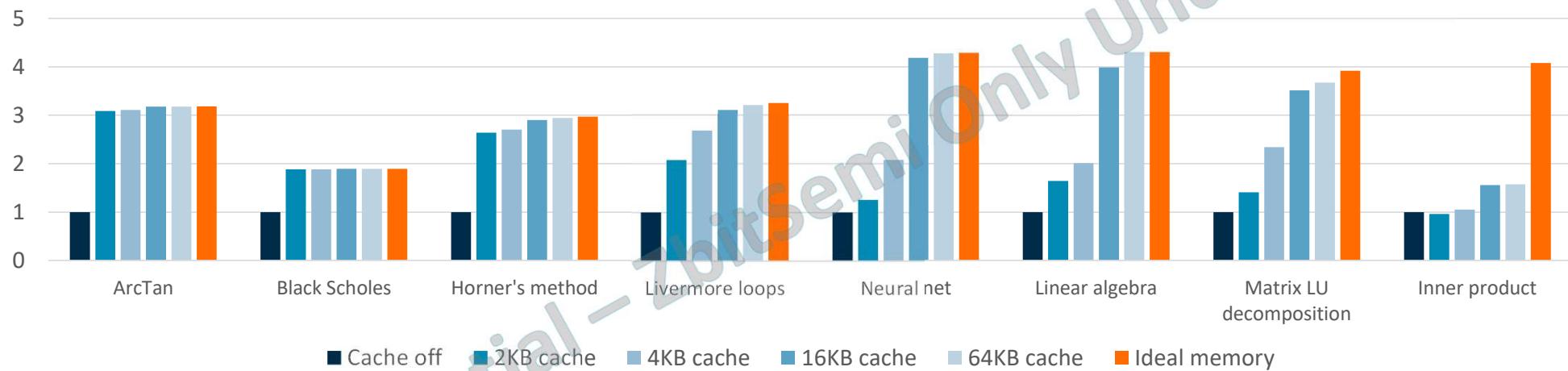
+ System setup

- Cortex-M33 is an Armv8-M processor and has two AHB5 master interfaces
- Code access at 1:1 clock, data access at 1:4 clock
- AHB Cache size configuration is swept
- Cortex-M33 Floating Point Unit (FPU) is enabled

Cortex-M33 & SRAM Performance Uplift with AHB Cache (2)

Cortex-M33 CPU clock to system memory clock is 1:4

Performance of Cortex-M33 with FPU, FPMark single precision small data set, Write Back policy



X-axis shows workloads from the [FPMark suite](#) - single precision versions with small data set.

Y-axis is the performance of each configuration setting relative to the performance of the 'cache off' setting.

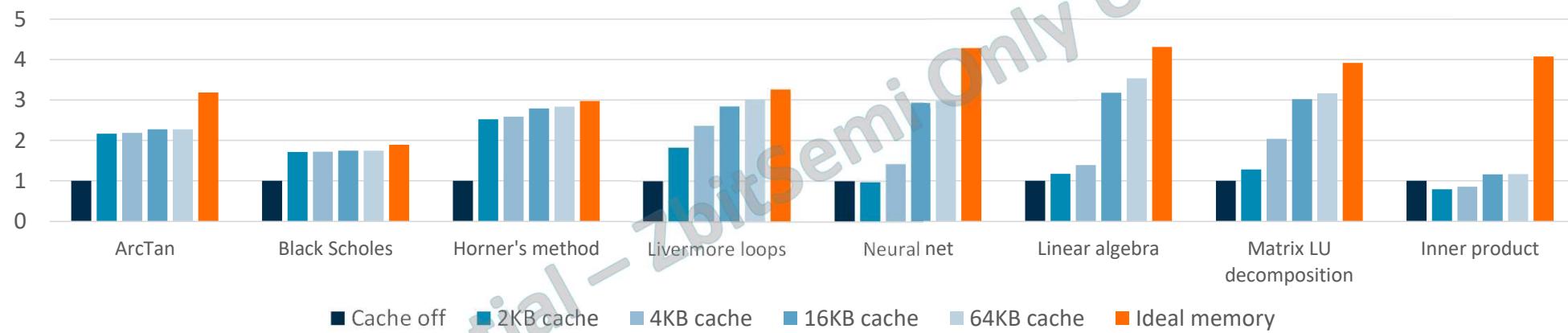
When cache is on, write policy is [Write-Back](#).

Code memory accesses are zero wait state for all runs. Ideal memory means zero wait state for data.

Cortex-M33 & SRAM Performance Uplift with AHB Cache (3)

Cortex-M33 CPU clock to system memory clock is 1:4

Performance of Cortex-M33 with FPU, FPMark single precision small data set, Write-Through policy



X-axis shows workloads from the [FPMark suite](#) - single precision versions with small data set.

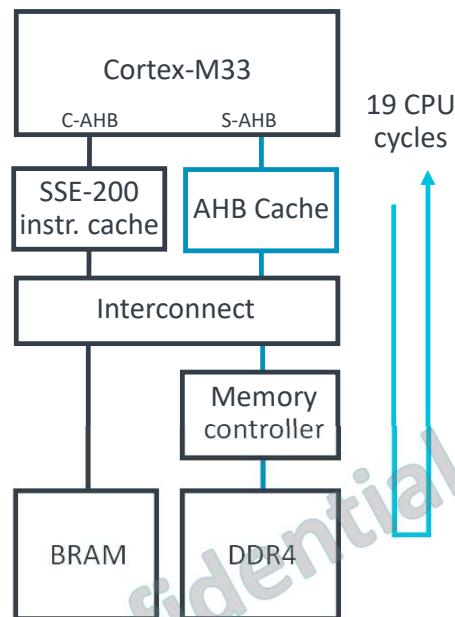
Y-axis is the performance of each configuration setting relative to the performance of the 'cache off' setting.

When cache is on, write policy is [Write-Through](#).

Code memory accesses are zero wait state for all runs. Ideal memory means zero wait state for data.

Cortex-M33 & DDR4 Performance Uplift with AHB Cache (1)

Round trip latency for a read to DDR4 memory is 19 CPU cycles



- System based on SSE-200 MPS3 FPGA
 - Cortex-M33 is an Armv8-M processor and has two AHB5 master interfaces
 - AHB Cache (CG095) is integrated on the S-AHB system master interface
 - Instruction cache from SSE-200 is integrated on the C-AHB code master interface
 - Data is mapped to DDR4, round trip latency for a read to DDR4 memory is 19 CPU cycles
 - Code is mapped to BRAM on FPGA
 - AHB Cache write policy is Write-Back

Only relevant part of SSE-200 is shown, AHB Cache is integrated on data path

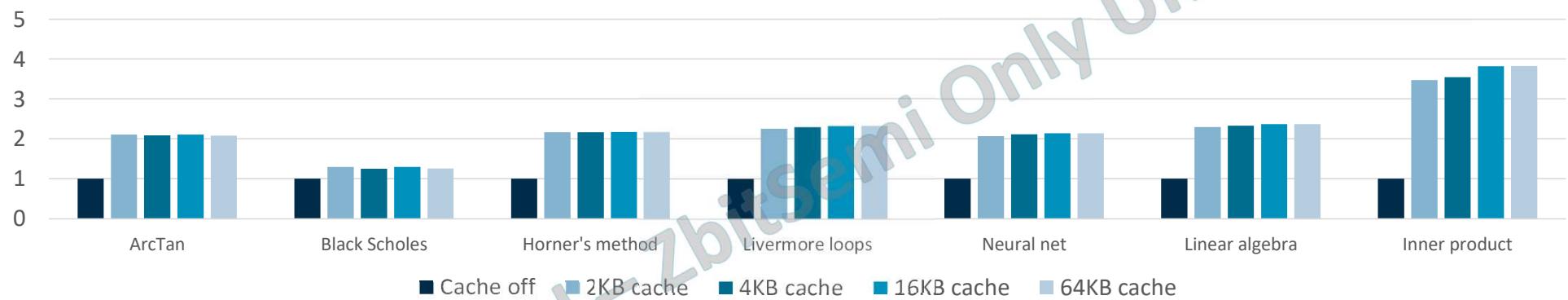
SSE-200 MPS3 FPGA: <https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/download-fpga-images>

SSE-200 Instruction Cache is only available as part of Corstone-200-201: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.101104_0200_00_en/fdc1490790871768.html

Cortex-M33 & DDR4 Performance Uplift with AHB Cache (2)

Round trip latency for a read to DDR4 memory is 19 CPU cycles

Performance of Cortex-M33 with FPU, FPMark single precision small data set



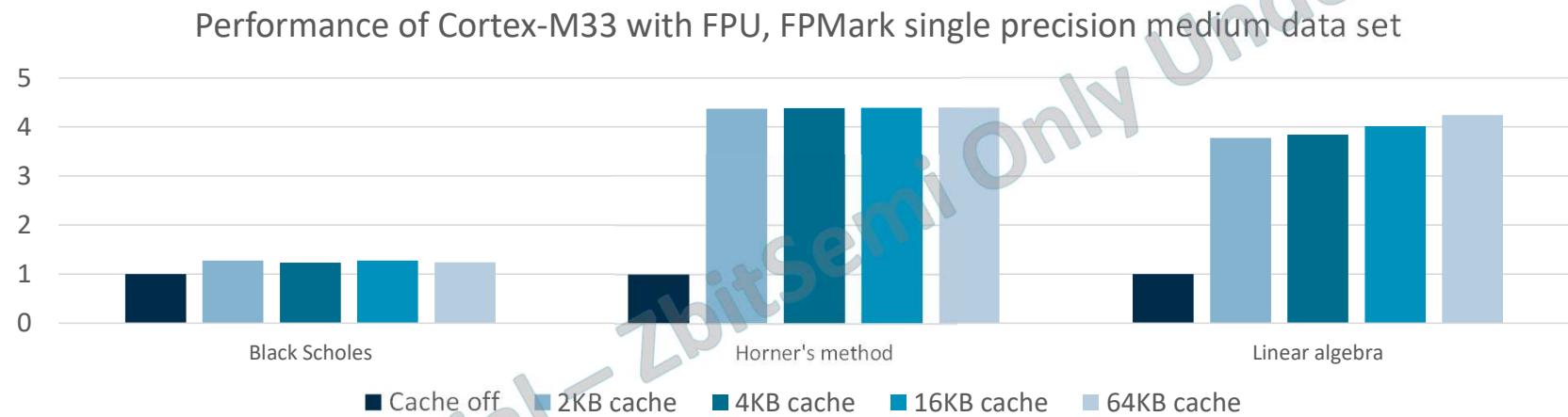
X-axis shows workloads from the [FPMark suite](#) - single precision versions with [small data set](#).

Y-axis is the performance of each configuration setting relative to the performance of the 'cache off' setting.

Cortex-M33 [Floating Point Unit \(FPU\) is included](#). When cache is on, write policy is [Write-Back](#).

Cortex-M33 & DDR4 Performance Uplift with AHB Cache (3)

Round trip latency for a read to DDR4 memory is 19 CPU cycles



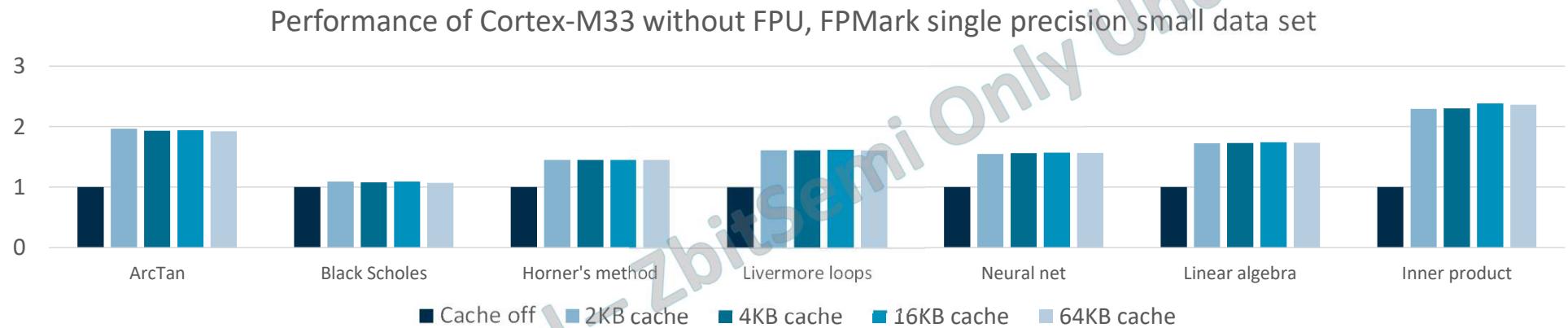
X-axis shows select workloads from the [FPMark suite](#) - single precision with **medium data set**. Due to technical issues not all workloads were run.

Y-axis is the performance of each configuration setting relative to the performance of the 'cache off' setting.

Cortex-M33 [Floating Point Unit \(FPU\)](#) is included. When cache is on, write policy is [Write-Back](#).

Cortex-M33 & DDR4 Performance Uplift with AHB Cache (4)

Round trip latency for a read to DDR4 memory is 19 CPU cycles



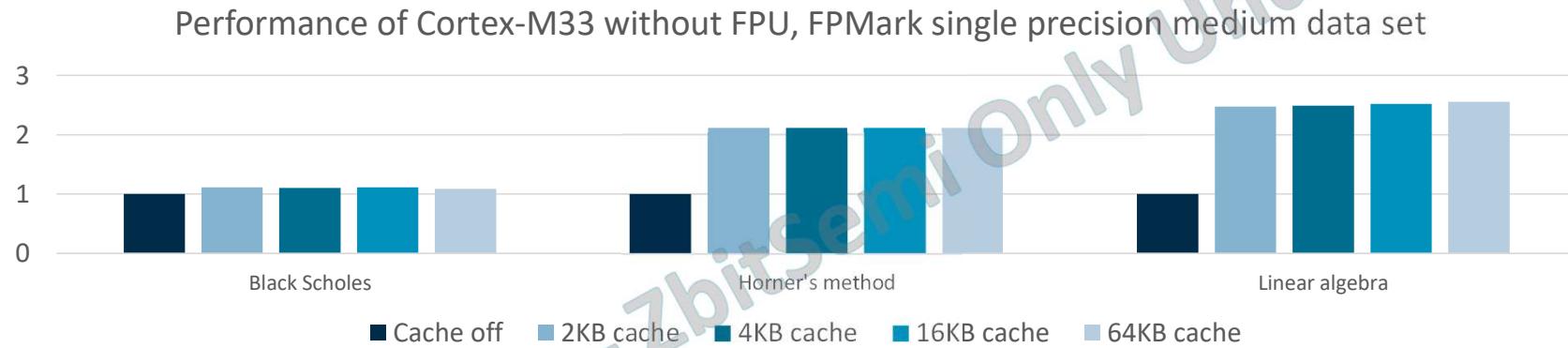
X-axis shows workloads from the [FPMark suite](#) - single precision versions with [small data set](#).

Y-axis is the performance of each configuration setting relative to the performance of the 'cache off' setting.

Cortex-M33 [Floating Point Unit \(FPU\) is not included](#) and FP operations are emulated in software. When cache is on, write policy is [Write-Back](#).

Cortex-M33 & DDR4 Performance Uplift with AHB Cache (5)

Round trip latency for a read to DDR4 memory is 19 CPU cycles



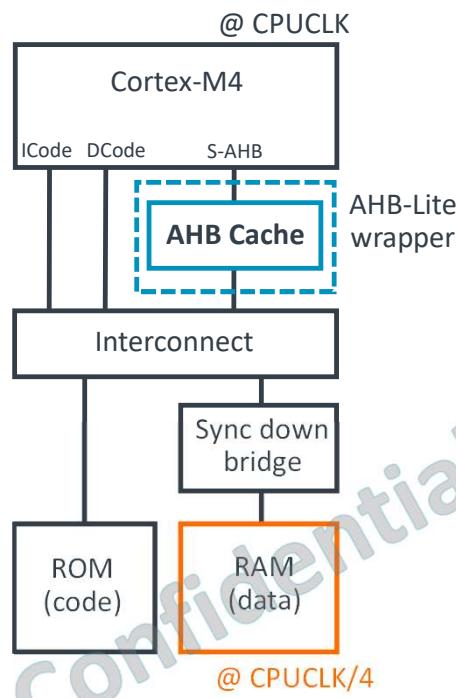
X-axis shows select workloads from the [FPMark suite](#) - single precision with [medium data set](#). Due to technical issues not all workloads were run.

Y-axis is the performance of each configuration setting relative to the performance of the 'cache off' setting.

Cortex-M33 [Floating Point Unit \(FPU\) is not included](#) and FP operations are emulated in software. When cache is on, write policy is [Write-Back](#).

Cortex-M4 & SRAM Performance Uplift with AHB Cache (1)

Cortex-M4 CPU clock to system memory clock is 1:4



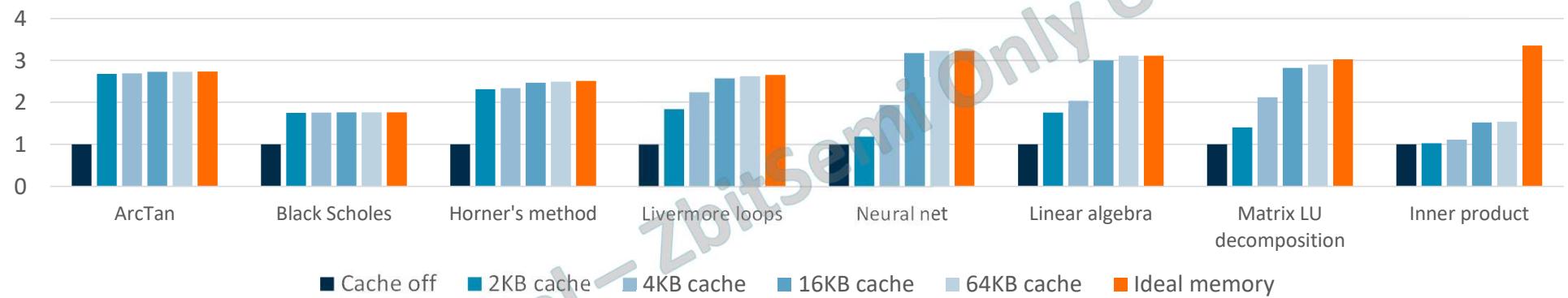
+ System setup

- Cortex-M4 is an Armv7-M processor and has three AHB-Lite master interfaces
- Code access at 1:1 clock, data access at 1:4 clock
- AHB Cache size configuration is swept
- Cortex-M4 Floating Point Unit (FPU) is enabled
- AHB Cache write policy is Write-Back

Cortex-M4 & SRAM Performance Uplift with AHB Cache (2)

Cortex-M4 CPU clock to system memory clock is 1:4

Performance of Cortex-M4 with FPU, FPMark single precision small data set, Write-Back policy



X-axis shows workloads from the [FPMark suite](#) - single precision versions with small data set.

Y-axis is the performance of each configuration setting relative to the performance of the 'cache off' setting.

When cache is on, write policy is [Write-Back](#).

Code memory accesses are zero wait state for all runs. Ideal memory means zero wait state for data.

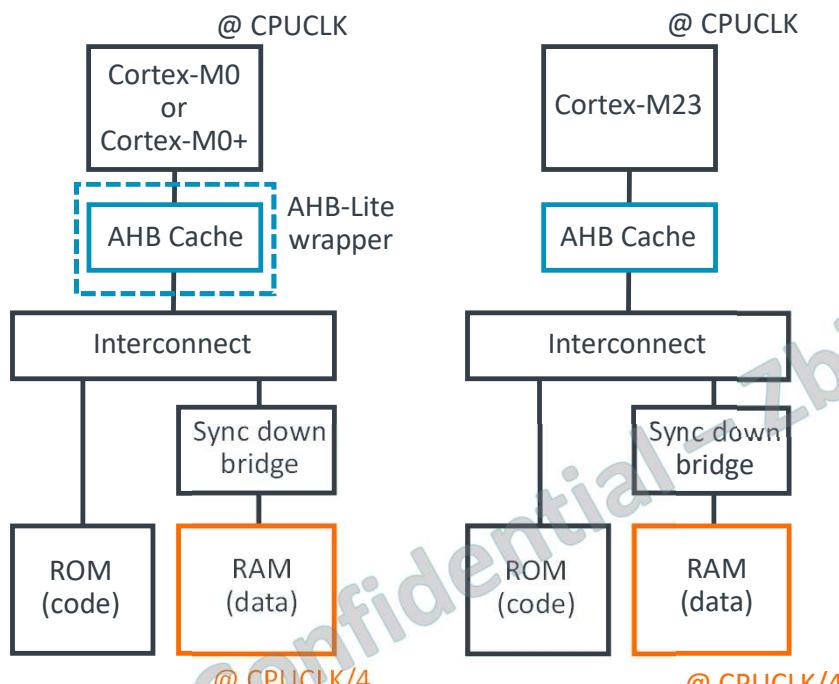
arm

CoreMark

Cortex-M0, Cortex-M0+, Cortex-M23,
Cortex-M3, Cortex-M4, Cortex-M33

Cortex-M0/M0+/M23 Performance Uplift with AHB Cache (1)

Cortex-M CPU clock to system memory clock is 1:4



+ System setup

- Cortex-M0 and Cortex-M0+ are Armv6-M processors and have single AHB master interface
- AHB-Lite wrapper handles the AHB5 <-> AHB-Lite conversion
- Cortex-M23 is an Armv8-M processor with a single AHB5 master interface
- AHB Cache write policy is Write-Back
- Code access at 1:1 clock, data access at 1:4 clock

Benchmark

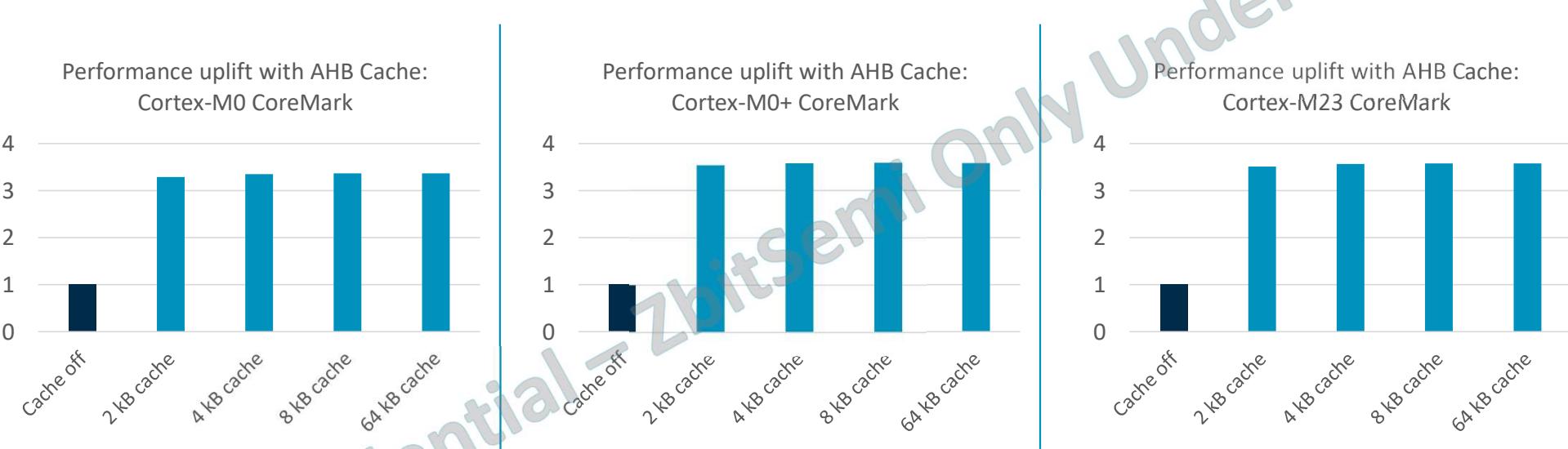
EEMBC CoreMark

Compiled with Arm Compiler 6.14 with below flags

Compiler flags: -O0 -fomit-frame-pointer -fno-common -fno-inline

Cortex-M0/M0+/M23 Performance Uplift with AHB Cache (2)

Cortex-M CPU clock to system memory clock is 1:4



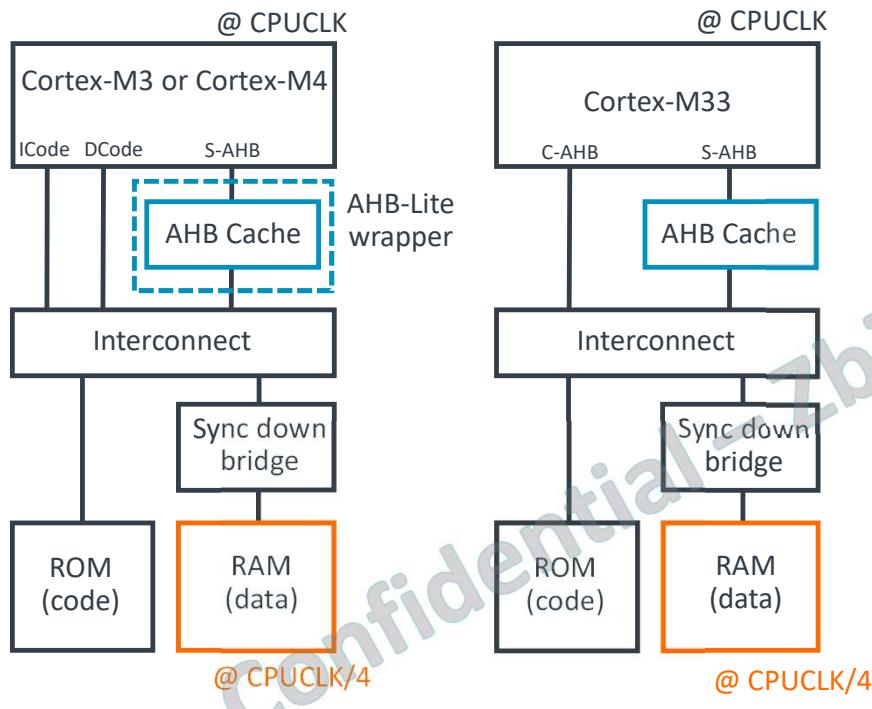
Y-axis is the **CoreMark** score for each configuration setting relative to the performance of the 'cache off' setting.

When cache is on, write policy is **Write-Back**.

Code memory accesses are zero wait state for all runs.

Cortex-M3/M4/M33 Performance Uplift with AHB Cache (1)

Cortex-M CPU clock to system memory clock is 1:4



+ System setup

- Cortex-M3 and Cortex-M4 are Armv7-M processors and have three AHB-Lite master interfaces
- AHB-Lite wrapper handles the AHB5 <-> AHB-Lite conversion
- Cortex-M33 is an Armv8-M processor and has two AHB5 master interfaces
- AHB Cache write policy is Write-Back
- Code access at 1:1 clock, data access at 1:4 clock

Benchmark

EEMBC CoreMark

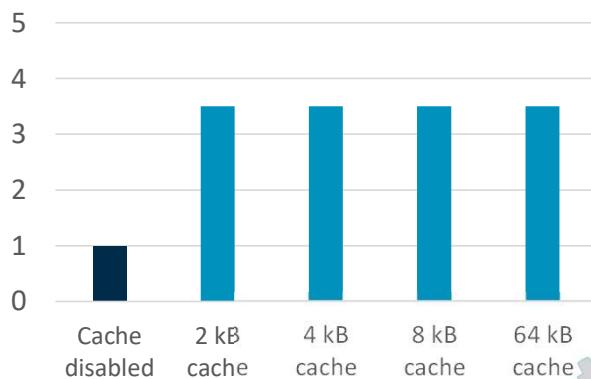
Compiled with Arm Compiler 6.14 with below flags

Compiler flags: -O0 -fomit-frame-pointer -fno-common -fno-inline

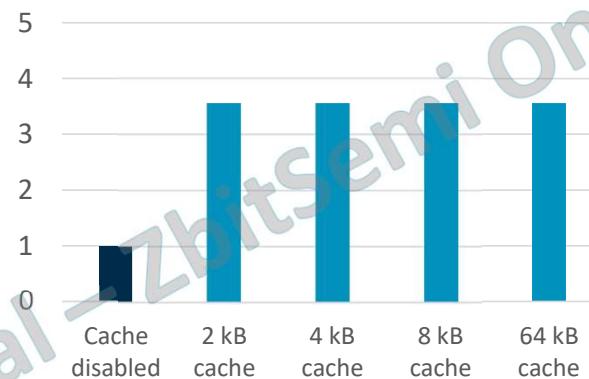
Cortex-M3/M4/M33 Performance Uplift with AHB Cache (2)

Cortex-M CPU clock to system memory clock is 1:4

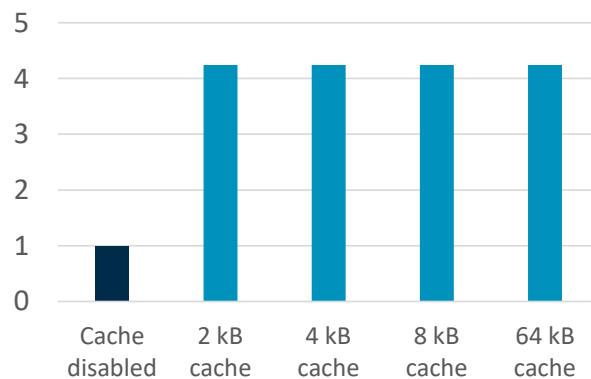
Performance uplift with AHB Cache:
Cortex-M3 CoreMark



Performance uplift with AHB Cache:
Cortex-M4 CoreMark



Performance uplift with AHB Cache:
Cortex-M33 CoreMark



Y-axis is the **CoreMark** score for each configuration setting relative to the performance of the 'cache off' setting.

When cache is on, write policy is **Write-Back**.

Code memory accesses are zero wait state for all runs.



Implementation PPA data

(CG095)

Confidential – ZbitSemi Only Under NDA

CG095 AHB Cache – Frequency and Area at 40ULP

TSMC 40 ULP, C40, 9 track library with SVTULP

PPA Measure		Config1-typ	Config2-min	Config3-max
Frequency	reg2reg	214 MHz	209 MHz	200 MHz
	reg2out	170 MHz	173 MHz	173 MHz
	in2reg	178 MHz	179 MHz	159 MHz
	in2out	167 MHz	173 MHz	175 MHz
Area	Cell count	11353	12704	14276
	Total area w/o physical cells	0.028 mm ²	0.030 mm ²	0.035 mm ²

█ Subgroup frequency █ Maximum frequency of overall design

Notes:

- Based on EAC release
- Without RAMs, 60% of the clock period is allocated to I/O constraints in line with ARM standards
- Configurations:
 - Config1-typ: 16 KB size, HxUSER_WIDTH=0, HMASTER_WIDTH=4
 - Config2-min: 2 KB size, HxUSER_WIDTH=0, HMASTER_WIDTH=4
 - Config3-max: 64 KB size, HxUSER_WIDTH=64, HMASTER_WIDTH=8, XOM, SNAPSHOTTING
- Cadence 19.x based flow



CG095 AHB Cache – Frequency and Area at 22ULL

TSMC 22 ULL, C30, 7 track library with SVT

PPA Measure		Config1-typ	Config2-min	Config3-max
Frequency	reg2reg	520 MHz	532 MHz	529 MHz
	reg2out	452 MHz	450 MHz	442 MHz
	in2reg	473 MHz	480 MHz	472 MHz
	in2out	455 MHz	452 MHz	444 MHz
Area	Cell count	12847	12397	15421
	Total area w/o physical cells	0.013 mm ²	0.013 mm ²	0.017 mm ²

█ Subgroup frequency █ Maximum frequency of overall design

Notes:

- Based on EAC release
- Without RAMs, 60% of the clock period is allocated to I/O constraints in line with ARM standards
- Configurations:
 - Config1-typ: 16 KB size, HxUSER_WIDTH=0, HMASTER_WIDTH=4
 - Config2-min: 2 KB size, HxUSER_WIDTH=0, HMASTER_WIDTH=4
 - Config3-max: 64 KB size, HxUSER_WIDTH=64, HMASTER_WIDTH=8, XOM, SNAPSHOTTING
- Cadence 19.x based flow



CG095 AHB Cache – Floorplan

Performance	40ULP	22ULL
Frequency Conditions	SSG/0.99V/0C Target 200 MHz	SSG/0.72V/0C Target 500MHz
Post-Route Timing	184.7 MHz	431.4 MHz
Post-STA Timing	163.9 MHz	434.4 MHz
Floor Plan	40ULP	22ULL
Floorplan Width x Height (mm)	0.27412 x 0.22274	0.17976 x 0.1463
Total Floor Plan Area (mm²)	0.061057	0.026299
Cell count	11353	12847
Cell Area (mm²)	0.028471842	0.013388270
Utilisation	0.5	0.5
Layers	1p7m_4x1z1u_utalrdl	1p8m_5x2r_ut-alrdl
Process, Library and implementation flow	40ULP	22ULL
Process	TSMC 40 ULP, C40, 9 track library with SVTULP	TSMC 22 ULL, C30, 7 track library with SVT
Cell Library	Arm Artisan SC9MC High Density Multi Channel Standard Cell Library r2p0-02eac0 PDK version: rel_164	Arm Artisan SC7MC Ultra High Density Standard Cell Base Library r1p0-00eac0 PDK version: rel_045
Memory Library	N/A	N/A
Vt usage	SVTULPC40	SVTC30



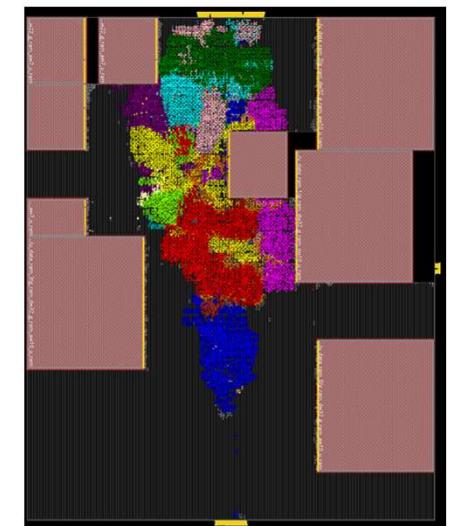
Notes:

- EAC quality
- Without RAMs, 60% of the clock period is allocated to I/O constraints in line with ARM standards
- Typical configuration: 16 KB size, HxUSER_WIDTH=0, HMASTER_WIDTH=4, no XOM, no SNAPSHOTTING
- Cadence 19.x based flow

CG095 AHB Cache – Power

- The cache power test is designed to estimate maximum power, activating interfaces and logic in parallel
 - Performs 10000 back-to-back cacheable AHB transfers with hit rate above 90%. 60% of transfers are reads and 40% are writes.
 - Performs read transfers on the APB interface
 - Requests quiescence on clock and power Q-Channels (which is denied by the cache)

Results of the cache power test	40ULP slow	22ULL slow
Frequency Conditions	TT/ 1.1V/85C Target 150 MHz	TT/0.8V/85C Target 400 MHz
Average total power incl. example RAMs	3.60 mW	2.62 mW
Average leakage power incl. example RAMs	0.053 mW	0.78 mW
Process, Library and implementation flow	40ULP	22ULL
Process	TSMC 40 ULP, C40, 9 track library with SVTULP	TSMC 22 ULL, C30, 7 track library with SVT
Cell Library	Arm Artisan SC9MC High Density Multi Channel Standard Cell Library r2p0-02eac0 PDK version: rel_164	Arm Artisan SC7MC Ultra High Density Standard Cell Base Library r1p0-00eac0 PDK version: rel_045
Memory Library	Arm Artisan Cln40ULP rf_sp_uplve_mvt/r3p0-02eac0	Arm Artisan Cln22ULL rf_sp_hde_svt_mvt/r0p0-00eac0
Vt usage	SVTULPC40	SVTC30



Notes:

- Based on EAC release
- With RAMs
- Typical configuration: 16 KB size, HxUSER_WIDTH=0, HMASTER_WIDTH=4, XOM=OFF, SNAPSHOTTING=OFF
- Cadence 19.x based flow

arm

CG095 AHB Cache - Implementation Area Breakdown

TSMC CLN40ULP, 9 TRACK, SVTULP, C40			
BLOCK	Area, μm^2	Gates/cells	% of total logic*
FB	5053	1979	18
APBIF	3286	1329	12
GWB	3134	994	9
PWB	2735	1126	10
EWB	2695	1015	9
MIU	2304	859	8
SIU	2515	1000	9
TAGIF	1854	786	7
EVCTRL	1444	756	7
DATAIF	1656	672	6
LPI	298	154	1
DIRTYIF	182	94	1
RGEN	79	30	0
MISC.	407	303	3
TOTAL LOGIC*	27642	11097	100
RAM WRAPPER	N/A	728	N/A
RAMs	86290	N/A	N/A
TOTAL DESIGN **	114224	11926	N/A

*Excluding RAM wrapper logic

**Including RAMs + other misc. logic





Value Proposition of CoreLink AHB Cache

(CG095)

CoreLink AHB5 Cache for Cortex-M systems (CG095)



Improve performance



Reduce power



Security support

- Get more efficient embedded systems
 - Enable faster systems with larger memory
 - Integrate even in secure embedded system
 - Reduce frequency of large memories to save power
- Improve memory access time for embedded processors
 - Versatile use as CPU or system cache to cache data and/or code
 - Support for AHB5 and TrustZone for Armv8-M
- AHB5 data and code cache
 - Usable with Cortex-M0/M0+/M3/M4/M23/M33
 - Support for Write-Through and Write-Back
 - Compatible with AHB and AHB5 (including new features)



Status: Released



Comparison of CoreLink AHB Cache with other Arm Caches

Confidential © 2022 Arm

Arm Caches covered in this section

Cache Controllers Web Page: <https://developer.arm.com/ip-products/system-ip/system-controllers/cache-controllers>

- CoreLink AHB Cache (CG095)
 - AHB Cache TRM:
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.101807_0000_02_en/index.html
- CoreLink AHB Flash Cache (CG092)
 - AHB Flash Cache TRM:
http://infocenter.arm.com/help/topic/com.arm.doc.ddi0569b/DDI0569B_corelink_cg092_flash_cache_trm.pdf
- SSE-200 Instruction Cache
 - SSE-200 Instruction Cache is only available as part of Corstone-200/-201, subsection in SSE-200 TRM:
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.101104_0200_00_en/fdc1490790871_768.html

AHB Flash Cache versus AHB Cache (1)

Flash Cache TRM: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0569b/DDI0569B_corelink_cg092_flash_cache_trm.pdf

	CG092 AHB flash cache	CG095 AHB Cache
Use-case in a system	<p>Not a generic data cache:</p> <ul style="list-style-type: none">• Write operation does not update the data in the cache and does not invalidate the cache either.• If the flash memory has been updated, a cache invalidate operation must be executed by software.	<p>AHB Cache can be configured as either a system, generic, or data cache. It can be used for both code and data.</p> <ul style="list-style-type: none">• It has configurable write-through and writeback policies.• It supports SW cache maintenance operations. <p>Writeback allows the downstream memory system to be powered down in a phase of many cache hits and be woken up only when there is a cache miss or eviction, thus potentially saving power.</p> <p>Write-through allows the cache to be powered down without loss of data as the downstream system memory will have the most up-to-date data.</p>

AHB Flash Cache versus AHB Cache (2)

Flash Cache TRM: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0569b/DDI0569B_corelink_cg092_flash_cache_trm.pdf

	CG092 AHB flash cache	CG095 AHB Cache
Master and slave interfaces	128-bit data interface for the master and 32-bit data interface for the slave	Supports 32-bit master and slave interfaces, allowing to “insert” as L1 cache or a shared system cache making it versatile
Support for TrustZone	No support for TrustZone (a TrustZone filter can be connected before the flash cache)	Has support for TrustZone for Arm v8-M
Associativity	2 way associative or direct mapped (one way)	4 way set associativity, enabling potentially higher hit rates, for e.g. when there are multiple streams
Line size	4 words per line (16B cache line size)	8 words per line (32B line size, allowing higher hit rates)
Power control	Limited power control	Q-Channel interface for power control – ON and OFF power states are supported

AHB Flash Cache versus AHB Cache (3)

Flash Cache TRM: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0569b/DDI0569B_corelink_cg092_flash_cache_trm.pdf

	CG092 AHB flash cache	CG095 AHB Cache
Performance monitoring	Optional support for <i>configurable hit and miss counters</i> .	<i>Hit and miss counters</i> separate for secure and non-secure, thus 4 counters. ‘Snapshot’ functionality copies the counters to separate registers in a single cycle which can be read by SW. This solves the issue of slightly different values caused by delay in reading over APB.
Pre-defined Cacheable region	Yes, Configurable flash address bus size (based on flash memory size) so that tag memory size can be minimized	No
Prefetcher	Yes, runtime configurable next-line prefetcher	No prefetcher

SSE-200 Instruction Cache versus AHB Cache (1)

Note: SSE-200 Instruction Cache is only available as part of Corstone-200/-201, TRM: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.101104_0200_00_en/fdc1490790871768.html

	SSE-200 instruction cache	CG095 AHB Cache
Data width (bits)	32	32
Address width (bits)	32	32
Cache size (kB)	0.5 to 16	2 to 64
Organization	2-way set associative, 16-byte cache line	4-way set associative, 32-byte cache line
Replacement Policy	Pseudo-random	Pseudo-random
TrustZone support	Yes, always AHB5 security related signals that are present on the cache interfaces. Read of secure data by non-secure access not allowed (it will count as a miss)	Yes, always
AHB Slave Access Type	SINGLE and INCR only	All access types including locked and exclusive access
AHB Master Access Type	INCR4/WRAP4 for line fill. All others are same as AHB Slave	INCR8/WRAP8 for line fills and writebacks. All others are same as AHB Slave

SSE-200 Instruction Cache versus AHB Cache (2)

Note: SSE-200 Instruction Cache is only available as part of Corstone-200/-201, TRM: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.101104_0200_00_en/fdc1490790871768.html

	SSE-200 instruction cache	CG095 AHB Cache
Access latency on hit	Single-Cycle, configurable register slicing Yes, but bypasses the cache.	Single-Cycle (no wait state) Write-Through or Write-Back, with configurable forced Write-Through support. Write buffer and eviction buffer functionality to improve overall cache performance.
Write Access support	Optional invalidate line on write lookup match which adds 1 cycle latency for write access.	Optional
Execute Only Memory (XOM)	Optional	
Performance monitoring	<i>Hit counter, miss counter</i> (any word of the line that is not present will count as a miss, subsequent will count as hit) and uncached counter (accesses outside cacheable region and accesses with attribute set as non-cacheable).	<i>Hit and miss counters</i> separate for secure and non-secure, thus 4 counters. ‘Snapshot’ functionality copies the counters to separate registers in a single cycle which can be read by SW. This solves the issue of slightly different values caused by delay in reading over APB.

SSE-200 Instruction Cache versus AHB Cache (3)

Note: SSE-200 Instruction Cache is only available as part of Corstone-200/-201, TRM: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.101104_0200_00_en/fdc1490790871768.html

	SSE-200 instruction cache	CG095 AHB Cache
Pre-defined Cacheable region	Yes, via RTL parameter. Cacheable region can be defined, default is 512 MB. Less number of bits in the address means save space for TAG RAM addresses and as a result optimise for low area.	No
Micro-DMA functionality	Yes. User can set start address and size and the micro-DMA will fetch the block. SW can select to lock or not lock the fetches lines. SW support available to invalidate unlocked lines.	No



Thank You

Danke

Gracias

Grazie

謝謝

ありがとう

Asante

Merci

감사합니다

ধন্যবাদ

Kiitos

شکرًا

ধন্যবাদ

תודה



Thank You

Danke

Gracias

Grazie

謝謝

ありがとう

Asante

Merci

감사합니다

ধন্যবাদ

Kiitos

شکرًا

ধন্যবাদ

תודה