ARM Cortex-A57 MPCore

r0p1-00lac0

Release Note

ARM-EPM-021309 7.0
Confidential

ARM Cortex-A57 MPCore Release Note

Proprietary notice

Document confidentiality status

Product status

The information in this document is for a product in development and is not final.

Web address

http://www.arm.com

Feedback

ARM limited welcomes feedback on both the product, and the documentation.

Feedback on this document

If you have any comments on about this document, please send email to errata@arm.com giving:

- The document title
- The document's number
- The page number(s) to which your comments refer
- A concise explanation of your comments

General suggestion for additions and improvements are also welcome.

Support and Maintenance on the ARM Cortex-A57 MPCore

Please contact support-cores@arm.com regarding any issues with the installation, content or use of this release and a member of the ARM Product Support Group will log your query in the support database and respond as soon as possible. Note that Support for this release of the product is only provided by ARM to a recipient who has a current support and maintenance contract for the product.

Change history

Issue	Date	Release note part version
1.0	14-Dec-2011	r0p0-00alp0
1.1	30-Apr-2012	r0p0-01alp0
2.0	07-Jun-2012	r0p0-02alp0
3.1	15-Sep-2012	r0p0-00bet0
3.2	03-Dec-2012	r0p0-01bet0
5.0	11-Mar-2013	r0p0-06bet0
6.0	28-May-2013	r0p0-00lac0
7.0	27-Sep-2013	r0p1-00lac0

CONTENTS

1 PRODUCT DELIVERABLES

1.1 Product Release Status

This is a Limited Access release (LAC) of the Cortex-A57 MPCore r0p1.

'Limited Access' release status has a particular meaning to ARM of which the recipient must be aware. A deliverable so designated has been extensively validated by ARM although that validation activity is not yet complete. The supply of a product at Limited Access status is intended to allow a limited number of licensees to contribute to the final stages of validation of the deliverables either through simulation or testchip. As a result the recipient of these deliverables is expected to work with the ARM design team sharing validation results and also should not take products containing the IP to volume production until they have been subsequently released at Early Access status or above.

1.2 About the ARM Cortex-A57 MPCore

Cortex-A57 is a fully-synthesizable high-performance ARMv8-A architecture compliant multiprocessor core. The key features of Cortex-A57 MPCore are:

- ARMv8 architecture support for AArch64 and AArch32 at exception levels EL0-EL3
- ARMv8 Debug
- ETMv4 instruction trace
- GICv3 CPU interface
- Multi-processing capable; MP1, MP2, MP3 and MP4 configurations
- Configurable unified L2 cache size
- Optional CPU L1 ECC/Parity
- AMBA4 ACE or AMBA5 CHI bus architecture
- Optional ARMv8 Cryptography Extensions

1.3 ARM Part Numbers for this product

The files are delivered through ARM's IP delivery server (http://connect.arm.com). Note that for this and all subsequent MP019 releases, ARM has separated documentation deliverables into a stand-alone bundle, which must be downloaded separately. This is done so that documents may be updated and re-released separately from – and with no impact to - the main engineering bundle.

The following tables list the ARM part numbers for the individual deliverables that form the MP019-BU-50000-r0p1-00lac0 Engineering Bundle and the MP019-BU-60000-r0p1-00lac0 Document Bundle.

Please note that the Reference Implementation Flow (MP019-RM-70000) is a stand-alone deliverable and delivered outside of the two bundles listed below.

MP019-BU-50000 Engineering Bundle:

Part Number	Description	Version
MP019-DC-06003	Release Notes – PDF	r0p1-00lac0
MP019-MN-22110	Synthesizable Verilog – RTL	r0p1-00lac0

MP019-MN-70053	RTL Model Configuration Script	r0p1-00lac0
MP019-MN-70052	RTL Model Configuration Check Script	r0p1-00lac0
MP019-MN-70002	RAM Integration Test Bench	r0p1-00lac0
MP019-MN-22010	Vector Replay and Capture Test Bench	r0p1-00lac0
MP019-MN-70009	Integration Kit	r0p1-00lac0
MP019-MN-70030	IP-XACT Description	r0p1-00lac0
MP019-VE-70006	Instruction Execution Test Source	r0p1-00lac0
MP019-VE-09001	Power Indicative Test Source	r0p1-00lac0
MP019-VE-70025	Maximum Power Indicative Test Source	r0p1-00lac0
MP019-VE-70027	WFI Leakage Pwr Vectors Source	r0p1-00lac0
MP019-VE-70053	Post Layout Configuration Verify Vectors	r0p1-00lac0
TM940-MN-22100	CoreSight DAP Lite Verilog RTL	r1p1-01rel0
TM918-MN-22100	APB Access Port Verilog RTL	r0p1-01rel0
TM919-MN-22100	APBMUX for DAP Verilog RTL	r0p1-00rel0
TM945-MN-22100	Serial Wire JTAG dualmode Debug Port RTL	r0p2-00rel0

MP019-BU-60000 Document Bundle:

Part Number	Description	Version
MP019-DA-00001	TRM – Framemaker	r0p1-00lac0
MP019-DA-03001	Technical Reference Manual – PDF	r0p1-00lac0
MP019-DC-70019	Configuration and Signoff Guide – PDF	r0p1-00lac0
MP019-DC-13001	Integration Manual – PDF	r0p1-00lac0
TM940-DC-06002	CoreSight DAP Lite Release Note - PDF	r1p2-00rel0
TM940-DA-00001	CoreSight DAP Lite TRM – FrameMaker	r1p2-00rel0
TM940-DA-03001	CoreSight DAP Lite TRM - PDF	r1p2-00rel0
TM093-DC-70014	CoreSight Generic Components IG – PDF	r1p3-00rel0

NOTE: The versions described are correct for the r0p1-00lac0 Documentation Bundle and may be different in later revisions of the document bundle. Please download the latest document bundle available to ensure you have access to the most recent documentation.

2 INSTALLATION

2.1 Introduction

These installation instructions only cover the UNIX platform/operating system.

2.2 Disk space required

This installation will require about 150 Mbytes of free disk space.

2.3 Installation procedure

You will have one or more tar files of the form:

```
arm-download-XXXX.tgz
```

The installation procedure is summarized below:

1. Relocate the shipment file

Copy the tar files to the directory where they are to be installed.

2. Extract tar files

Extract the tar file contents using the UNIX GNU tar utility:

```
%> gtar -zxvf <file>.tgz
```

> **NOTE:** A version of GNU tar later than 1.13 should be used to untar the deliverables as some versions of tar have problems dealing with very long path names. To find the version of gtar being used type `gtar --version`.

This will extract the deliverables into a directory named the same as the bundle number MP019-BU-50000-r0p1-00lac0, alongside a directory named the same as the documentation bundle MP019-BU-60000-r0p1-00lac0. You will also see two files named:

`ARM_DELIVERY_<transaction_id>.txt`

`ARM_MANIFEST_<transaction_id>.txt`

where `<transaction_id>` is a transaction number unique to your download.

These contain summary information of the content of the release and checksums to verify the integrity of the data.

3. Merge deliverables

Use the following steps to arrange the deliverables into a single installation directory:

The DAPLite must be relocated within the **Cortex-A57 MPCore MP019** bundle directory. Use the following UNIX command:

```
%> cp -r MP019-BU-50000-r0p1-00lac0/coresight/logical/* MP019-BU-50000-r0p1-
00lac0/cortexa57/logical/testbench/execution_tb/verilog
```

NOTE: The UNIX command "`cp`" should be unaliased. You can do this with the command "`unalias cp`".

The user can copy the entire directory contents of `docs/` from the Documentation Bundle into an equivalent location within the Engineering Bundle.

2.4 Directory structure

Figure 2-1 shows the principal directory structure of this release created after installation and merging of the deliverables:

```
MP019-BU-50000-r0p1-00lac0/
   |
   |-- cortexa57/
   |    |
   |    |-- logical/
   |    |    |
   |    |    |-- atlas/
   |    |    |     |-- verilog/
   |    |    |
   |    |    |-- atl_*/
   |    |    |     |-- verilog/
   |    |    |
   |    |    |-- shared/
   |    |    |     |-- verilog/
   |    |    |     |-- tools/
   |    |    |
   |    |    |-- models/
   |    |    |     |-- cells/
   |    |    |     |-- rams/
   |    |    |
   |    |    |-- testbench/
   |    |    |     |-- ram_integration_tb/
   |    |    |     |-- execution_tb/
   |    |    |     |-- integration_kit_daplite/
   |    |    |     |-- integration_kit_cssoc/
   |    |    |     |-- shared/
   |    |    |
   |
   |-- ARM_Cortex-A57_MPCore_r0p1-00lac0_Release_Note.pdf

MP019-BU-60000-r0p1-00lac0/
   |
   |-- cortexa57/
   |    |
   |    |-- docs/
```

Figure 2-1: Principal directory structure after unpacking the bundle

NOTE: This is a simplified view of the installed directory structure; some directories are not shown to aid clarity. The Cortex-A57 MPCore Configuration and Sign-off Guide contains a more complete description of the directory structure.

3 DOCUMENTATION

3.1 Technical Publication Documents

The following technical publications are included in the `MP019-BU-60000-r0p1-00lac0` bundle.

1. ARM Cortex-A57 MPCore Technical Reference Manual (TRM) (ARM DDI 0488)
2. ARM Cortex-A57 MPCore Configuration and Sign-off Guide (CSG) (ARM DII 0279)
3. ARM Cortex-A57 MPCore Integration Manual (IM) (ARM DII 0280)

The Technical Reference Manual provides details for programming the Cortex-A57 MPCore processor.

The Configuration and Sign-off Guide provides details for configuring the RTL, performing memory integration, checking the configured RTL and sign-off.

The Integration Manual provides details for integrating the Cortex-A57 MPCore processor in your SoC.

3.2 Additional Documents

The following engineering documents are included in the `MP019-BU-60000-r0p1-00lac0` bundle.

1. Cortex-A57 Software Optimisation Notes (ARM-EPM-040982)
2. Cortex-A57 Implementation Defined PMU Events (ARM-EPM-042751)
3. Cortex-A57 Synchronizer Flops (ARM-EPM-022126)
4. ARM Tarmac Specification (ARM-EPM-041435)

NOTE: The Cortex-A57 Reference Implementation Flow User Guide is included in the `MP019-RM-70000-r0p1-00lac0` bundle.

3.3 Adobe PDF

Documents are supplied as “Adobe PDF” (Portable Document Format) files. These files are readable on most common computer platforms and operating systems using the appropriate file reader. A suitable file reader can be downloaded from Adobe at http://www.adobe.com. Select “Adobe Reader" and download the reader for your computer platform/operating system.

4 TOOLS

4.1 Tool Versions

This release has been developed with the following tools and libraries:

Description	Tool	Version
ARM RealView tools	RealView Compiler Tools (RVCT)	4.1.514
GNU Compiler tools	GNU Compiler Collection (GCC)	4.7.3
Simulation tools	Synopsys VCS	2011.03
	Cadence Incisive Enterprise Simulator	11.10.008
	Mentor Graphics QuestaSim	10.2
Other tools	Perl	5.8.8
	Open Verification Library (OVL)	2.2
	Google Protocol Buffers	2.4.1
	Accellera Universal Verification Methodology (UVM)	1.1a
GNU core utility	sum	5.97

The supplied execution_tb AArch64 ELF tests have been pre-compiled with ARM's release of AArch64 GCC tool chain (aarch64-toolchain-2013-01.tar), which is available to download from Connect:

http://connect.arm.com/dropzone/ARMv8-A%20Annex%20open-source%20support/

If you do not have access to this dropzone, please contact support-cores@arm.com .

You can also re-compile the tests to Linaro's GCC 4.8-2013.07 release, which is available to download from Linaro:

https://launchpad.net/linaro-toolchain-binaries/trunk/2013.07/+download/gcc-linaro-aarch64-none-elf-4.8-2013.07-1_linux.tar.xz

The Protocol Buffers are available to download from Google:

https://developers.google.com/protocol-buffers/

The Universal Verification Methodology (UVM) library is available to download from Accellera:

http://www.accellera.org

Implementation tool versions are listed in the Reference Implementation Flow User Guide.

NOTE: Other tool versions may be used, but they have not been tested by ARM.

4.2 Operating Systems

This release has been developed with the following operating systems:

Description	Tool	Version
Platform OS	Red Hat Linux	Enterprise 5

5 USAGE NOTES

For initial product usage, please refer to the Cortex-A57 MPCore Configuration and Sign-off Guide.

6 KNOWN ISSUES AND LIMITATIONS

The design includes all functionality in-line with the specification and has been assessed by ARM to be at LAC quality status. However, users should be aware of the following issues and limitations:

1. The GICv3 CPU interface has not been fully verified with any GICv3 or GICv4 Distributor component. It is recommended that the Cortex-A57 top-level input pin, GICCDISABLE, be tied to HIGH (1'b1) and an external GIC-400 or similar external interrupt controller solution is used.

See the iRM documentation for known issues and limitations with the Reference Implementation Flow.

7 DIFFERENCES FROM PREVIOUS RELEASE

This is a maintenance release of Cortex-A57 MPCore, focused on correcting errata identified in r0p0. In addition, some RTL optimizations to improve Cryptography AES and Stream copy performance are included in this release.

The GICv3 CPU interface has been further validated and has been updated with fixes and changes from the GIC Architecture Specification.

The following errata are fixed in this r0p1-00lac0 release:

ID	Cat	Summary of Erratum
806969	CatB rare	A57 may send WriteEvict or Evict transaction for cache line that is still valid when using Write-Back No-Allocate memory
808671	CatC	ETM might not output an Address packet immediately after a Trace Information packet.
808870	CatB	Unconditional VLDM instructions might cause an alignment fault even though the address is aligned
808871	CatC	Double bit ECC error on an invalid L1 data cache line can incorrectly trigger an imprecise abort
809370	CatC	Watchpoint might be incorrectly taken on first half of unaligned ld/st crossing a 64-byte-aligned boundary
809370	CatC	Watchpoint might be incorrectly taken on first half of unaligned ld/st crossing a 64-byte-aligned boundary
809371	CatB	Domain fault might be missed when executing specific AT instructions
811120	CatA rare	Back-pressure on AMBA 5 CHI TXRSP link can cause data corruption
811619	CatC	Return stack erroneous match for not taken indirect branch
811671	CatB	A57 issues multiple concurrent DVMOp(Sync) transactions on AMBI 5 CHI interconnect
811672	CatC	When a single-bit ECC error occurs in the L2, uncorrected data might be returned
812171	CatC	Erroneous ESR ISV for ARMv8-added A32/T32 load/store acquire/release exclusive instructions
812319	CatB	TLBIIPAS2IS, TLBIIPAS2LIS, TLBIIPAS2, TLBIIPAS2L instructions might not invalidate targeted IPA cache entry.
813419	CatB	TLB maintenance instructions targeting EL3 may not invalidate the targeted EL3 TLB entries
813420	CatB	A57 DCCMVA or DCCSW targeting L2 cache might cause data corruption or deadlock

Please see the most recent copy of the Errata document available from ARM (PEN and/or SDEN) for further information.

The previous release of Cortex-A57, r0p0-00lac0, included the UPF description in the iRM bundle. In this r0p1-00lac0 release, the UPF description has been moved to this directory in the MP019-BU-50000 bundle.

`MP019-BU-50000-r0p1-00lac0/cortexa57/logical/atlas/power_intent/upf`
