Juno ARM Development Platform SoC
Technical Overview

Copyright © 2014, 2015. All rights reserved.

Release Information

The following changes have been made to this book.

<table>
<thead>
<tr>
<th>Date</th>
<th>Issue</th>
<th>Confidentiality</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 July 2014</td>
<td>A</td>
<td>Non-Confidential</td>
<td>First release, for r0p0</td>
</tr>
<tr>
<td>01 May 2015</td>
<td>B</td>
<td>Non-Confidential</td>
<td>First release, for r1p0</td>
</tr>
</tbody>
</table>

Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of ARM. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED “AS IS”. ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, ARM makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to ARM’s customers is not intended to create or refer to any partnership relationship with any other company. ARM may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any signed written agreement covering this document with ARM, then the signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Words and logos marked with © or ™ are registered trademarks or trademarks of ARM Limited or its affiliates in the EU and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow ARM’s trademark usage guidelines at http://www.arm.com/about/trademark-usage-guidelines.php

Copyright © 2014, 2015. All rights reserved. ARM Limited or its affiliates.


110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20348
Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

http://www.arm.com
Contents

Juno ARM Development Platform SoC Technical Overview

Preface

About this book .......................................................................................................... vii
Feedback ..................................................................................................................... x

Chapter 1 Introduction
1.1 Purpose ................................................................................................................... 1-2
1.2 Components ............................................................................................................. 1-3
1.3 Software development ............................................................................................. 1-4
1.4 Compliance .............................................................................................................. 1-5

Chapter 2 Hardware Functional Description
2.1 Functional overview .................................................................................................. 2-2
2.2 Trusted Execution Environment (TEE) ..................................................................... 2-7
2.3 Power control and thermal management .................................................................. 2-8
2.4 ADP motherboard specification ................................................................................. 2-10

Chapter 3 Software Functional Description
3.1 About the ADP software .......................................................................................... 3-2
3.2 System Control Processor (SCP) firmware ............................................................. 3-4
3.3 Application processor (AP) firmware ....................................................................... 3-6
3.4 Linaro Engineering Build (LEB) .............................................................................. 3-10

Appendix A Hardware Components
A.1 ADP SoC ................................................................................................................ A-2
A.2 ADP motherboard .................................................................................................. A-3
Preface

This preface introduces the *Juno ARM Development Platform (ADP) SoC Technical Overview.* It contains the following sections:

- *Feedback* on page x.
About this book

This book is for the Juno ARM Development Platform (ADP) SoC. It provides a high-level overview of the ADP.

Product revision status

The rmpn identifier indicates the revision status of the product described in this book, for example, r0p0, where:

- **rm** Identifies the major revision of the product, for example, r0.
- **pn** Identifies the minor revision or modification status of the product, for example, p0.

Intended audience

This book is written for software engineers who want to work with an ARM® reference platform. It describes the high-level functionality of the ADP SoC.

Using this book

This book is organized into the following chapters:

- **Chapter 1 Introduction**
  Read this for an introduction to the ADP SoC, a description of its features, and the components that it contains.

- **Chapter 2 Hardware Functional Description**
  Read this for a description of the major hardware interfaces, the hardware components of the ADP, and how they operate.

- **Chapter 3 Software Functional Description**
  Read this for a description of the ADP software.

- **Appendix A Hardware Components**
  Read this for a detailed description of the hardware components that the ADP SoC and the motherboard contain.

- **Appendix B Revisions**
  Read this for a description of the technical changes between released issues of this book.

Glossary

The *ARM Glossary* is a list of terms used in ARM documentation, together with definitions for those terms. The *ARM Glossary* does not contain terms that are industry standard unless the ARM meaning differs from the generally accepted meaning.

Conventions

Conventions that this book can use are described in:

- **Typographical conventions.**
- **Timing diagrams.**
- **Signals on page ix.**

**Typographical conventions**

The following table describes the typographical conventions:

<table>
<thead>
<tr>
<th>Style</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>italic</em></td>
<td>Introduces special terminology, denotes cross-references, and citations.</td>
</tr>
<tr>
<td><strong>bold</strong></td>
<td>Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.</td>
</tr>
<tr>
<td>monospace</td>
<td>Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.</td>
</tr>
<tr>
<td>monospace italic</td>
<td>Denotes arguments to monospace text where the argument is to be replaced by a specific value.</td>
</tr>
<tr>
<td>monospace bold</td>
<td>Denotes language keywords when used outside example code.</td>
</tr>
<tr>
<td>&lt;and&gt;</td>
<td>Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: MRC p15, 0 &lt;Rd&gt;, &lt;Crm&gt;, &lt;Opcode_2&gt;</td>
</tr>
<tr>
<td>SMALL CAPITALS</td>
<td>Used in body text for a few terms that have specific technical meanings, that are defined in the <strong>ARM glossary</strong>. For example, IMPLEMENTATION DEFINED, UNKNOWN, and UNPREDICTABLE.</td>
</tr>
</tbody>
</table>

**Timing diagrams**

The figure named **Key to timing diagram conventions** explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

![Key to timing diagram conventions](image-url)
Timing diagrams sometimes show single-bit signals as HIGH and LOW at the same time and they look similar to the bus change shown in *Key to timing diagram conventions on page viii*. If a timing diagram shows a single-bit signal in this way then its value does not affect the accompanying description.

**Signals**

The signal conventions are:

**Signal level**

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

**Lower-case n**

At the start or end of a signal name denotes an active-LOW signal.

**Additional reading**

This section lists publications by ARM and by third parties.

See Infocenter, [http://infocenter.arm.com](http://infocenter.arm.com) for access to ARM documentation.

See onARM, [http://www.onarm.com](http://www.onarm.com) for embedded software development resources including the *Cortex® Microcontroller Software Interface Standard* (CMSIS).

**ARM publications**

This book contains information that is specific to this product. See the following documents for other relevant information:

Feedback

ARM welcomes feedback on this product and its documentation.

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to errata@arm.com. Give:

- The title.
- The number, ARM DTO 0038B.
- The page numbers to which your comments apply.
- A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

ARM tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.
Chapter 1
Introduction

This chapter introduces the Juno ARM® Development Platform (ADP) SoC. It contains the following section:

- **Purpose** on page 1-2.
- **Components** on page 1-3.
- **Software development** on page 1-4.
- **Compliance** on page 1-5.
1.1 Purpose

The ADP SoC is a development platform for:

- ARMv8 AArch64 and AArch32 compute.
- big.LITTLE Multi-Processing (MP).
- Graphics Processing Unit (GPU) compute.
- 3D Graphics.
- Control and management of:
  - Security.
  - Power.
  - Thermal.

The ADP SoC enables you to develop software and tooling for ARMv8 AArch64 and AArch32.
1.2 Components

The ADP SoC consists of the following:

- A standalone development motherboard, ARM® Versatile™ Express Juno Development Platform V2M-Juno r1, instantiating the ADP SoC fabricated in TSMC28HPM.

- A Software Development Kit (SDK) that supports each hardware platform and contains the following:
  - ADP AArch64 firmware with standardized Application Programming Interfaces (APIs).
  - AArch64 Linux kernel with big.LITTLE MP support.
  - AArch32 and AArch64 Linux user space example.

- DS-5 tool support.

The processor clusters contain the following fully coherent processor clusters:

- Dual core Cortex-A57 processor cluster.
- Quad core Cortex-A53 processor cluster.

The GPU cluster contains an I/O-coherent Mali™-T624 series GPU.

The compute platform also contains a Cortex-M3 System Control Processor (SCP) for power control and thermal management.

External interfaces include:

- External memory capability includes DDR3-1600 dual channel striped memories.
- USB2 and a custom SoC to Field Programmable Gate Array (FPGA) prototyping extension interface.
- A Peripheral Component Interconnect Express (PCIe) Gen2.0 four lanes, Root Port and PHY, that has both I/O coherent and non-coherent modes.
1.3 Software development

The ADP delivers heterogeneous compute to software developers including big.LITTLE64 and *General-Purpose computing on Graphics Processing Units* (GPGPU) compute, for example:

- OpenCL.
- Direct Compute.
- OpenGL-ES.
- Direct-X graphics.
- *Trusted Execution Environment* (TEE) integration.
- Advanced power control and thermal management.
1.4 Compliance

The ADP conforms to ARM Platform Design Documents (PDDs) and white paper guidance documents, in particular, the following:

- Trusted Base System Architecture 1 (ARM DEN 0007).
- Trusted Board Boot Requirements (ARM DEN 0006).
- Power State Coordination Interface (ARM DEN 0022).
- Principles of ARM® Memory Maps White Paper (ARM DEN 0001).

This document provides a technical overview of the ADP design and associated software deliverables for the platform.
Chapter 2
Hardware Functional Description

This chapter describes the functionality of the Juno ARM® Development Platform (ADP) SoC hardware.

It contains the following sections:
- Functional overview on page 2-2.
- Trusted Execution Environment (TEE) on page 2-7.
- Power control and thermal management on page 2-8.
- ADP motherboard specification on page 2-10.
2.1 Functional overview

Figure 2-1 shows a block diagram of the ADP SoC.

The main components are as follows:
- Cortex-A57 processor cluster subsystem on page 2-3.
- Cortex-A53 processor cluster subsystem on page 2-3.
- Graphics subsystem on page 2-3.
- System Control Processor (SCP) subsystem on page 2-4.
- Interconnects and on-system memory on page 2-4.
- System Memory Management Unit (SMMU) on page 2-4.
- Generic Interrupt Controller (GIC) on page 2-4.
- Memory subsystem on page 2-5.
- Peripheral Component Interconnect Express (PCIe) on page 2-5.
- USB 2.0 Enhanced Host Controller Interface (EHCI) on page 2-5.
- Direct Memory Access (DMA) on page 2-5.
- High Definition Liquid Crystal Display (HDLCD) controllers on page 2-5.
- Static Memory Controller (SMC) on page 2-6.
- Universal Asynchronous Receiver-Transmitter (UART) on page 2-6.
- I2C on page 2-6.
- Thin Links SoC to FPGA prototyping interface on page 2-6.
- CoreSight™ on page 2-6.
2.1.1 Cortex-A57 processor cluster subsystem

The Cortex-A57 processor subsystem consists of the following:

- ARMv8 dual core Cortex-A57 cluster that is configured with an AMBA4 ACE interface and 2MB L2 cache.
- CoreLink™ ADB-400 asynchronous bridge to enable Dynamic Voltage and Frequency Scaling (DVFS).

2.1.2 Cortex-A53 processor cluster subsystem

The Cortex-A53 processor subsystem consists of the following:

- ARMv8 quad core Cortex-A53 cluster that is configured with an AMBA4 ACE interface and 1MB L2 cache.
- CoreLink ADB-400 asynchronous bridge to enable DFVS.

2.1.3 Graphics subsystem

The Mali™-T624 Graphics Processing Unit (GPU) is a high-performance hardware accelerator for 2D and 3D graphics. The GPU subsystem consists of the following:

- Mali-T624 containing:
  - Four shader cores.
  - A hierarchical tiler.
  - A Power-Management Unit (PMU).
- A job manager that distributes workloads to the four shader cores.
- A Memory-Management Unit (MMU) that performs address translation of data reads and writes from components in the system.
- A CoreLink ADB-400 asynchronous bridge to enable DFVS.

The Mali-T624 series GPU is configured with 128KB L2 RAM and an AMBA4 ACE-Lite interface.

The GPU and its associated software are compatible with the following graphics standards:

- OpenGL ES 1.1 and 2.0.
- OpenCL 1.1 full profile.
- EGL 1.4.
- Renderscript compute.
- DirectX 11 feature level 9_1, 9_3 through DX9 DDI, including Direct3D.
- DirectX 11 full-feature through DX10/11 DDI:
  - Direct3D.
  - DirectCompute.
2.1.4 System Control Processor (SCP) subsystem

A Cortex-M3 processor controls and manages the SoC. The SCP:
• Controls clocks and resets.
• Is responsible for power state transitions for the power regions in the SoC.
• Has direct control over the Power-Management Integrated Circuit (PMIC) on the board.

The Operating System (OS) can send power-management commands to the SCP using a hardware Message Handling Unit (MHU).

2.1.5 Interconnects and on-system memory

The ADP contains a CoreLink CCI-400 Cache Coherent Interconnect that provides:
• Full coherency between the processor clusters.
• I/O coherency between the GPU and processor clusters.

A 128-bit I/O coherent slave and master interface is extended from the coherent interconnect using CoreLink NIC-400 Network Interconnect components and connected to peripherals such as Direct Memory Access (DMA) and interchip interconnect.

The following static RAMs are instantiated for the application processors:
• 128KB trusted.
• 16KB non-trusted.

The following ROMs are instantiated for the application processors:
• 128KB trusted.
• 16KB non-trusted.

The ROM code is fully committed at the time of manufacture. However, it is possible to override the internal code of both ROMs for development and debug purposes.

2.1.6 System Memory Management Unit (SMMU)

Individual CoreLink MMU-401 or CoreLink MMU-400 memory management components connect all non-processor masters such as the following, to the CCI-400 and NIC-400 interconnects:
• Graphics subsystem.
• Universal Serial Bus (USB).
• DMA.
• Debug subsystem.

These SMMUs implement stage 2 address translations, that translates an Intermediate Physical Address (IPA) to a Physical Address (PA).

2.1.7 Generic Interrupt Controller (GIC)

The ADP interrupt controller complies with the GICv2m architecture. GICv2m includes a message-based interrupt feature that is necessary to handle Message Signaled Interrupts (MSIs).

GICv2m enables MSIs to set GICv2 Shared Peripheral Interrupts (SPIs) to pending. This provides a similar mechanism to the message-based interrupt features added in GICv3.

The ADP SoC is compliant to level 1 of the Server Base System Architecture specification. See Compliance on page 1-5.
2.1.8 Memory subsystem

The memory subsystem contains a Dynamic Memory Controller (DMC) that interfaces with external DDR memory using a dual 32-bit DDR3 PHY. The PHY supports both DDR3, and DDR3L and operates to 1600MT/s.

A CoreLink TZC-400 TrustZone Address Space Controller exists at the interface of the DMC-400 to the system. The TZC-400 enables the trusted OS to define multiple regions within the DDR memory that have different security access permissions.

2.1.9 Peripheral Component Interconnect Express (PCIe)

The ADP includes a 4-lane PCIe Root Port capable of operating at up to 5GTps per lane. This supports high-bandwidth connectivity with external peripherals such as SATA disk controllers and Gigabit Ethernet NIC. The PCIe Root Port and PHY are integrated on the chip.

2.1.10 USB 2.0 Enhanced Host Controller Interface (EHCI)

The ADP SoC instantiates a bus mastered USB 2.0 Enhanced Host Controller Interface (EHCI) host controller for attaching peripherals such as keyboard, mouse, and flash drive to the system. The EHCI operates with native OS drivers and supports the following available speeds:

- Low speed.
- Full speed.
- Hi speed.

USB 2.0 supports data rates of 480Mbps. The host controller is on the SoC. It is connected to a USB PHY on the board through the 60MHz 12-pin UTMI+ Low Pin Interface (ULPI) Single Data Rate (SDR) interface¹. The USB controller is configured to provide one OHCI controller and one EHCI controller. Debug is an optional feature in EHCI but the ADP does not support it.

The USB 2.0 controller supports 64-bit EHCI addressing capability therefore supports Large Physical Address Extension (LPAE) and 64-bit Operating Systems (OSs).

2.1.11 Direct Memory Access (DMA)

The ADP SoC includes a system DMA-330 Direct Memory Access (DMA) controller. You can use the DMA controller to transfer data:

- Within memory.
- Between memory and peripherals.

2.1.12 High Definition Liquid Crystal Display (HDLCD) controllers

The ADP SoC includes two independent High Definition Liquid Crystal Display (HDLCD) controllers. The HDLCD controllers can run from:

- An HDLCD clock generated by an on-chip PLL.
- A shared clock fed directly from outside through the input pad.

This scheme enables both displays to run at high resolution, and you can switch either or both of the displays to low-resolution mode, such as VGA, if required. The ADP includes a single HDLCD PLL and this means that it is not possible to run both displays at different high-resolution modes. The achievable frame rate of Full-High Definition (FHD) is 60fps.

An I²S controller supplies audio, and the output serves the two High Definition Multimedia Interface (HDMI) connectors that are located on the board.

---

¹. USB 2.0 Transceiver Macrocell Interface (UTMI).
2.1.13 **Static Memory Controller (SMC)**

The ADP includes an SMC-354 *Static Memory Controller* (SMC) to provide access to a 64MB off-chip NOR flash. The interface also provides access to off-chip peripherals such as *Non-Volatile* (NV) counter for anti-replay protection, secure keypad input, and *Real Time Counter* (RTC) instantiated within a *Field Programmable Gate Array* (FPGA).

2.1.14 **Universal Asynchronous Receiver-Transmitter (UART)**

The ADP SoC provides secure and non-secure *Universal Asynchronous Receiver-Transmitters* (UARTs), PL011s, for:

- Firmware logs and interactive firmware shell.
- Debugging the OS kernel.

Both serial ports operate at up to 115200Bd.

2.1.15 **I²C**

The ADP SoC includes the following I²C controllers:

1. I²C controller that is only accessible from the SCP and used for PMIC control.
2. I²C controller that is mapped in the application processor area and used for board functions such as:
   - HDMI controller configuration.
   - *Small Outline Dual In-line Memory Module* (SODIMM) discovery.
3. I²C controller that is mapped in the application processor area and used exclusively for trusted user input from a keypad.

2.1.16 **Thin Links SoC to FPGA prototyping interface**

An AXI expansion interface covers both master and slave interfaces for prototyping and driver development for external components such as GPU and I/O peripherals. The Thin Links expansion interface supports 40-bit addressing, QoS, and I/O coherency. The interface supports the following bandwidths:

**Juno r1 SoC master interface**
- Forward direction, that is, from the Juno r1 SoC to the FPGA: 68Mbps.
- Reverse direction, that is, from the FPGA to the Juno r1 SoC: 78Mbps.

**Juno r1 SoC slave interface**
- Forward direction, that is, from the FPGA to the Juno r1 SoC: 246Mbps.
- Reverse direction, that is, from the Juno r1 SoC to the FPGA: 305Mbps.

2.1.17 **CoreSight™**

ARM CoreSight™ provides debug and trace capability and includes an enhanced capability for extracting bandwidth and latency measurements from the system.
2.2 **Trusted Execution Environment (TEE)**

The ADP provides a software development environment to enable the development of a trusted OS.

Specifically, the following peripherals are instantiated:

- **Trusted entropy sources**.
- **Trusted key storage**.
- **Non-Volatile (NV) counter**.
- **Non-invasive attack prevention**.

2.2.1 **Trusted entropy sources**

Two trusted entropy sources are instantiated.

2.2.2 **Trusted key storage**

The **Trusted Board Boot Requirements** (TBBR) PDD defines a set of cryptographic keys to store in **One Time Programmable** (OTP) or eFuse memory. The ADP does not instantiate an OTP or eFuse macro. Instead, keys are tied in hardware, using registers that TIE cells drive, to a default and fixed value. This includes:

- 128-bit **Hardware Unique Key** (HUK).
- 256-bit **Endorsement Key** (EK).
- 256-bit Hash of the **Root Of Trusted Public Key** (ROTPK).

--- **Note** ---

The optional **Secret Symmetric Key** (SSK) is not implemented.

2.2.3 **Non-Volatile (NV) counter**

The **Trusted Base System Architecture** (TBSA) PDD defines a series of NV counters that retain state even when the SoC has been completely powered down. For the purposes of the development platform, the trusted and non-trusted counters are tied to the limit of increment:

- **31** For the trusted counter.
- **223** For the non-trusted counter.

The 232 state anti-replay counter exists through implementation in an external FPGA and you can access it by using a secure I²C access.

2.2.4 **Non-invasive attack prevention**

The ADP does not instantiate peripherals to prevent non-invasive attacks such as glitch or brown-out detection.
2.3 Power control and thermal management

This section contains the following subsections:
- Voltage domains.
- Power-gated regions on page 2-9.
- Sensor-based power-management on page 2-9.

2.3.1 Voltage domains

The following core voltage domains exist on the ADP SoC:

VA57 Core supply to the dual core Cortex-A57 cluster. 0.8-1.0V. Switchable on the board.

VA53 Core supply to the dual core Cortex-A53 cluster. 0.8V. Switchable on the board.

VGPU Core supply to the Mali-T624 GPU. 0.8-0.9V. Switchable on the board.

VSYSTOP Main 0.9V supply to the SoC top-level, including peripherals such as USB, top-level logic, and the CoreSight subsystem. Switchable on the board.

VAON Always on 0.9V supply to the SCP subsystem, debug access ports, and the digital side of the General Purpose Input and Output (GPIO).

The ADP contains asynchronous clock domains that fully support DVFS of the Cortex-A57 cluster, Cortex-A53 cluster, and the GPU. Figure 2-2 shows the ADP voltage domains.
2.3.2 Power-gated regions

The ADP contains the following gated power components:

- The two Cortex-A57 cores, the *Snoop Control Unit* (SCU), and the L2 can be power-gated separately.
- The four Cortex-A53 cores, the SCU, and the L2 can be power-gated separately. The *Embedded Trace Macrocell* (ETM) is not separately gated.

The GPU has no provision for individual power-gating of each shader core.

2.3.3 Sensor-based power-management

Sensor-based power-management provides feedback to the SCP and application processors to enable real-time thermal management. *Power, Voltage, Temperature* (PVT) monitors are instantiated at key locations in the ADP, such as physically close to the GPU and Cortex-A57 processor clusters. Unique SQ adjustment parameters for each device support sensor calibration.
2.4 ADP motherboard specification

The ADP motherboard includes the following hardware:

- PMIC.
- Ball Grid Array (BGA), BGA-1156 socket instantiating an ADP SoC.
- Dual HDMI Controllers.
- Two sockets for DDR3-1600, 800MHz, providing 8GB of memory.
- USB 2.0 PHY and USB hub.
- A Peripheral Component Interconnect Express (PCIe) Gen2.0 four lanes, Root Port and PHY, that has both I/O coherent and non-coherent modes.
- NOR Flash, 64MB, two banks of 32MB.
- Two RS232 UARTs.
- I/O component FPGA providing:
  - Secure I²C anti-replay counters.
  - Battery-backed Real Time Clock (RTC).
  - Secure keyboard and keypad socket.
- ARM CoreSight Interfaces, Joint Test Action Group (JTAG), and trace.

Figure 2-3 on page 2-11 shows a block diagram of the ADP board.
Figure 2-3 ADP board block diagram
Chapter 3
Software Functional Description

This chapter describes the Juno ARM® Development Platform (ADP) SoC software.

It contains the following sections:

• About the ADP software on page 3-2.
• System Control Processor (SCP) firmware on page 3-4.
• Application processor (AP) firmware on page 3-6.
• Linaro Engineering Build (LEB) on page 3-10.
3.1 About the ADP software

The ADP is supplied with low-level firmware to initialize the ADP SoC and associated board. The package also includes frameworks that enable control and communication between various layers of the software stack, including boot, power, and security.

The ADP Board Support Package (BSP) contains the following:

**System Control Processor (SCP) firmware**
Controls the system and manages power in the ADP.

**Application Processor (AP) firmware**
Manages the security and virtualization of the platform, and loads the Rich Operating System (OS). The supported Rich OS example for the ADP is a variant of Linux.

**Linaro Engineering Build (LEB)**
Shows Linux running on the ADP. This combines all the ADP software that is required to exploit and demonstrate the following ADP hardware features:
- Mali™-T624 Linux kernel driver.
- OS Power-Management (OSPM) integration.
- big.LITTLE Multi-Processing (MP) integration. big.LITTLE MP is an implementation of the Global Task Scheduling (GTS) big.LITTLE execution model.
- Trusted world software integration.
- Linux device drivers for all the ADP board I/O components.

The LEB example delivers the firmware, kernel, and other board-specific features as a single hardware pack image. A root file system, user space image contains the board-independent components and the Mali-T624 user space driver.

The BSP enables additional feature development outside the scope of the ADP deliverables, for example, Advanced Configuration and Power Interface (ACPI) or Android.

**Figure 3-1 on page 3-3** shows:
- Software provided.
- Software features.
- High-level software integration.
- Open source and closed source software.
Figure 3-1 ADP software overview
3.2 **System Control Processor (SCP) firmware**

The *System Control Processor* (SCP) manages the overall power, clock, reset, and system control of the ADP. Because of the hardware design, the SCP firmware is an inherently trusted part of the ADP software system. All the memory that the SCP uses for execution and private storage is on-chip to prevent attackers tampering with it.

3.2.1 **SCP trusted boot ROM**

The SCP trusted boot ROM is the first code to execute on the ADP after a cold reset. This code is fixed for the lifetime of the device and therefore executes minimal code to maximize robustness and reduce the risk of security vulnerabilities. It implements a ROM bypass functionality that the *System Configuration Controller* (SCC) registers configure, and this enables an image in flash memory to supersede the actual AP or SCP ROM images.

This section contains the following subsections:
- *Platform boot initialization*.
- *Boot protocol*.

**Platform boot initialization**

The SCP trusted boot ROM configures the initial state of the hardware platform, for example:
- Cores that are released from reset.
- Clocks that are available.

**Boot protocol**

This is the generic implementation of the protocol required to release the AP from reset and handshake with the AP to load the SCP trusted RAM firmware using the *Message Handling Unit* (MHU). The protocol passes execution control to the *SCP trusted RAM firmware*.

3.2.2 **SCP trusted RAM firmware**

The SCP contains an on-chip trusted RAM that it uses to execute most of its runtime code and store its private data.

This section contains the following subsections:
- *System Control and Power Interface (SCPI) on page 3-5*.
- *System management on page 3-5*.
System Control and Power Interface (SCPI)

The SCPI is the generic runtime interface to the SCP from the AP through the MHU. It includes:

- Inquiring capabilities of the system and individual devices.
- Obtaining and setting the state of the entire system and individual devices under SCP control. This includes a thermal sensor interface.
- Obtaining and setting the performance level of the processors and GPUs, that is, Digital Voltage and Frequency Scaling (DVFS).
- Watchdog services to non-trusted AP software. The AP non-trusted world does not have direct access to a hardware watchdog in the ADP hardware architecture. The SCP has access to a hardware watchdog and uses this to help implement the interface.
- Reporting fault conditions.

System management

The SCP manages the state of individual devices under its control, for example, clocks, interconnects, power supplies, and sensors. It manages the states of individual devices where applicable. These can be ON, OFF, and retention. It also manages the power domains and power gates in the system.

The SCP manages valid combinations of device states, power states, and the overall system state. It also reports device and SCP faults to the AP.
3.3 **Application processor (AP) firmware**

The AP firmware consists of the code required to boot the ADP up to the start of Rich OS execution. It contains the code required to set up the initial security environment, and load Linux from a variety of boot media.

--- **Note** ---

The ARM deliverables only contain partial support for security and virtualization, and third-party software vendors must work with the supplied software to integrate commercial trusted OSs and hypervisors.

This section contains the following subsections:
- *AP trusted boot ROM.*
- *AP trusted RAM firmware* on page 3-7.
- *Unified Extensible Firmware Interface (UEFI)* on page 3-8.
- *Test secure payload* on page 3-9.

### 3.3.1 AP trusted boot ROM

The AP contains an on-chip trusted ROM that executes the first AP code on the ADP SoC. This code is fixed for the lifetime of the device and therefore executes minimal code to maximize robustness and reduce the risk of security vulnerabilities. The SCP trusted boot ROM implements a ROM bypass functionality that the SCC registers configure, and this enables an image in flash to supersede the actual AP trusted boot ROM image.

This section contains the following subsections:
- *Boot ROM trusted board boot process.*
- *Fast boot path.*

#### Boot ROM trusted board boot process

The AP boot ROM establishes the root of trust in the trusted board boot process. It is the first-level firmware image in the boot process. It authenticates the AP trusted RAM firmware and passes execution control and the chain of trust on to that.

--- **Note** ---

The ARM trusted board boot implementation is a partial implementation of the *Trusted Board Boot Requirements (TBBR) Platform Design Document* (PDD). In particular, it does not implement the following:

- A manufacturer’s key provisioning mode.
- A manufacturer’s test mode.
- Strong cryptography algorithms. You must obtain these separately.
- Firmware encryption.

#### Fast boot path

The AP boot ROM supports an alternative fast boot path, permitting cores in a low-power state to quickly enter execution in the Rich OS or trusted OS without going through the cold board boot sequence. This minimizes the amount of time spent in firmware.
3.3.2 AP trusted RAM firmware

The AP trusted RAM firmware is stored in Non-Volatile (NV) storage. It executes in on-chip trusted RAM. Subsequent AP firmware images are also stored in NV storage.

This section contains the following subsections:

- **Power State Coordination Interface (PSCI).**
- **TrustZone initialization.**
- **Secure Monitor Framework (SMF).**
- **Main trusted board boot process on page 3-8.**

**Power State Coordination Interface (PSCI)**

The AP trusted RAM firmware defines a Static Memory Controller (SMC) interface to support Rich OS Power-Management (OSPM) in accordance with the Power State Coordination Interface (PSCI) PDD. For example, Linux CPU_idle uses this interface when it powers down individual processor cores. The AP trusted RAM firmware contains standard implementations of architectural core context save and restore operations at each exception level.

Additionally, the PSCI supports big.LITTLE MP.

**TrustZone initialization**

The AP trusted RAM firmware initializes trusted world ADP resources, for example:

- CP15.
- CoreLink TZC-400 TrustZone Address Space Controller.
- CoreLink DMC-400 Dynamic Memory Controller.
- CoreLink SMC-35x Static Memory Controller.
- CoreLink MMU-401 System Memory Management Unit (SMMU).
- CoreLink MMU-400 SMMU.

**Secure Monitor Framework (SMF)**

The Secure Monitor Framework (SMF) in the AP trusted RAM firmware handles all Secure Monitor Calls (SMCs). The framework handles the transition to and from the trusted world and distributes SMCs to the correct SMC handler, according to the ARM Recommended SMC Interface. Each SMC handler potentially has a different owner, for example:

- ARM Limited.
- Silicon partner.
- Original Design Manufacturer (ODM).
- Original Equipment Manufacturer (OEM).
- Trusted OS vendor.
Main trusted board boot process

The AP trusted RAM firmware is responsible for the majority of the trusted board boot process. It is the second-level firmware image in the boot process. It loads all third level images into the appropriate memory locations and authenticates them. The third level images include:

- The SCP trusted RAM firmware.
- The EL3 SMF.
- The trusted OS.
- *Unified Extensible Firmware Interface (UEFI).*

During boot, the AP trusted RAM firmware passes execution control to the trusted OS.

3.3.3 *Unified Extensible Firmware Interface (UEFI)*

The *Unified Extensible Firmware Interface (UEFI)* is a boot firmware specification that the UEFI forum maintains and develops. ARM is a member of the UEFI forum and contributes to the UEFI ARM bindings. ARM provides an AArch64 implementation of this specification for ADPs, based on the *EFI Development Kit 2 (EDK2)*, available from the Tianocore project in Sourceforge. See [http://sourceforge.net/projects/tianocore/](http://sourceforge.net/projects/tianocore/).

This section contains the following subsections:

- *Dynamic Flattened Device Tree (FDT) generation.*
- *Linux loader application.*
- *Boot manager example.*
- *UEFI device drivers on page 3-9.*

**Dynamic Flattened Device Tree (FDT) generation**

The UEFI implementation dynamically generates a *Flattened Device Tree (FDT)* descriptor, that is passed to Linux. The FDT describes the hardware resources of an ADP in a platform-neutral and portable way. This minimizes the amount of static, platform-specific functionality in Linux and to a lesser extent, the size of the firmware.

**Linux loader application**

The UEFI implementation provides a generic Linux loader application that minimizes the amount of vendor-specific code required to launch Linux. It supports FDT and implements a standardized interface to Linux.

**Boot manager example**

The UEFI implementation includes a boot manager example that provides a simple user interface to exercise the main UEFI functionality on the ADP.

The boot options enable you to launch:

- **Linux:**
  - From NOR flash.
  - From NOR flash with command-line arguments \(XYZ\).
  - From hard disk or USB.
  - Over the network.
- **UEFI shell.**

You can dynamically add boot options, for example, to launch Linux with alternative command-line arguments.
**UEFI device drivers**

The UEFI implementation contains the following ADP device drivers:
- Generic Timer.
- CoreLink CCI-400.
- CoreLink NIC-400.
- CoreLink DMC-400.
- USB 2.0, *Enhanced Host Controller Interface* (EHCI) only.
- CoreLink SMC-354 Static Memory Controller.
- Ethernet.

### 3.3.4 Test secure payload

A trusted OS provides runtime secure services to non-trusted world software, for example, authentication services to a Rich OS. The ARM test secure payload provides a simple example of Rich OS interaction with trusted world on an ADP. The test secure payload is not a trusted OS, and trusted OS vendors are expected to replace it. The test secure payload is packaged with the AP firmware although architecturally, it is at the same level as the Linux kernel.
3.4  **Linaro Engineering Build (LEB)**

The ARM ADP **Linaro Engineering Build (LEB)** example integrates the platform-independent Linux software with the platform-specific ADP **hardware pack** to demonstrate the capabilities of the ADP.

This section contains the following subsections:

- Linux kernel.
- Linux user space on page 3-11.

3.4.1  **Linux kernel**

The Linux kernel for the ARM ADP LEB is based on the standard AArch64 **Linaro Stable Kernel** (LSK) combined with the features that the following subsections describe.

- **big.LITTLE Multi-Processing (MP).**
- **OS Power-Management (OSPM) framework implementations.**
- **FDT awareness.**
- **Device drivers including Mali kernel driver** on page 3-11.

**big.LITTLE Multi-Processing (MP)**

The LEB makes all big, that is, Cortex-A57 processor, and LITTLE, that is, Cortex-A53 processor, cores visible to Linux at the same time, to enable the task scheduler to best decide how to use them. This is called **big.LITTLE MP**.

In addition to the big.LITTLE MP mechanics, the LEB contains optimizations to the Linux kernel to facilitate efficient operation on ADPs. This includes the integration of optional Linux frameworks to describe heterogeneous core topologies, and tuning to ensure that threads are scheduled efficiently across both big and LITTLE clusters. These optimizations are validated against key use-cases, for example, web browsing and video playback.

**OS Power-Management (OSPM) framework implementations**

In addition to big.LITTLE power-management, the LEB examples contain implementations of the following standard Linux OSPM frameworks, optimized for ADPs:

- **CPUfreq** for DVFS control.
- **CPUIdle** for managing core sleep states, including core context save and restore integration.
- Driver modifications for GPU DVFS.

--- Note ---

This includes the necessary communication with the PSCI and **System Control and Power Interface (SCPI)**.

---

**FDT awareness**

The kernel in the LEB auto-configures according to the FDT descriptor in the UEFI implementation.
Device drivers including Mali kernel driver

The LEB contains the following drivers for ADP devices:

- Mali-T624 kernel driver.
- Generic Timer.
- System Profiler (SP) for bandwidth and latency monitoring.
- CPUfreq and CPUidle drivers for power-management.
- SCPI and MHU.
- UART.
- High Definition Liquid Crystal Display (HDLCD).
- CoreLink SMC-354 Static Memory Controller.
- USB 2.0.
- Ethernet.

3.4.2 Linux user space

Linaro offer a standard OpenEmbedded AArch64 user space against which the ADP is tested.

Mali user space driver integration

The standard Linaro OpenEmbedded user space contains the Mali-T624 user space driver binary image. Because the kernel in the LEB already contains the Mali-T624 kernel driver, mounting the OpenEmbedded user space enables Mali GPU support for OpenGL-ES.
Appendix A
Hardware Components

This chapter describes the Juno ARM Development Platform (ADP) SoC hardware. It contains the following sections:

- *ADP SoC* on page A-2.
- *ADP motherboard* on page A-3.
A.1 ADP SoC

The ADP SoC includes the following hardware:

- Dual core Cortex-A57 processor cluster.
- Quad core Cortex-A53 processor cluster.
- Mali-T624 series GPU with four shader cores.
- System Control Processor (SCP) based on a Cortex-M3 processor.
- CoreLink CCI-400 Cache Coherent Interconnect.
- CoreLink NIC-400 Network Interconnect.
- CoreLink MMU-401 System Memory Management Unit.
- CoreLink MMU-400 System Memory Management Unit.
- CoreLink GIC-400 Generic Interrupt Controller.
- CoreLink DMC-400 DDR3 Dynamic Memory Controller.
- CoreLink DMA-330.
- Dual 32-bit DDR3, two × 1600MT/s.
- Dual ARM HDLCD display controllers, 1920 × 1080 at 60fps, with single I2S with four stereo channels.
- Bus mastered EHCI USB2 host controller, 480Mbps, UTMI+ Low Pin Interface (ULPI) interface to off-chip PHY1.
- CoreLink SMC-354 Static Memory Controller, 64MB NOR flash and board peripherals.
- UART, two × PL011.
- I2C, high-speed mode, 3.4Mb/s.
- PVT sensor subsystem.
- Security peripherals, RNG, NV counters, fuses, 32KHz oscillator.
- AXI master and slave SoC to FPGA interface based on TLX-400, Thin-Links technology.

1. USB 2.0 Transceiver Macrocell Interface (UTMI).
A.2 ADP motherboard

See *ADP motherboard specification* on page 2-10.
Appendix B
Revisions

This appendix describes the technical changes between released issues of this book.

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table B-1 Issue A</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Change</strong></td>
<td><strong>Location</strong></td>
<td><strong>Affects</strong></td>
</tr>
<tr>
<td>First release.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<p>| Table B-2 Differences between issue A and issue B | | |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replaced the PL352 component with the PL354 component.</td>
<td>Throughout the document.</td>
<td>r1p0</td>
</tr>
<tr>
<td>Updated references to the board, and board documentation, to Juno r1 and V2M-Juno r1.</td>
<td>Throughout the document.</td>
<td>r1p0</td>
</tr>
<tr>
<td>Added PCIe information.</td>
<td>• ADP SoC block diagram on page 2-2.</td>
<td>r1p0</td>
</tr>
<tr>
<td></td>
<td>• Peripheral Component Interconnect Express (PCIe) on page 2-5.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ADP motherboard specification on page 2-10.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ADP board block diagram on page 2-11.</td>
<td></td>
</tr>
<tr>
<td>Updated Thin Links speed values.</td>
<td>Thin Links SoC to FPGA prototyping interface on page 2-6.</td>
<td>r1p0</td>
</tr>
<tr>
<td>Updated voltage values.</td>
<td>Voltage domains on page 2-8.</td>
<td>r1p0</td>
</tr>
<tr>
<td>Removed SODIMM.</td>
<td>ADP motherboard specification on page 2-10</td>
<td>r1p0</td>
</tr>
</tbody>
</table>