Arm® Ethos™-U55 NPU
Version 3.0

Application development overview
Arm® Ethos™-U55 NPU

Application development overview

Copyright © 2020 Arm Limited or its affiliates. All rights reserved.

Release Information

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Confidentiality</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100-01</td>
<td>27 January 2020</td>
<td>Confidential</td>
<td>First release of version 1.0.</td>
</tr>
<tr>
<td>0200-02</td>
<td>27 March 2020</td>
<td>Confidential</td>
<td>First release of version 2.0.</td>
</tr>
<tr>
<td>0300-03</td>
<td>15 May 2020</td>
<td>Non-Confidential</td>
<td>First release of version 3.0.</td>
</tr>
</tbody>
</table>

Non-Confidential 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 click through or signed written agreement covering this document with Arm, then the click through or 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.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US 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/company/policies/trademarks.

Copyright © 2020 Arm Limited (or its affiliates). All rights reserved.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)
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.

Unrestricted Access is an Arm internal classification.

Product Status
The information in this document is Final, that is for a developed product.

Web Address
www.arm.com
Contents

**Arm® Ethos™-U55 NPU Application development overview**

*Preface*
- About this book .......................................................... 6
- Feedback ........................................................................... 8

**Chapter 1**

*Introduction*
- 1.1 Introduction ........................................................................................................ 1-10
- 1.2 Design configurability .................................................................................... 1-11

**Chapter 2**

*NPU software overview*
- 2.1 NPU software components ........................................................................... 2-13
- 2.2 NPU software tooling ................................................................................... 2-14
- 2.3 NPU runtime software stack ........................................................................ 2-15

**Chapter 3**

*Getting started*
- 3.1 Accessing software components ................................................................. 3-17

**Appendix A**

*Revisions*
- A.1 Revisions ........................................................................................................ Appx-A-19
This preface introduces the Arm® Ethos™-U55 NPU Application development overview.

It contains the following:

- *Feedback* on page 8.
About this book

This manual gives an overview of the flow of data between an application and the Arm® Ethos™-U55 NPU.

Intended audience

This manual is written for machine learning application developers who want to run their applications on the NPU.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction
This chapter gives an overview of the NPU and its incorporation into an embedded system.

Chapter 2 NPU software overview
This chapter gives an overview of the software components of the NPU. The chapter covers how the different components work together to send data to and from the NPU.

Chapter 3 Getting started
This chapter describes how to obtain the software components to begin Ethos-U55 NPU software development.

Appendix A Revisions
This appendix describes the technical changes between releases 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.

See the Arm® Glossary for more information.

Typographic conventions

italic
Introduces special terminology, denotes cross-references, and citations.

bold
Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

monospace
Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

monospace underlined
Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

monospace italic
Denotes arguments to monospace text where the argument is to be replaced by a specific value.

monospace bold
Denotes language keywords when used outside example code.

SMALL CAPITALS
Used in body text for a few terms that have specific technical meanings, that are defined in the Arm® Glossary. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.
Additional reading

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

**Arm publications**
- Developer resources:
  

**Other publications**
- None.
Feedback

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 Arm Ethos-U55 NPU Application development overview.
• The number 101888_0300_03_en.
• If applicable, the page number(s) to which your comments refer.
• 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 gives an overview of the NPU and its incorporation into an embedded system.

It contains the following sections:

• 1.1 Introduction on page 1-10.
• 1.2 Design configurability on page 1-11.
1.1 Introduction

This chapter gives an overview of the Ethos™-U55 Neural Processing Unit (NPU) and describes how this is incorporated into an embedded system.

It also provides a description of the software tooling and the runtime software stack that is used by an embedded application which runs on the NPU.
1.2 Design configurability

The Ethos-U55 NPU is a small and power-efficient processor that is used to reduce both the inference time and memory requirements needed to run Machine Learning (ML) Neural Networks (NN).

The Ethos-U55 NPU is attached to a Cortex®-M series Central Processing Unit (CPU), which can be incorporated into an embedded system that contains SRAM memory (for read–write storage) and flash memory (for read-only storage).

- The SRAM memory is used for the dynamic storage of runtime data during the inference of the neural network.
- Flash memory is used for the non-volatile storage of the runtime software stack (including the User Application) and the neural network definition (including weights).

To optimize performance of the Ethos-U NPU, the read/write AXI interface should be connected to a high speed, low latency memory, such as SRAM. The read-only AXI interface should be connected to memory that is slower or less Burst efficient, for example flash or DRAM.

The NPU is configurable to meet various performance points as outlined in the following table.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Number of MACs per cycle</th>
<th>Internal RAM</th>
<th>Performance (GOP/s @500MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>256</td>
<td>48KB</td>
<td>256</td>
</tr>
<tr>
<td>128</td>
<td>128</td>
<td>24KB</td>
<td>128</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>16KB</td>
<td>64</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>16KB</td>
<td>32</td>
</tr>
</tbody>
</table>
Chapter 2
NPU software overview

This chapter gives an overview of the software components of the NPU. The chapter covers how the different components work together to send data to and from the NPU.

It contains the following sections:

- 2.1 NPU software components on page 2-13.
- 2.2 NPU software tooling on page 2-14.
- 2.3 NPU runtime software stack on page 2-15.
2.1 NPU software components

The Ethos-U55 NPU software comprises both offline software tooling and the online Cortex-M runtime software stack.

Vela software tooling is used to compile (optimize and convert) a User Application's NN model. The input model is a fully trained and quantized TensorFlow Lite for Microcontrollers (TFLμ) model. The output is an optimized model that is able to run optimally on an Ethos-U55 NPU embedded system. All software tooling is done on a desktop PC or similar device.

The runtime software stack provides all the software that will execute on the Cortex-M series CPU. This includes the User Application, which uses the TFLμ library to execute parts of the optimized model (command stream) on the NPU.

![Figure 2-1 NPU software overview](image-url)
2.2 NPU software tooling

Software tooling consists of the following optimization tools, all of which run in a PC environment.

This section contains the following subsection:

• 2.2.1 The Vela compiler on page 2-14.

2.2.1 The Vela compiler

This tool is used to compile a TFLμ model into an optimized version that can run on the Ethos-U55 NPU.

The optimized model contains TensorFlow Lite custom operators (supported operators) for those parts of the model that can be accelerated by the Ethos-U55 NPU. Parts of the model that cannot be accelerated are left unchanged and will instead run on the Cortex-M series CPU using an appropriate kernel.

Vela trials a number of different compilation strategies and applies a cost function to each one. It then chooses the optimal execution schedule for each supported operator or group of operators.

Memory optimization

The Vela compiler also performs various memory optimizations to reduce both the permanent (for example flash) and runtime (for example SRAM) memory requirements.

One such technique for permanent storage is the compression of all the weights in the model.

Another technique is cascading, which addresses the runtime memory usage. Cascading reduces the maximum memory requirement by splitting the feature maps (FM) of a group of consecutively supported operators into stripes. A stripe can be either the full or partial width of the FM. And it can be the full or partial height of the FM. Each stripe in turn is then run through all the operators in the group.

The parts of the model that can be optimized and accelerated are grouped and converted into TensorFlow Lite custom operators. The operators are then compiled into a command stream that can be executed by the Ethos-U55 NPU.

Finally, the optimized model is written out as a TFLμ model and a Performance Estimation report is generated that provides statistics, such as memory usage and inference time.

The compiler includes numerous configuration options that allow you to specify various aspects of the embedded system configuration (for example the Ethos-U55 NPU configuration, memory types, and memory sizes). There are also options to control the types of optimization that are performed during the compilation process.
2.3 NPU runtime software stack

The runtime software stack consists of components that interact with each other in specific ways.

User Application
The User Application runs the required functions and makes calls to the TensorFlow Lite for Microcontrollers (TFLμ) library when it performs an inference of the model.

TensorFlow Lite for Microcontrollers
The TFLμ framework is compiled into a C++ library that contains a copy of the optimized model along with versions of Reference and CMSIS-NN kernels. This library is then used by the User Application to perform inferences. During an inference, the model is parsed one operator at a time and the corresponding kernels are executed. The exception to this is when it encounters a TensorFlow Lite Custom operator. In this case, the library sends the operator and associated tensor data to the NPU driver instead.

Reference kernels
Contains a set of kernels for all operators in the TFLμ framework.

CMSIS-NN
The CMSIS-NN contains highly optimized and performant kernels that accelerate a subset of operators in the TFLμ framework.

NPU driver
The NPU driver handles the communication between the TFLμ framework and the Ethos-U55 NPU to process a custom operator. When the NPU has completed its processing, it signals back to the driver, which in turn informs the TFLμ library.
Chapter 3
Getting started

This chapter describes how to obtain the software components to begin Ethos-U55 NPU software development.

It contains the following section:
• 3.1 Accessing software components on page 3-17.
3.1 Accessing software components

Arm makes use of open-source components to allow you to develop the Ethos-U55 NPU software.

To access the Ethos-U55 open-source software and tools, go to https://review.mlplatform.org/plugins/gitiles/ml/ethos-u/ethos-u.
Appendix A
Revisions

This appendix describes the technical changes between releases of this book.

It contains the following section:
  • *A.1 Revisions* on page Appx-A-19.
A.1 Revisions

This appendix describes the technical changes between releases of this book.

Table A-1 First release of version 1.0

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First development release</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-2 First release of version 2.0

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarified offline and runtime handling and optimizing of models.</td>
<td>2.1 NPU software components on page 2-13</td>
<td>All</td>
</tr>
<tr>
<td>Added reference to related document providing details of operators.</td>
<td>Memory optimization on page 2-14</td>
<td>All</td>
</tr>
<tr>
<td>Changed versioning to reflect standard software releases.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-3 First release for version 3.0

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added information about the AXI interfaces.</td>
<td>1.2 Design configurability on page 1-11</td>
<td>All</td>
</tr>
<tr>
<td>Added the getting started chapter.</td>
<td>Chapter 3 Getting started on page 3-16</td>
<td>All</td>
</tr>
</tbody>
</table>