AN14474

i.MX 9 - L3 Cache Partitioning for Predictable Real-Time Performance

Rev. 1.3 — 3 November 2025 Application note

Document information

Information	Content
Keywords	AN14474, L3 Cache, L3 Cache Partitioning, Real-Time, i.MX 93, i.MX 943, i.MX 95, ARM DynamIQ Shared Unit, DSU
Abstract	This application note describes how to partition the L3 cache between the cores, using features of the Arm DynamIQ Shared Unit.

1 Introduction

In most CPU clusters, L3 cache is a shared resource, which typically gives the best overall performance of the system for a given cache size. However, this may not be ideal in real-time situations. For example, if a low-priority task is memory-intensive, it may pollute the entire L3 cache of the cluster, increasing the latency of a higher-priority task. This is undesirable in a real-time environment.

This document describes how to partition the L3 cache between the cores, using features of the Arm DynamIQ Shared Unit. Furthermore, it shows how to allocate specific tasks (high priority real-time) on cores with dedicated partitions of L3 cache. The code in this document was tested on i.MX 943, i.MX 95, and i.MX 93.

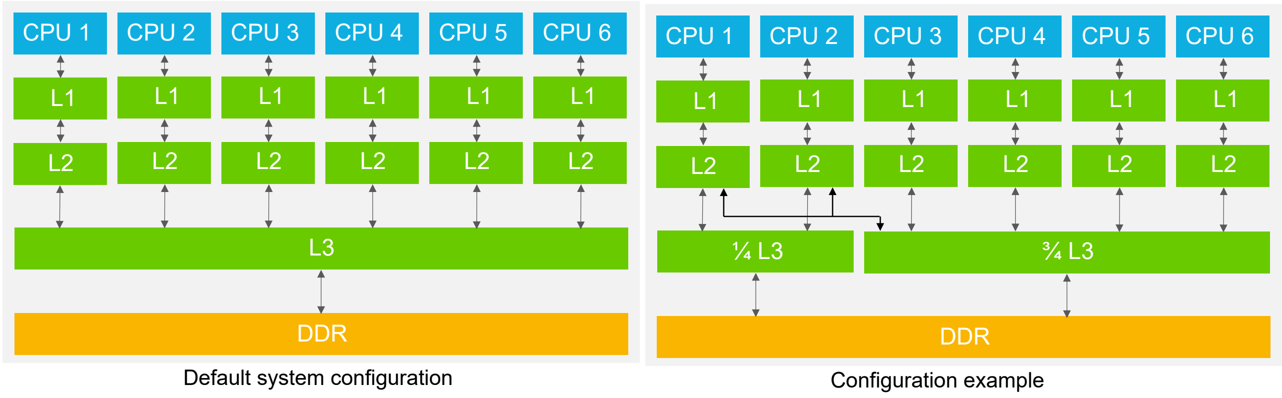

Figure 1. System configuration

As shown in Figure 1,

- the left image shows the default L3 cache configuration on i.MX 95, shared among all the cores.
- the right image shows a possible configuration in which ¼ of L3 cache is used exclusively by CPU 1 and CPU 2, while the remaining ¾ of L3 cache is shared among all six cores.

2 General approach

For a complete description of how to partition L3 cache, see Arm® DynamIQ™ Shared Unit Technical Reference Manual. In a nutshell, the L3 cache is divided in four equal way groups (a fancy name for parts), numbered 0-3. Each way group can be assigned to one or more of the eight "schemes", numbered 0-7. A scheme is simply a set of way groups. All unassigned way groups are shared among all eight schemes. Each CPU has to be allocated to one of the schemes and have access to the cache of that scheme. To implement the example in Figure 1, perform the following steps:

1. Assign way group 0 of the cache to scheme ID 1.
2. Leave way groups 1-3 of the cache unassigned (or, alternatively, assign way groups 1-3 to both schemes ID 0 and ID 1).
3. Set CPU 1-2 to use scheme ID 1.
4. Set CPU 3-6 to use scheme ID 0.

It is obligatory that all **used** schemes have exclusive access, or shared access to at least one-way group of the cache.

To implement the cache partitioning, this solution provides access from the Linux user-space to the following registers: `CLUSTERPARTCR_EL1` and `CLUSTERTHREADSID_EL1`. This provides flexibility and ease of configuration, but some protections may be required in a production system. For the complete description of the registers, see the manual referenced above. Briefly:

- `CLUSTERPARTCR_EL1` configures the allocation of the cache way groups to the schemes.
- `CLUSTERTHREADSID_EL1`, one per core, allows setting the scheme ID used by that core.

In this implementation, `CLUSTERACPSID_EL1` and `CLUSTERSTASHSID_EL1` are left on their default value (0), which means that ACP transactions and stash requests are directed to scheme ID 0. Make sure that scheme ID 0 has access to at least one way group.

This solution adds a kernel module, which exposes one read/write `sysfs` file for each of the relevant registers. Writes and reads to these files are forwarded to a patched TF-A, which is able to read/write these registers (running at Exception Level 3).

For testing, we reserve a part of the cache for the exclusive use of some of the cores. We evaluate the performance of a real-time task running on the cores with exclusive cache while the rest of the processes, run on the other cores. We use memory-intensive tasks which are strongly affected by the cache performance.

3 Implementation

The current implementation is based on the LF-6.12.20_2.0.0 BSP release. Other versions may require porting.

1. On the Linux PC, set up the Yocto environment according to Section 3, 4, and 5 in the *Real-time Edge Yocto Project User Guide* (document RTEDGEYOCTOUG).
2. Clone the `recipes-cachepartition` repository in the `meta-imx/meta-imx-bsp` directory. The `recipes-cachepartition` directory contains the following recipe appends:
   - A kernel patch implementing the kernel module for the L3 cache partitioning (in `linux-imx` subdirectory).
   - An ATF patch implementing the necessary SMC calls for setting the registers (in `imx-atf`).
   - A small tool, usecache, which can be used to test the amount of available cache (in `usecache`).

   ```
   $ cd ~/yocto-real-time-edge/sources/meta-imx/meta-imx-bsp
   $ git clone -b lf-6.12.20-2.0.0 https://github.com/nxp-imx-support/recipes-
   cachepartition
   ```
3. Add the `usecache` package to the image. `usecache` is a cache stress test, which can be used for the L3 cache partitioning validation. Add the below line in the `conf/local.conf` file.

   ```
   CORE_IMAGE_EXTRA_INSTALL += " usecache"
   ```
4. Build the `Real-time Edge` image.

   ```
   bitbake nxp-image-real-time-edge
   ```
5. Write the resulted `<image_name>.wic.zst` image located in the `tmp/deploy/images/<machine>` directory on the SD card using the following command:

   ```
   $ zstd -d <image_name>.wic.zst
   $ sudo dd if=<image_name>.wic of=/dev/sd<x> bs=1M conv=fsync
   ```

4 Testing

To test, perform the following steps:

1. Connect the USB debug port of the board to the PC using a USB cable. This action creates four virtual serial ports on the PC. Typically, the third serial port corresponds to the Linux console. Open this port in a terminal emulator using the following parameters: 115200 baud rate, 8 data bits, no parity, and 1 stop bit.
2. Boot the board.
3. Load the `cachepartition` module. The module prints various debug messages in the kernel log, which can be inspected with `dmesg`.

```
modprobe cachepartition
```

4. Go to the `/sys/kernel/cachepartition` directory. Here locate the following files: `PARTCR_EL1`, and `THREADSID_EL1_[0-5]` - one for each core.
5. You can check the current value of the registers using `cat`. Example:

```
cat partcr_el1
```

6. You can set the value of the registers using `echo`. Example:

```
echo e1 > partcr_el1
```

4.1 Examples of configuration

To configure, perform the following steps:

1. Default configuration: All the L3 caches are shared among all cores.

```
echo 0 > partcr_el1
```

2. ¾ L3 cache to cores 4-5 (Scheme ID 1), and ¼ L3 cache to cores 0-3, ACP and STASH (Scheme ID 0). The value written in the `partcr_el1` register is E1, which is 11100001 in binary. It means that way groups 3, 2, 1 are assigned to Scheme ID 1 (first 4 bits) and the way group 0 – assigned to Scheme ID 0. Then, we select Scheme ID 0 for cores 0-3 and Scheme ID 1 for cores 4-5.

```
echo e1 > partcr_el1
echo 0 > threadsid_el1_0
echo 0 > threadsid_el1_1
echo 0 > threadsid_el1_2
echo 0 > threadsid_el1_3
echo 1 > threadsid_el1_4
echo 1 > threadsid_el1_5
```

You can check that the configuration is working using the `usecache` tool.

- ```
  taskset -c 0 usecache 512 100 0
  taskset -c 4 usecache 512 100 4
  ```

  The previous commands run the same `usecache` test on CPU 0 and on CPU 4. `usecache` is a memory intensive task, and it uses, in this case, 512 K of memory. Given that CPU 4 has access to more L3 cache than CPU 0, it runs significantly faster.

- ```
  taskset -c 0 usecache 64 100 0
  taskset -c 4 usecache 64 100 4
  ```

  In this case, the memory space used by the `usecache` is 64 K, and fits in the L1-L3 cache of all CPUs, so it runs in approximately equal time.

- ```
  taskset -c 0 usecache 512 100 0 & taskset -c 4 usecache 512 100 4
  ```
  ```

In this case, we run `usecache` simultaneously on CPU 0 and CPU 4, but because they have access to separate parts of the cache, the time remains about the same as when we run separately.

- ```
  taskset -c 0 usecache 512 100 0 & taskset -c 1 usecache 512 100 1
  ```

  In this case, we run `usecache` simultaneously on CPU 0 and CPU 1, but because they share the same L3 cache partition, the time increases, compared to when run separately.

3. Cores 4-5 have access to all the L3 cache (Scheme ID 2), cores 0-3 have access only to the first ¼ of L3 cache (Scheme ID 1), ACP and STASH have access only to the second ¼ of L3 cache (Scheme ID 0). The value F84 written in the `partcr_el1` register is 111110000100, which means that Scheme ID 2 has access to all 4-way groups (the four most significant bits), Scheme ID 1 has access only to way group 3 (the middle four bits) and Scheme ID 0 has access only to way group 2 (the four least significant bits).

   ```
   echo f84 > partcr_el1
   echo 1 > threadsid_el1_0
   echo 1 > threadsid_el1_1
   echo 1 > threadsid_el1_2
   echo 1 > threadsid_el1_3
   echo 2 > threadsid_el1_4
   echo 2 > threadsid_el1_5
   ```

   Again, you can use `usecache` to test the performance in this case.

Use the `cyclictest` to test the system latency on the various cores.

When using the Jailhouse hypervisor, configure the L3 cache before enabling Jailhouse. Once Jailhouse is enabled, it restricts the communication between the kernel and the ATF. You can always disable Jailhouse temporarily, to change the L3 cache configuration.

5 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:
   ```

6 Revision history

Table 1 summarizes the revisions to this document.

Table 1. Revision history

Document ID	Release date	Description
AN14474 v.1.3	03 November 2025	• Updated Section 3 • Added i.MX 943 and updated the LF version
AN14474 v.1.2	19 February 2025	Use the GitHub repository for the `recipes-cachepartition`, instead of a SW archive
AN14474 v.1.1	28 October 2024	Added hyperlink of AN14474SW in Section 3
AN14474 v.1.0	24 October 2024	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Contents

1 Introduction 2
2 General approach 2
3 Implementation 3
4 Testing 4
4.1 Examples of configuration 4
5 Note about the source code in the document 5
6 Revision history 6
Legal information 7