

Contents

<

Back to search

<

All Armv7-A Documentation

ARM Cortex-A Series Programmer's Guide for ARMv7-A

>

Preface

>

Introduction

>

ARM Architecture and Processors

>

ARM Processor Modes and Registers

>

Introduction to Assembly Language

>

ARM/Thumb Unified Assembly Language Instructions

>

Floating-Point

>

Introducing NEON

>

Caches

▼

The Memory Management Unit

>

Virtual memory

>

The Translation Lookaside Buffer

Choice of page sizes

First level address translation

Level 2 translation tables

>

Memory attributes

▼

Multi-tasking and OS usage of translation tables

Address Space ID

Translation Table Base Register 0 and 1

The Fast Context Switch Extension

>

Memory Ordering

Version: 4.0 (Latest) ▼

Rate this page: ☆ ☆ ☆ ☆ ☆

# Address Space ID

When we described the translation table bits in *Level 2 translation tables* we noted a bit called nG (non-global). If the nG bit is set for a particular page, the page is associated with a specific application. When the MMU performs a translation, it uses both the virtual address and an ASID value.

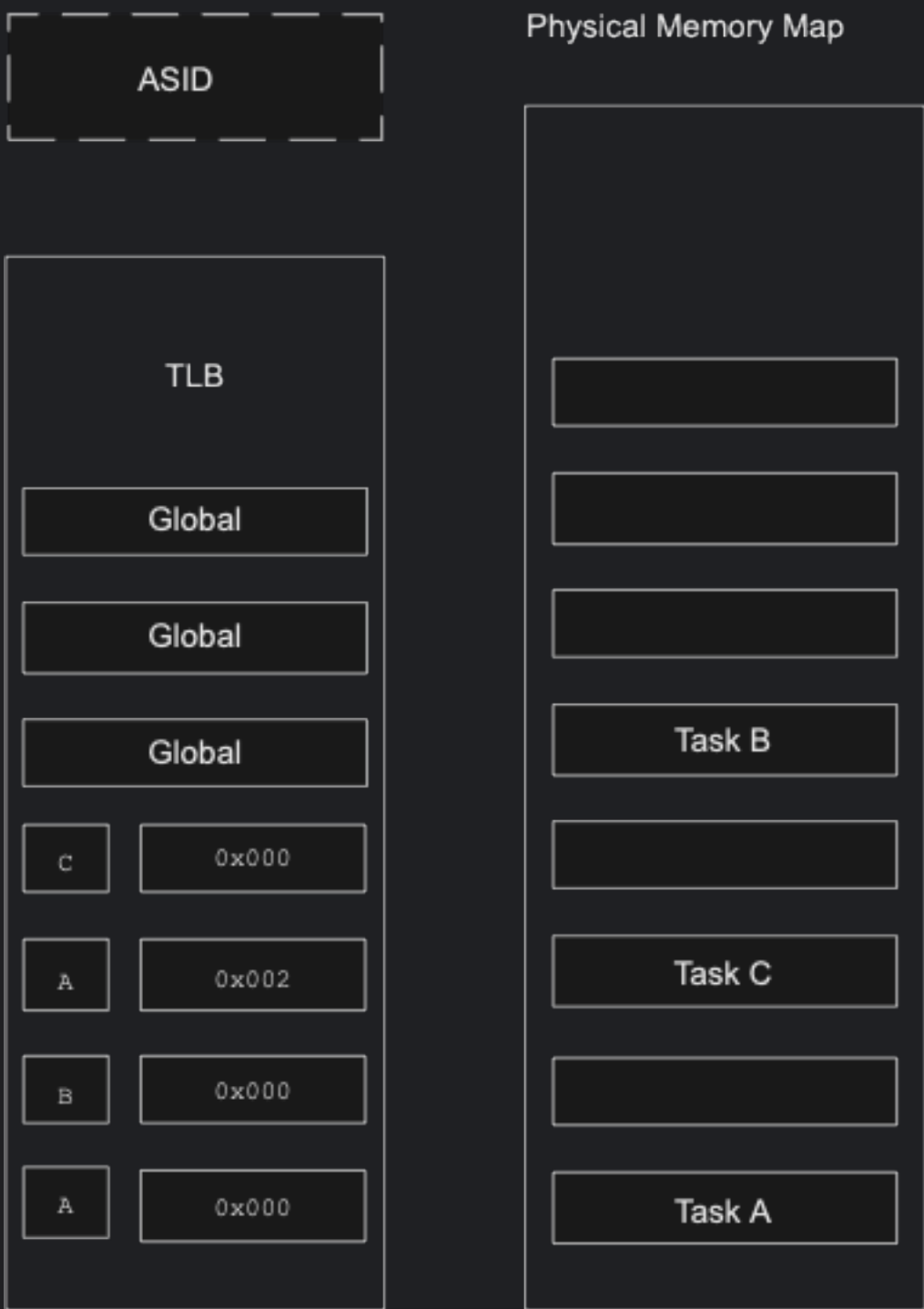
The ASID is a number assigned by the OS to each individual task. This value is in the range 0-255 and the value for the current task is written in the ASID register (accessed using CP15 c13). When the TLB is updated and the entry is marked as non-global, the ASID value will be stored in the TLB entry in addition to the normal translation information. Subsequent TLB look-ups will only match on that entry if the current ASID matches with the ASID that is stored in the entry. You can therefore have multiple valid TLB entries for a particular page (marked as non-global), but with different ASID values. This significantly reduces the software overhead of *context switches*, as it avoids the requirement to flush the on-chip TLBs. The ASID forms part of a larger (32-bit) process ID register that can be used in task-aware debugging.

Note

A *context switch* denotes the scheduler transferring execution from one process to another. This typically requires saving the current process state and restoring the state of the next process waiting to be run.

Figure 9.11 illustrates this. Here, you have multiple applications (A, B and C), each of which is linked to run from virtual address 0. Each application is located in a separate address space in physical memory. There is an ASID value associated with each application so you can have multiple entries within the TLB at any particular time, that will be valid for virtual address 0.

Figure 9.11. ASIDs in TLB mapping the same virtual address



## Developing on Arm

- AI and ML
- Android Development
- Automotive
- Cloud-to-Edge and Networking
- Gaming, Graphics and VR
- High Performance Computing
- Internet of Things
- Windows on Arm

## Architecture

- AMBA
- Compression Technologies
- CoreSight Architectures
- CPU Architecture
- GPU Architectures
- Instruction Sets
- Security Architectures
- System Architectures

## Products

- CPU Processors
- Graphics and Multimedia
- Physical IP
- System IP
- SoC Design and Simulation
- Software Development Tools

## Support

- Design Reviews and Checklists
- Training
- Documentation
- Licensing
- Downloads
- Contact Support
- Arm Security Center

## Community

- Communities
- Forums
- Blogs

## About Arm

- Leadership
- Security
- News
- Contact Us
- Arm Offices

Feedback

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Cook

Cookies Settings

Accept All Cookies

Accessibility | Subscription Center | Trademarks

arm

Copyright © 1995-2024 Arm Limited or its affiliates. All rights reserved.