HSEM internal peripheral

HSEM internal peripheral

Stable: 04.02.2020 - 15:58 / Revision: 04.02.2020 - 15:47

Contents

1 Article purpose

The purpose of this article is to briefly introduce the hardware semaphore peripheral (HSEM) and its main features.

2 Peripheral overview

The peripheral hardware spinlock is used to provide synchronization and mutual exclusion between heterogeneous processors.

2.1 Features

Refer to the STM32MP15 reference manuals for the complete list of features, and to the software components, introduced below, to see which features are implemented.

- 32 hardware semaphores are available on the platform.
- semaphores could be accessed by the Arm® Cortex®-A7 core and the Arm® Cortex®-M4

2.2 Security support

The hardware semaphores is a **non-secure** peripheral (under ETZPC control).

3 Peripheral usage and associated software

3.1 Boot time

The hardware semaphore is used at boot time for GPIO access protection between the Arm® Cortex®-A7 and Cortex®-M4 cores.

3.2 Runtime

3.2.1 Overview

The hardware spinlock can be used by:

- the Arm Cortex-A7 non-secure core to be controlled in Linux® by the hardware spinlock framework
- the Arm Cortex-M4 to be controlled in STM32Cube MPU Package by HSEM HAL driver

Notice that the Arm Cortex-A7 secure could also use the spinlock, but there is no such using yet in OpenSTLinux distribution.

3.2.2 Software frameworks

<table>
<tr><th>Do</th><th>Peri</th><th colspan="3">Software frameworks</th><th>Comment</th></tr>
<tr><td>main Cortex-A7 secure (OP-TEE)</td><td>Cortex-A7 non-secure (Linux)</td><td colspan="3">Cortex-M4

(STM32Cube)</td><td></td></tr>
<tr><td>Domain</td><td></td><td></td><td>Linux hardware spinlock framework</td><td>HSEM HAL driver</td><td></td></tr>
</table>

3.2.3 Peripheral configuration

The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in the corresponding software framework article or, for Linux in the Hardware spinlock overview article.

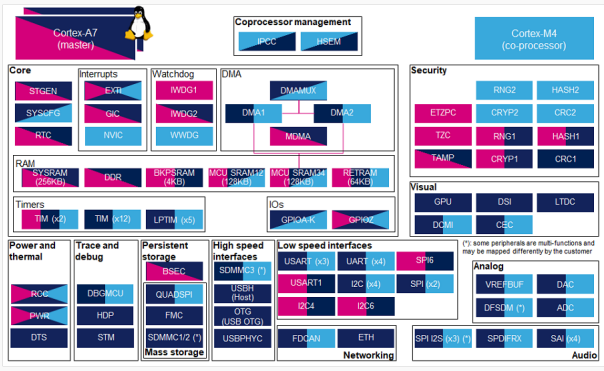
The HSEM peripheral is shared between the Cortex-A and Cortex-M contexts, so a particular attention must be paid to have a complementary configuration on both contexts.

3.2.4 Peripheral assignment

It does not make sense to allocate HSEM to a single runtime execution context, that is why it is enabled by default for both cores in the STM32CubeMX.

Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- means that the peripheral can be assigned () to the given runtime context.
- is used for system peripherals that cannot be unchecked because they are statically connected in the device.



Internal peripherals

Refer to How to assign an internal peripheral to a runtime context for more information on how to assign peripherals manually or via STM32CubeMX.
The present chapter describes STMicroelectronics recommendations or choice of implementation. Additional possiblities might be described in STM32MP15 reference manuals.

Domain	Peripheral	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Coprocessor	HSEM	HSEM				

Hardware Semaphore

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

Microprocessor Unit

Open Portable Trusted Execution Environment