Bus_Register_Protection_1 for KIT_AURIX_TC297_TFT

Register access protection

AURIX™ TC2xx Microcontroller Training
V1.0.0

Scope of work

This example shows how to protect registers from unintended write access by using AURIX™ register access protection mechanisms.

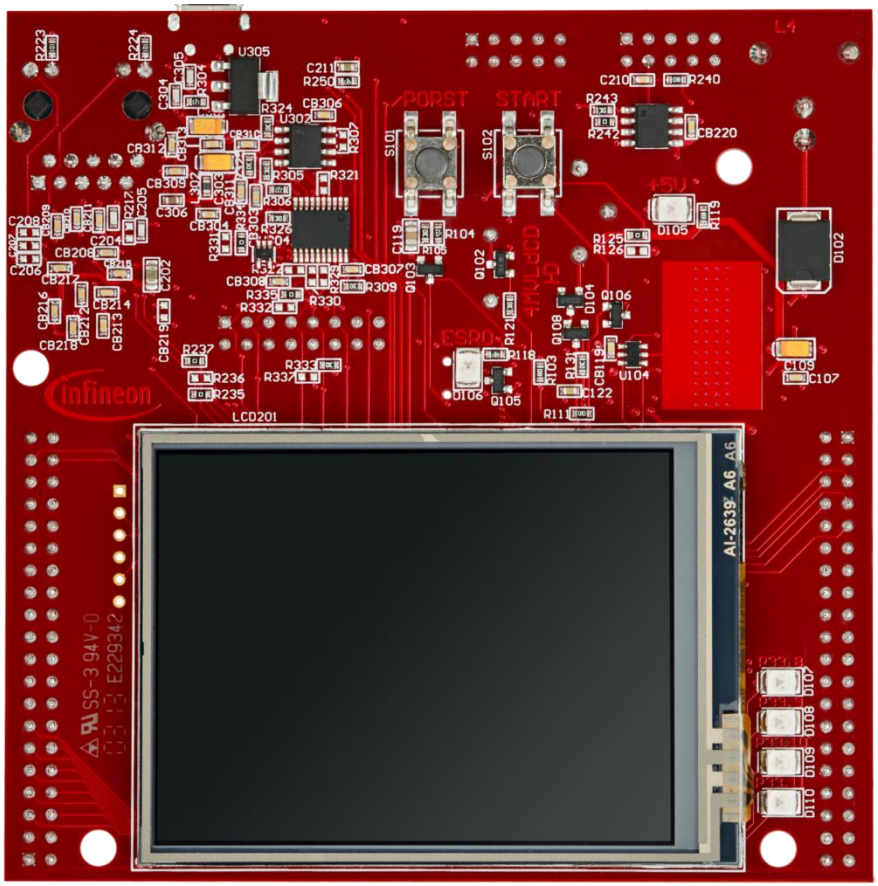
The CPU0 tries to make a write access to an ASCLIN0 module register two times: when the access is enabled for CPU0 and when the access is disabled for CPU0. During the second register write access, a Trap is expected to be raised, to inform about the illegal write access.

Introduction

- Write accesses of any master to any slave's registers can be enabled/disabled for safety reasons using the register access protection mechanism (ACCEN).
- Each master is identified via a unique TAG ID when accessing the system bus.
- Based on the masters' TAG IDs, the application can select which master is allowed to perform register write operation to a slave.
- If a master attempts to write to a protected register, then a bus error trap event is generated and the write operation is blocked.

Hardware setup

This code example has been developed for the board KIT_AURIX_TC297_TFT_BC-Step.



Implementation

Initialize the used peripherals:

- As first step, the port pins are configured to General Output Push-Pull Mode. To enable LEDs‘ usage, the following function is used:
  IfxPort_setPinModeOutput()

- Then the module **ASCLIN0** is enabled in order to demonstrate the access protection mechanism through the function:
  IfxAsclin_enableModule()

Implementation

Register access protection implementation:

- Clear Safety EndInit protection
  - Get Safety Watchdog password: ***IfxScuWdt_getSafetyWatchdogPassword()***
  - Clear Safety EndInit using the safety watchdog password ***IfxScuWdt_clearSafetyEndinit()***
- Disable all accesses of CPU0 master to slave module **ASCLIN0** by resetting the corresponding access enable bit (**ENx**) of the **ACCEN0** register as following:
  - set ***MODULE_ASCLIN0.ACCEN0.B.EN0 = 0x0;*** (0 to disable access and 1 to enable access, by default access is enabled)

  Note: *ACCEN0.B.ENx bits* are correlated to TAG IDs (EN0 corresponds to TAG ID 000000_B, ..., EN31 corresponds to TAG ID 011111_B)
  - TAG ID 000000_B corresponds to CPU0, therefore EN0 is the bitfield that has to be modified
- Restore Safety EndInit protection using the safety watchdog password ***IfxScuWdt_setSafetyEndinit()***

Implementation

Trap Service Routine implementation:

Writing to protected registers leads to a Bus Error Trap generation. For this reason, a Trap Service Routine (TSR) is needed.

- All TSRs are already implemented within the iLLD drivers and they contain a hook which enables specific user code to be added inside each TSR function, using the following steps:
  - In ***Ifx_Cfg.h*** file, the ***IFX_CFG_EXTEND_TRAP_HOOKS*** macro needs to be un-commented to enable the possibility to overwrite the default hooks.
  - Add a new file called ***Ifx_Cfg_Trap.h*** which contains the redirection of the default hook function to the implemented one:
    - Default Bus Error TSR hook ***IFX_CFG_CPU_TRAP_BE_HOOK()*** is replaced by the implemented hook ***busErrorTSRHook()***
  - The hook source code is implemented in order to verify if the register value was modified or the write access was denied. An LED indicates the execution result.

Implementation

Trap Service Routine implementation (Cont.):

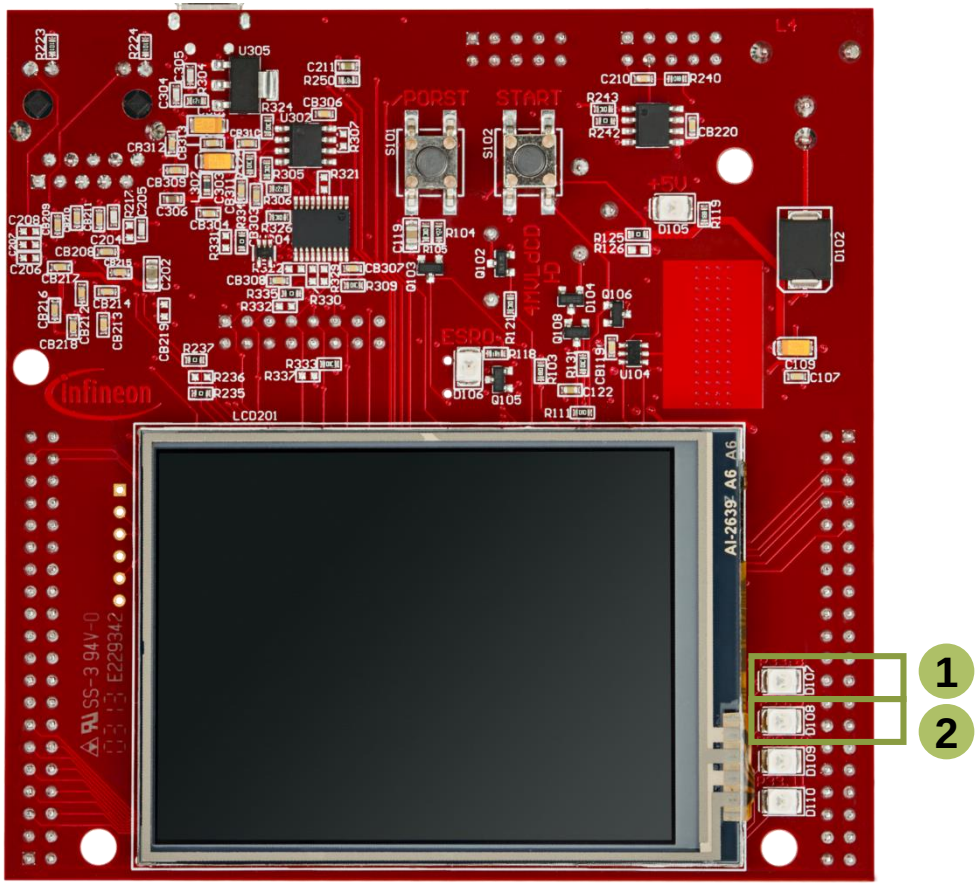
Please note that in Debug mode, the iLLD TSR stops the program execution by calling the DEBUG instruction, this can be avoided by defining the ***IFX_CFG_CPU_TRAP_DEBUG*** macro inside the ***Ifx_Cfg_Trap.h*** file.

Run and Test

After code compilation and flashing the device, observe the behavior of the LEDs.

Check that ***LED1*** (1) and ***LED2*** (2) are switched on:

- ***LED1*** switches on to indicate that the register write access was successful when the access protection is disabled
- ***LED2*** switches on to indicate that a Trap is generated and the register write access was denied when the access protection is enabled



References

- AURIX™ Development Studio is available online:
- https://www.infineon.com/aurixdevelopmentstudio
- Use the *„Import...“* function to get access to more code examples.

- More code examples can be found on the GIT repository:
- https://github.com/Infineon/AURIX_code_examples

- For additional trainings, visit our webpage:
- https://www.infineon.com/aurix-expert-training

- For questions and support, use the AURIX™ Forum:
- https://www.infineonforums.com/forums/13-Aurix-Forum

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-02
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
BUS_Register_Protection_1
_KIT_TC297_TFT

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.