Cortex M On-Chip Emulation

Contents

1 Introduction

This document describes Cortex-M specific winIDEA configuration and settings. Configuration and settings common to all: Cortex-M, Cortex-A and Cortex-R architectures are described in Cortex On-Chip Emulation section / document.

User is also encouraged to get familiar with documentation from ARM. Following are suggested documents:

- ARM®v7-M Architecture Reference Manual
- ARM®v6-M Architecture Reference Manual
- Cortex™-M0 Technical Reference Manual
- Cortex™-M0+ Technical Reference Manual
- Cortex™-M1 Technical Reference Manual
- Cortex™-M3 Technical Reference Manual
- Cortex™-M4 Technical Reference Manual
- ETMv3.5 Architecture Specification
- ARM ETM-M4 Technical Reference Manual

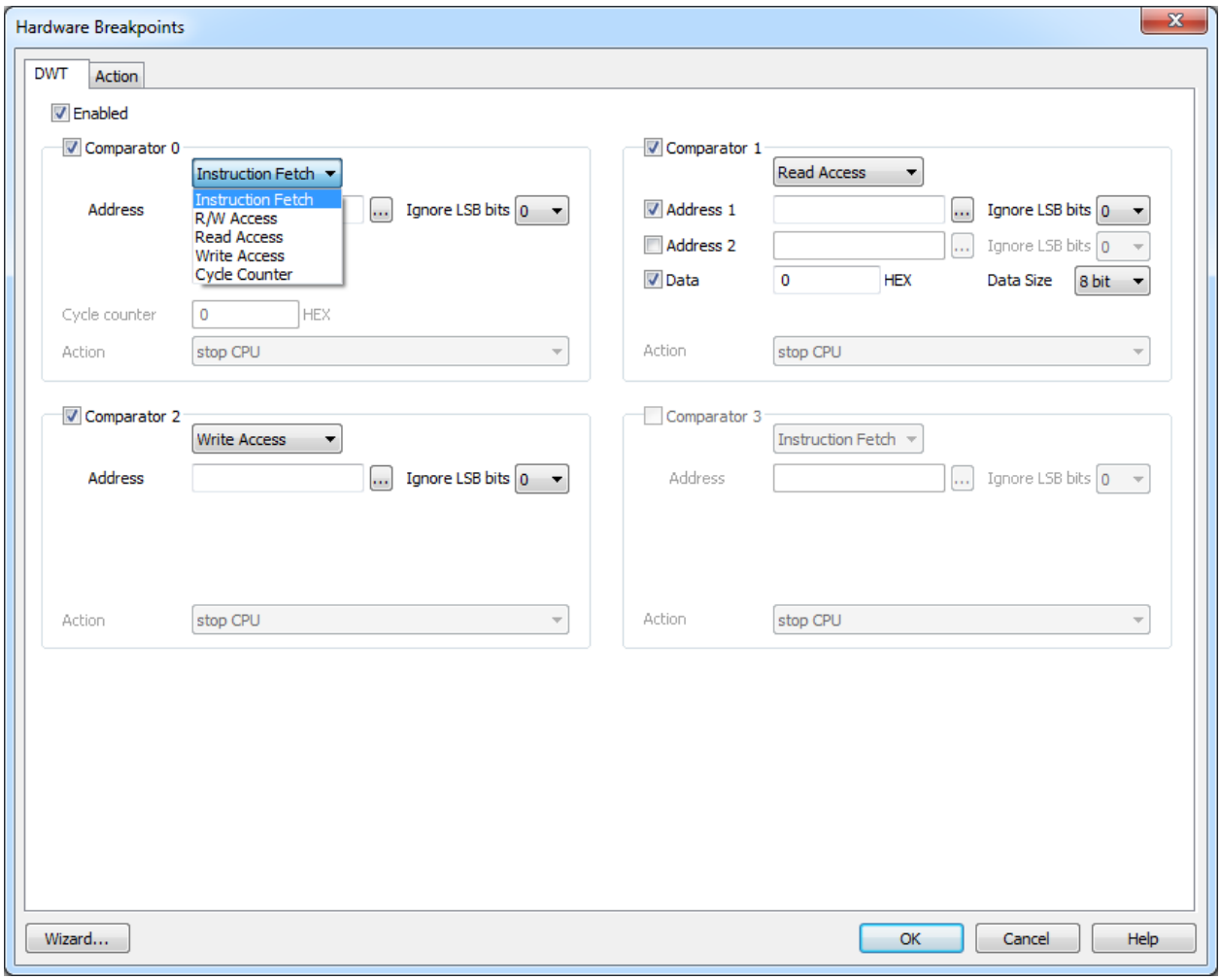
2 Access Breakpoints

Access breakpoints feature on Cortex M based MCUs is provided by the DWT (Data Watchpoint Trigger) unit. The DWT can also be used as the optional trace source unit. It provides up to 4 comparators which support instruction address, data address, data value (comparator 1 only) and CPU cycle count matching (comparator 0 only). Exact level of comparator function support depends on the DWT implementation on a particular microcontroller. For instance DWT comparators on some microcontrollers may support only address matching and no data value matching. Others may implement both address and data value matching.

winIDEA performs run time check of available debug resources (number of comparators, data value comparison, etc.) when access breakpoints are configured and activated and pops up a warning when some of configured resources are not supported by the microcontroller.

Each DWT comparator can be configured to perform instruction address or data address matching while only comparator 1 can additionally perform data value matching. Comparator 0 can also be configured to perform CPU cycle count matching. When a comparator detects a match it will take the specified action.

Note: The DWT unit is shared between access breakpoints and trace trigger. Consequentially DWT used by one debug functionality is not available for the other functionality. In practice this means that no trace trigger can be set on data access when access breakpoint is configured already and vice versa.



Cortex-M hardware breakpoints dialog

Above access breakpoints dialog is generic for Cortex-M based microcontrollers. The actual dialog window displayed for a specific M-based microcontroller might hide some of the controls depending on the actual DWT hardware implementation on the chip.

3 Trace

Please, refer to a separate document describing Cortex trace and functionalities based on it (profiler, execution coverage).

4 NXP LPC

4.1 Boot and Memory Remapping

On Cortex-M devices from LPC family a Boot Rom is present with code that is executed on reset and a memory mapping register (MEMMAP, SYSMEMREMAP)

In winIDEA version 9.12.178 a following change of handling session initialization and memory remapping was introduced.

After debug connection is established the target CPU is released from reset and stopped at beginning of boot code. Initial stack pointer and reset vector is read from flash. Breakpoint is set on initial reset vector from flash. At this point CPU is ran and boot is executed. When CPU hits the breakpoint it is stopped. If the flash does not contain valid reset vector CPU is stopped with delay. Flash is then mapped to address 0x00000000 and initial stack pointer and reset vector are loaded to R13 and R15 registers respectively. User finds the CPU in this state.

Initialization script can be used if mapping RAM to beginning of memory space is desired after reset.

4.2 LPC17xx Startup

The flash boot loader code is executed every time the part is powered on or reset. The loader can execute the ISP command handler or the user application code. A LOW level after reset at pin P2.10 is considered an external hardware request to start the ISP command handler. Assuming that power supply pins are on their nominal levels when the rising edge on RESET pin is generated, it may take up to 3 ms before P2.10 is sampled and the decision on whether to continue with user code or ISP handler is made. If P2.10 is sampled low and the watchdog overflow flag is set, the external hardware request to start the ISP command handler is ignored. If there is no request for the ISP command handler execution (P2.10 is sampled HIGH after reset), a search is made for a valid user program. If a valid user program is found then the execution control is transferred to it. If a valid user program is not found, the auto-baud routine is invoked.

Pin P2.10 is used as a hardware request signal for ISP and therefore requires special attention. Since P2.10 is in high impedance mode after reset, it is important that the user provides external hardware (a pull-up resistor or other device) to put the pin in a defined state. Otherwise unintended entry into ISP mode may occur.

When ISP mode is entered after a power on reset, the IRC and PLL are used to generate the CCLK of 14.748 MHz.

Criterion for valid user code

The reserved Cortex-M exception vector location 7 (offset 0x 001C in the vector table) should contain the 2’s complement of the check-sum of table entries 0 through 6. This causes the checksum of the first 8 table entries to be 0. The boot loader code checksums the first 8 locations in sector 0 of the flash. If the result is 0, then execution control is transferred to the user code.

If the signature is not valid, the auto-baud routine synchronizes with the host via serial port 0.

4.1 LPC11A02/04 Debug Pin Remapping in Boot

For WLCSP packages only: The boot loader changes the default pin configuration from reset values to the following pin functions:

• PIO0_2 register: default pin function configured as SWCLK.

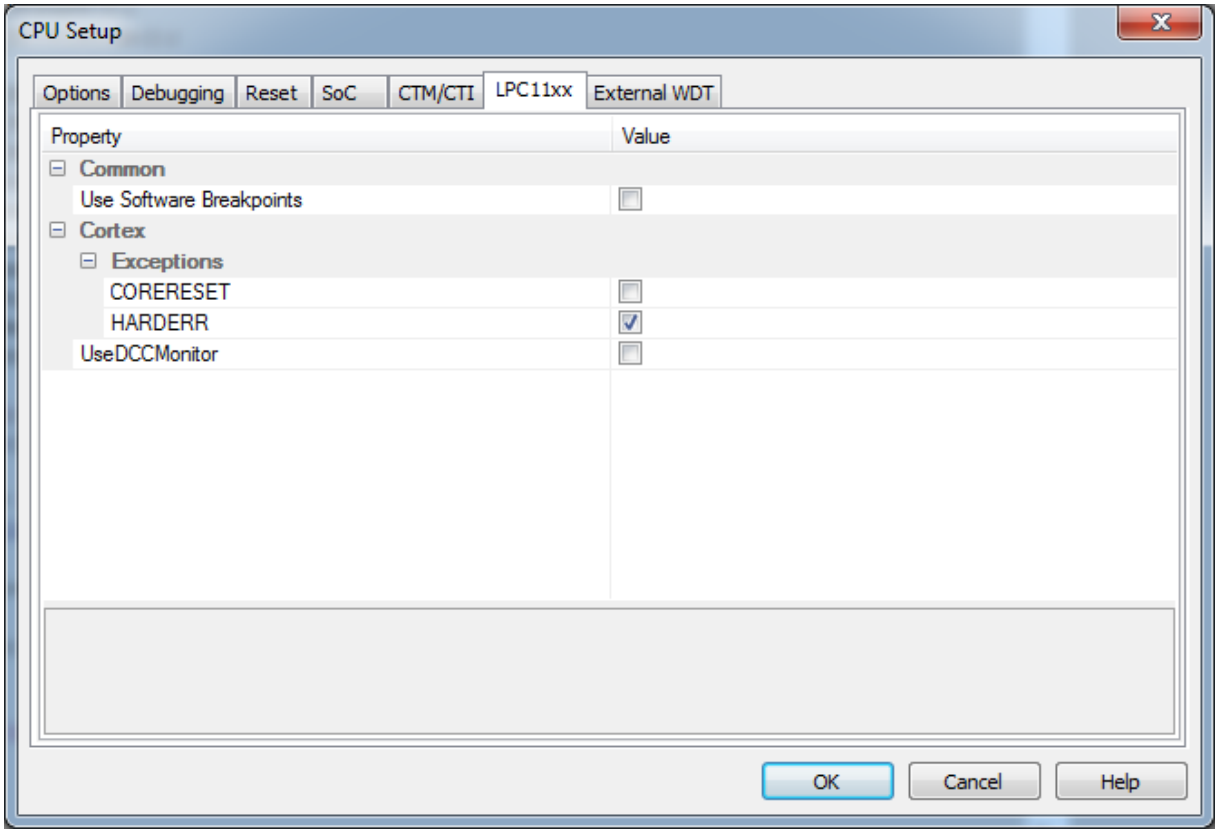
• PIO0_3 register: default pin function configured as SWDIO.

• TCK_PIO0_5 register: default pin function configured as PIO0_5.

• SWDIO_PIO0_10 register: default pin function configured as PIO0_10.

PIO0_2 and PIO0_3 should be connected to debug connector and used for debugging.

User must not enable CORERESET exception catch in winIDEA or the CPU will be stopped before boot remaps debug pins and debug session will not be initialized successfully.



LPC11xx specific options in CPU Options

4.2 LPC13xx Startup

The flash boot loader code is executed every time the part is powered on or reset. The loader can execute the ISP command handler or the user application code, or in case of LPC13xx it can obtain the boot image as an attached MSC device through USB. A LOW level during reset at pin PIO0_1 is considered an external hardware request to start the ISP command handler or the USB device enumeration without checking for a valid user code first. The state of PIO0_3 determines whether the UART or USB interface will be used (refer to CPU user manual for more details).

Assuming that power supply pins are on their nominal levels when the rising edge on RESET pin is generated, it may take up to 3 ms before PIO0_1 is sampled and the decision on whether to continue with user code or ISP handler/USB is made. If PIO0_1 is sampled low and the watchdog overflow flag is set, the external hardware request to start the ISP command handler is ignored. If there is no request for the ISP command handler execution (PIO0_1 is sampled HIGH after reset), a search is made for a valid user program. If a valid user program is found then the execution control is transferred to it. If a valid user program is not found, the auto-baud routine is invoked.

Pin PIO0_1 is used as a hardware request signal for ISP UART/USB and requires special attention. Since PIO0_1 is in high impedance mode after reset, it is important that the user provides external hardware (a pull-up resistor or other device) to put the pin in a defined state. Otherwise unintended entry into ISP mode may occur.

Note: The sampling of pin PIO0_1 can be disabled through programming flash location 0x0000 02FC

Criterion for valid user code

The reserved Cortex-M exception vector location 7 (offset 0x 001C in the vector table) should contain the 2's complement of the check-sum of table entries 0 through 6. This causes the checksum of the first 8 table entries to be 0. The boot loader code checksums the first 8 locations in sector 0 of the flash. If the result is 0, then execution control is transferred to the user code.

If the signature is not valid, the auto-baud routine synchronizes with the host via serial port 0 or boots from the USB port (PIO0_3 is sampled high).

Watchdog

Regular reset cannot be used with LPC devices. For more information about reset modes see Reset section. This results in short core run before it is stopped. In that time the watchdog might get enabled. It remains counting in stop mode and it is only disabled on external reset or overflow reset. Another external reset cannot be used – it would cause endless loop. Instead the watchdog status is checked after each reset. If watchdog is enabled the debug session is paused until the overflow reset is detected. Since the core is set to halt on reset the debug initialization can continue.

Note that NXP advises the use of the interrupt watchdog mode while debugging.

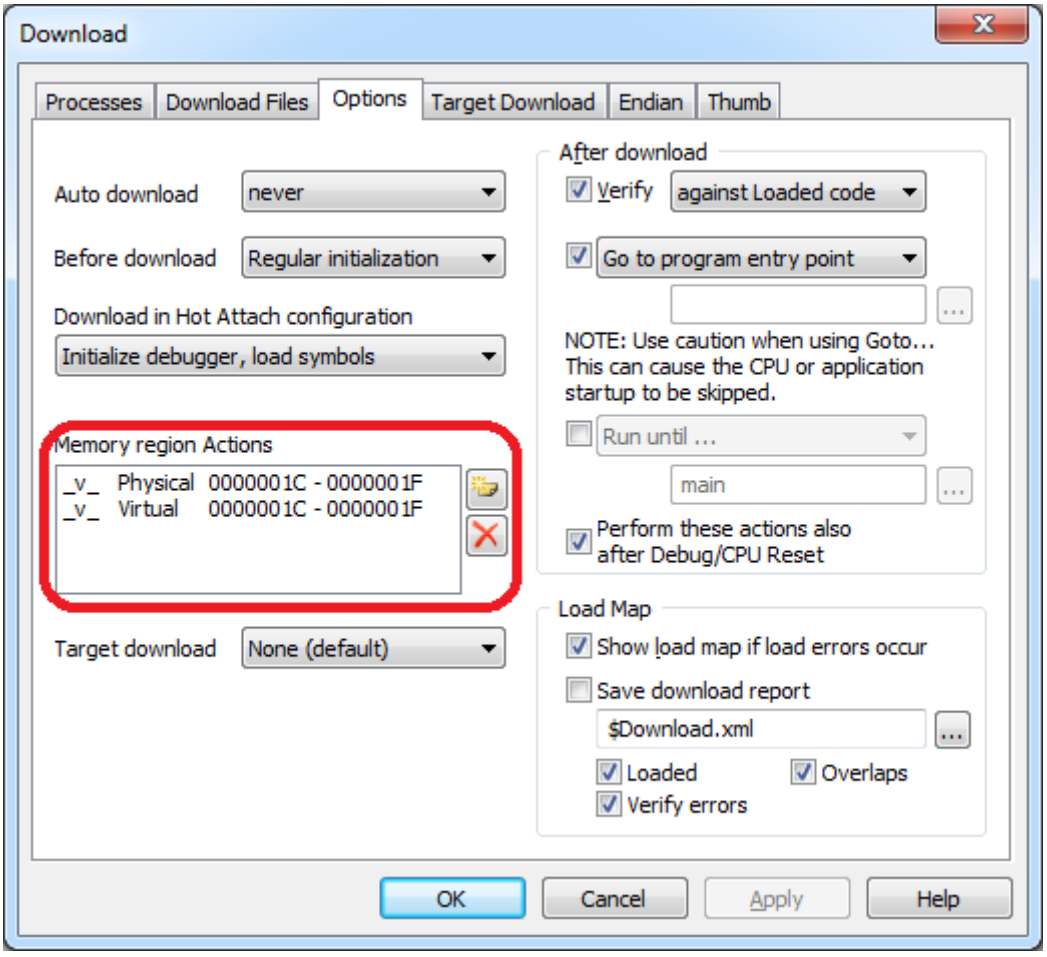
4.3 Internal Flash Programming

The debugger programs the code directly into the internal flash memory through the standard debug download. Based on the selected CPU the debugger identifies which code from the download file fits into the internal flash and loads it to the flash through the flash programming procedure hidden to the user. The flash programming procedure is implemented using NXP IAP (In-Application Programming) interface being already part of the CPU Flash Boot Loader firmware. All other code allocated outside of the flash boundaries is downloaded to the target through standard the memory writes.

Note: Proper target CPU must be selected in the 'Hardware/Emulation Options' dialog since corresponding flash programming procedure is selected based on the selected CPU.

Note: IAP commands use 32 bytes of space in the top portion of the on-chip RAM for execution. The user program should not use this space.

Due to the CPU requirements winIDEA extracts the necessary interrupt vectors from the download file before programming a 32-bit value to the 0x1C address, makes the 2's complement of the check-sum of these vectors and programs the calculated value to the 0x1C address. This yields the CPU starting from the user code after the reset. Consequentially when 'Verify download' is configured it's executed after the debug download and the user would normally get error at address 0x1C since the programmed value doesn't match with the one in the download file. The user can ignore this error or adjust his download file in a way that a 32-bit value at the address 0x1C contains proper value, which results in the CPU start executing the user code after the reset. The alternative is also to skip verifying 4 bytes at address 0x1C. Below picture shows the necessary setting in the Download dialog.



NXP LPC download verify exclusion

Code Read Protection (CRP)

Code Read Protection is a mechanism that allows user to enable different levels of security in the system so that access to the on-chip flash and use of the ISP can be restricted. When needed, CRP is invoked by programming a specific pattern in flash location at 0x000002FC.

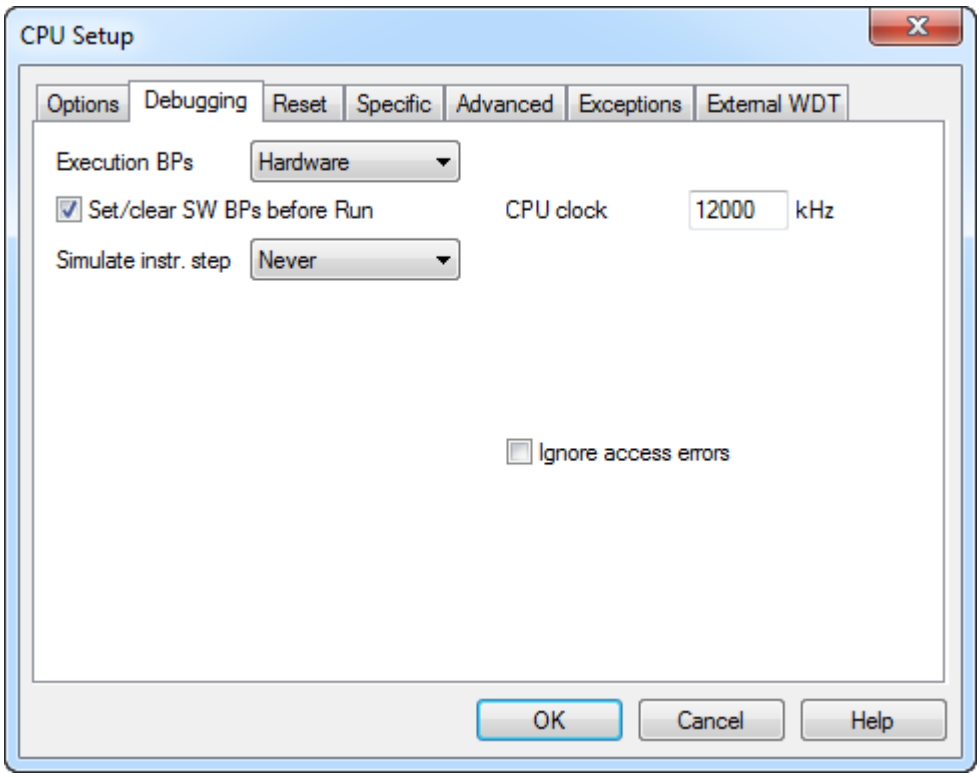
If value 0x12345678 is programmed to the address 0x2FC (CRP1), access to chip via the JTAG pins is disabled, which means debugger can no longer have control over the CPU via the JTAG debug interface. Hence, use code read protection with caution.

CPU clock and flash programming

Since most of Cortex-M LPC devices need CPU clock to correctly program flash memory, user must take care to correctly set the option in *CPU Setup -> Debugging -> CPU clock*.

Note: This parameter is also required for correctly decoding SWO trace.

In case that user application is reconfiguring CPU clock (using PLL to increase CPU clock, etc.), then initialization script can be used to have winIDEA initialize PLL to the same CPU clock that is later configured in application. This will ensure that *CPU Setup -> Debugging -> CPU clock* is set correctly at the time of initial download operation and during live debug session if flash memory is modified through memory window.

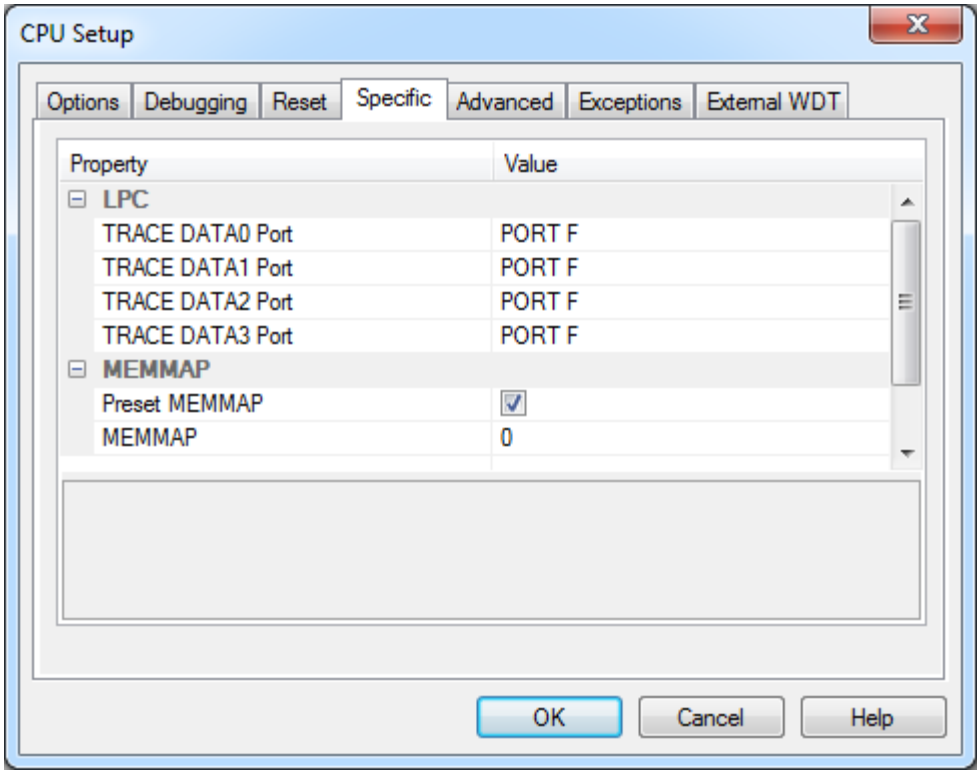


CPU Setup, Debugging options

Note: LPC15xx devices do not need CPU clock information to correctly program flash.

Note: LPC15xx devices can have parts of SRAM disabled to reduce power consumption. winIDEA requires that all SRAM is enabled for flash programming to work.

4.4 Specific Options



CPU Setup, NXP LPC options

Preset MEMMAP / SYSMEMREMAP

When the option is checked the debugger presets SYSMEMREMAP register.

TRACE DATAn Port

Select which trace pins should be used for parallel trace (TRACE).

4.5 LPC15xx Trace Specific Options

On LPC15xx devices user can configure I/O pin to use for SWO trace. User should take special care to correctly choose a pin that is not otherwise used by application. If SWO trace is not needed, it can be disabled by choosing appropriate option under *SWO Port*.

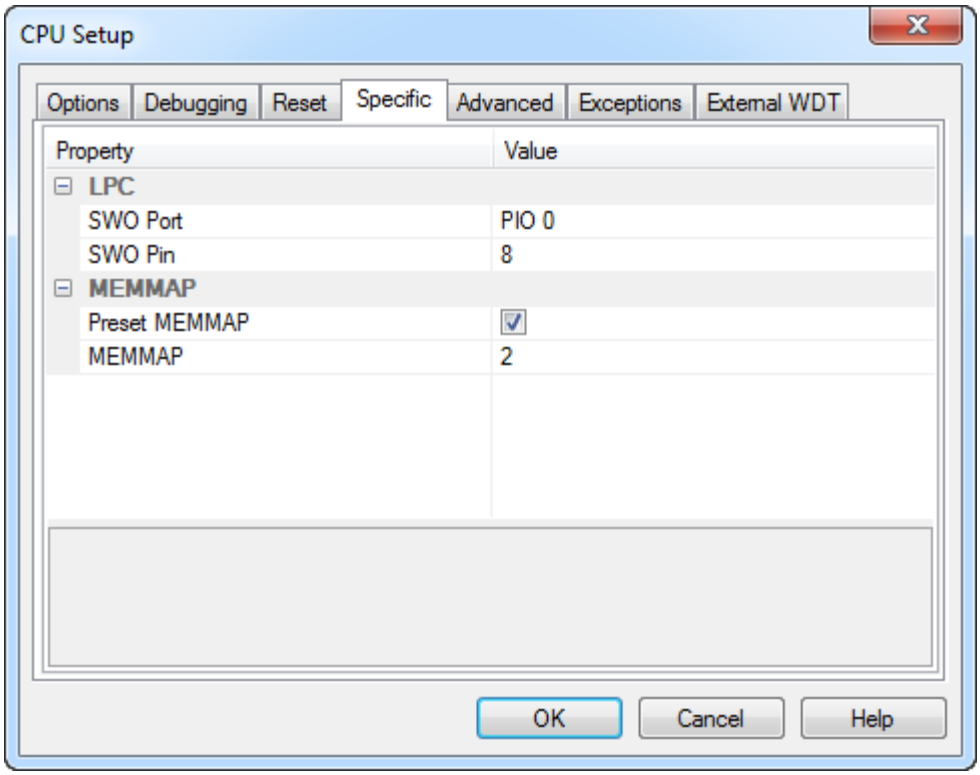
Note: Environment will remap selected pin to output SWO trace, disabling any function previously assigned by application.

SWO Port

Select I/O port that should be used for SWO trace. Set to DISABLE to prevent reconfiguration of I/O pin if SWO is not used.

SWO Pin

Select I/O pin that should be used for SWO trace. Set value 0..31 when PORT0 or PORT1 are selected and 0..11 when PORT2 is selected.



CPU Setup, LPC15xx trace specific options

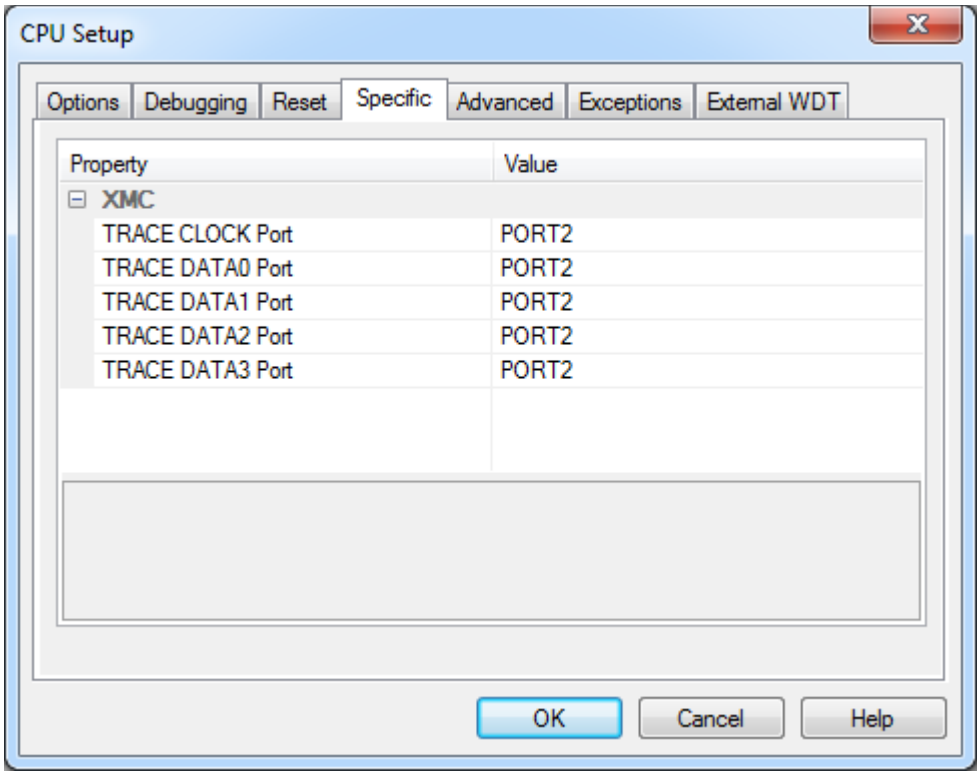
5 Infineon XMC

5.1 Halt On Reset (HAR)

XMC devices implement special functionality for stopping the CPU at reset vector after reset when debugger is connected. This is handled by iSYSTEM tools when **'Regular'** reset method is selected.

5.2 Trace port settings

Specify which IO port to use for parallel trace (**'TRACE'** trace protocol) in Specific pane of CPU Settings. Settings are only applicable on devices which support parallel trace.



CPU Setup, Infineon XMC options

5.3 Single Pin Debug (SPD)

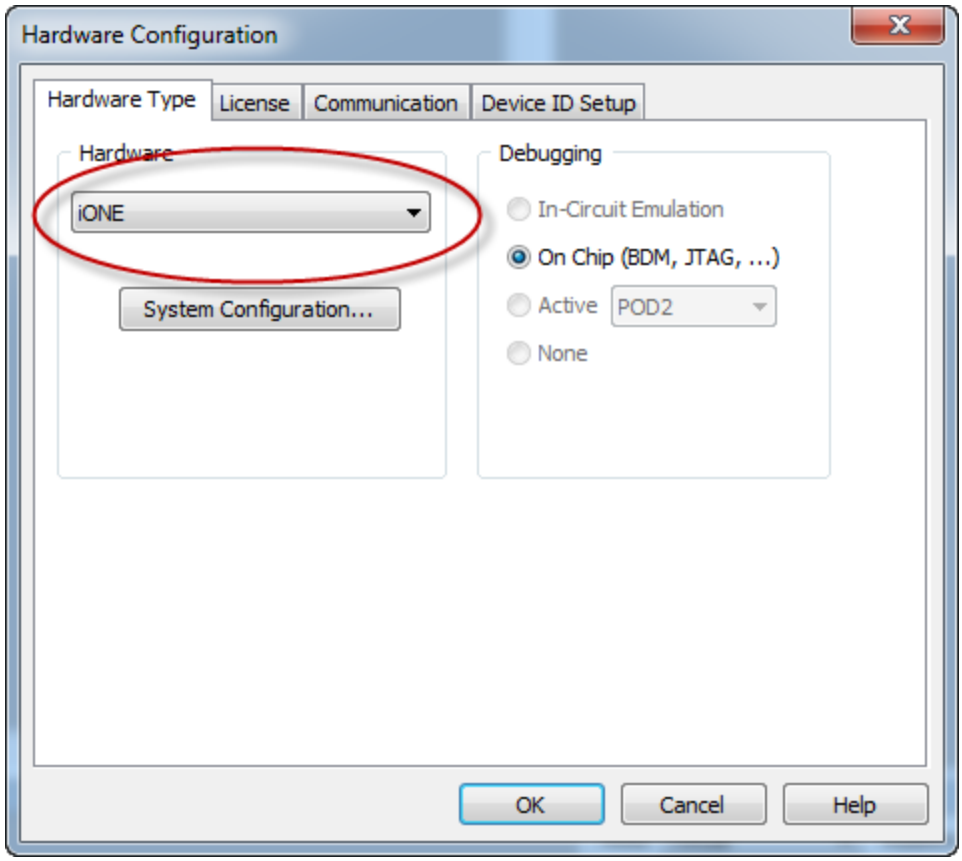
The SPD protocol based tool interface access allows to debug the system using a single pin only. The bit frequency is 2 MHz and allows an effective SWD telegram of 1.4 Mbits/s. The protocol is very robust against clock deviations between tool and device. The SPD protocol encodes the SWD protocol bits with the distance between the SPD
signal edges. A SWD value of '0' is encoded with a short distance of 0.5 µs, a value of '1' with a distance of 1 µs. This encoding is used in both transfer directions.

SPD is available on XMC1000 devices.

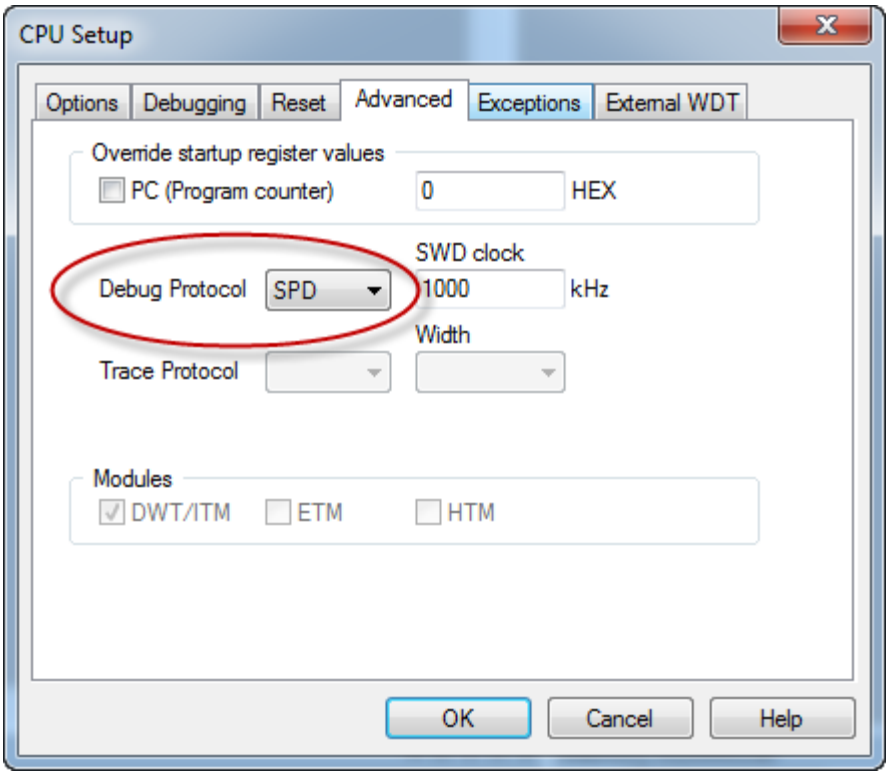
The Boot Mode of XMC1000 depends on the Boot Mode Index (BMI) value located at memory address 0x10000E00. To be able to debug device BMI must be in either Debug or HAR boot mode.

winIDEA supports SPD protocol only on iONE, iTAG and iBRIDGE hardware. To be able to select SPD debug protocol in *Hardware -> CPU Options… -> Advanced -> Debug Protocol*,

workspace must have selected XMC1000 device and the in dialog window Hardware -> Hardware… -> Hardware Type -> Hardware chosen platform must be iONE.



CPU Setup, Hardware selection



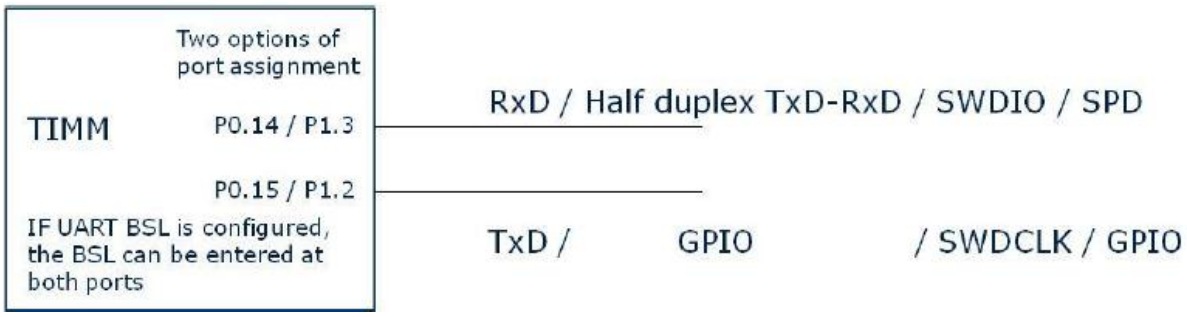
CPU Setup, Debug protocol selection

Note: XMC1000 device must be in either SPD or SWD mode before winIDEA can successfully establish connection.

BMI mode switching

winIDEA automatically changes BMI mode based on user selection of Debug Protocol. User should take special care to have Debug Protocol correctly selected before initiating debug session!

Note: XMC1000 devices have two options for debug port assignment.



Warning: Selecting SWD protocol, when only SWDIO pin is connected to debug connector, will switch XMC1000 device to SWD, but will then fail to establish debug connection, since SWDCLK is not connected!

Table: Mode switching in winIDEA

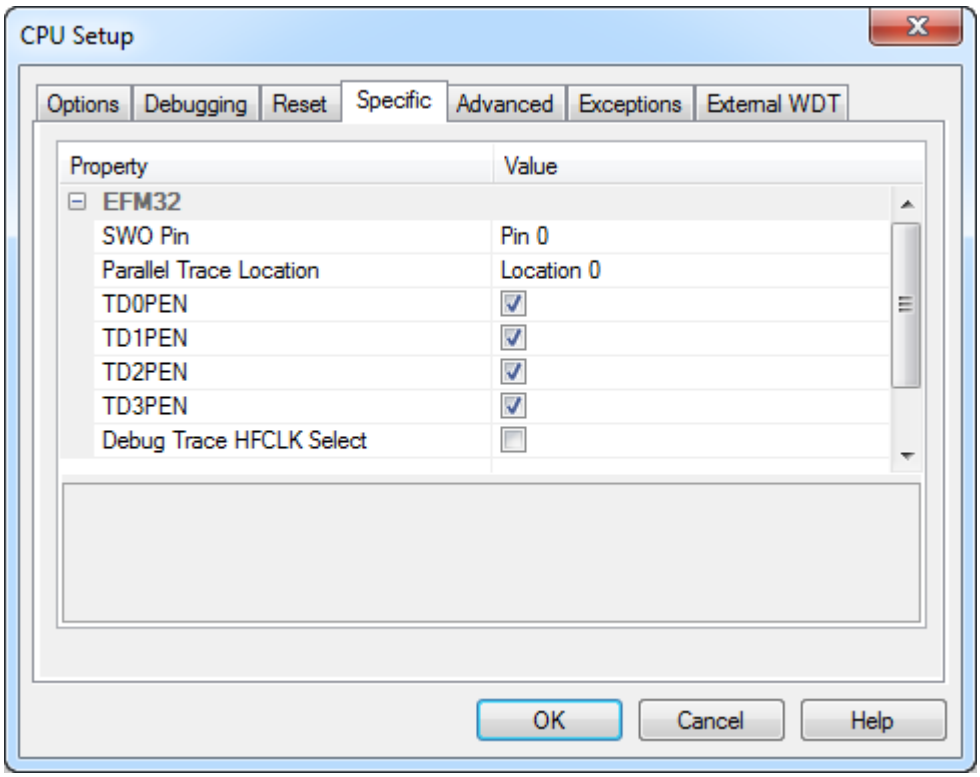
Current boot mode	New mode (SPD selected)	New mode (SWD selected)
ASC Bootstrap Load Mode (ASC_BSL)	(cannot switch)	(cannot switch)
User Mode (Productive)	(cannot switch)	(cannot switch)
User Mode (Debug) SWD0	User Mode (HAR) SPD0	User Mode (HAR) SWD0
User Mode (Debug) SWD1	User Mode (HAR) SPD1	User Mode (HAR) SWD1
User Mode (Debug) SPD0	User Mode (HAR) SPD0	User Mode (HAR) SWD0
User Mode (Debug) SPD1	User Mode (HAR) SPD1	User Mode (HAR) SWD1
User Mode (HAR) SWD0	User Mode (HAR) SPD0	(no change)
User Mode (HAR) SWD1	User Mode (HAR) SPD1	(no change)
User Mode (HAR) SPD0	(no change)	User Mode (HAR) SWD0
User Mode (HAR) SPD1	(no change)	User Mode (HAR) SWD1

6 Energy Micro EFM32

6.1 Trace port settings

Select SWO pin for when **'SWO'** trace protocol is used. Select the location for parallel trace port location for when **'TRACE'** is selected as trace protocol. Additionally only some of the trace pins can be enabled on trace port. Be sure to connect them accordingly to the debug/trace connector. Also be sure that the number of enabled pins matches the Trace Width set in Advanced pane of CPU Setup dialog.

HFCLK can be forced as trace clock when '**Debug Trace HFCLK Select'** is enabled.



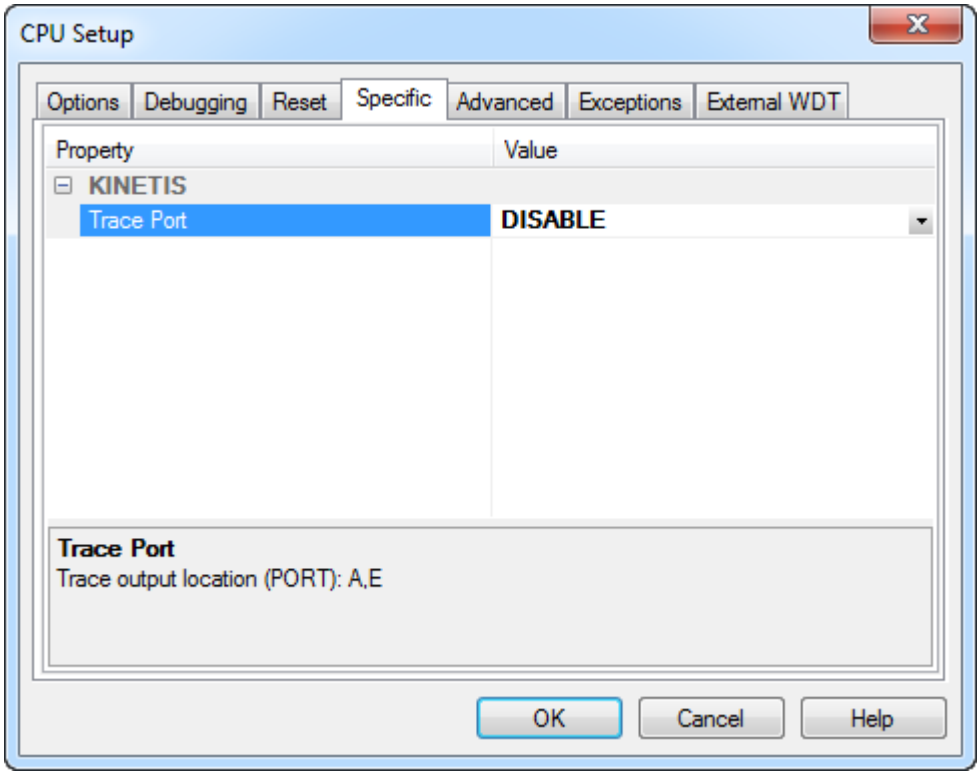
CPU Setup, Energy Micro EFM32 options

7 Freescale Kinetis

7.1 Trace port settings

Kinetis devices include several devices with various availability of Embedded Trace Macrocell (ETM) and output pins for parallel trace. User should check appropriate reference manual for what is available on target device.

Specify which IO port to use for parallel trace (**'TRACE'** trace protocol) in Specific pane of CPU Settings. Set to '**DISABLE**' if device does not support parallel trace output, you do not need trace and/or output pins are used by application.



CPU Setup, Freescale Kinetis options

7.2 Embedded Trace Buffer (ETB)

Embedded Trace Buffer is currently not supported.

8 Freescale Kinetis K2x

8.1 Flash Configuration Field

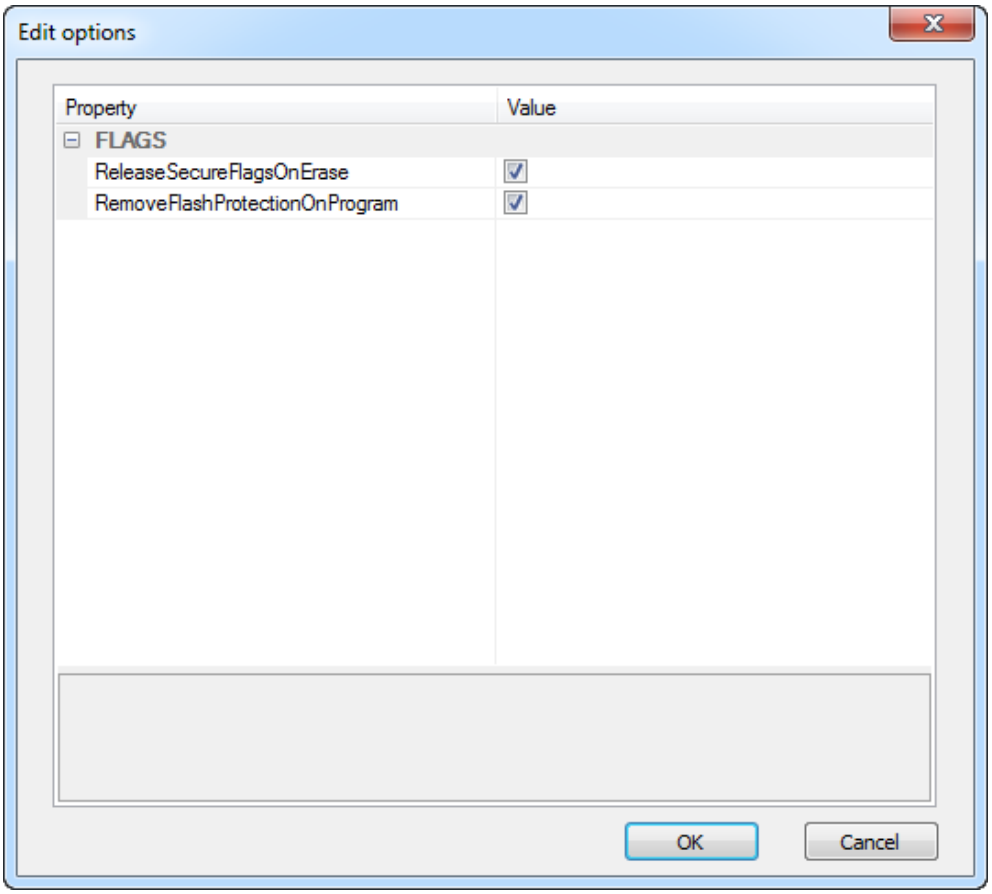
The program flash memory contains a 16-byte flash configuration field that stores default protection settings (loaded on reset) and security information that allows the MCU to restrict access to the flash memory module.
On Kinetis devices this configuration field is located in flash memory at location 0x400 to 0x40F. For more information please check reference manual for your device.

Important note: For K2x devices the range from 0x400 to 0x40F is defined as a special region in winIDEA that will not be downloaded into. This is to prevent users accidently locking the device. User should take care when linking application that it will avoid these memory locations!

9 Freescale Kinetis EA

9.1 Flash Configuration Field

Like on all Kinetis devices the EA family also has special Flash Configuration Field. Unlike on other Kinetis families handling of this filed is different.



FLASH Setup -> Configuration, Flags

Selecting **'ReleaseSecureFlagsOnErase'** will appropriately fix Flash Configuration Field when **'Mass Erase'** is used. This will write 0xFE to location 0x40E.

Selecting **'RemoveFlashProtectionOnProgram'** will appropriately fix Flash Configuration Field when downloading user application. This will write 0xFFFF to location 0x40C and 0x40D.

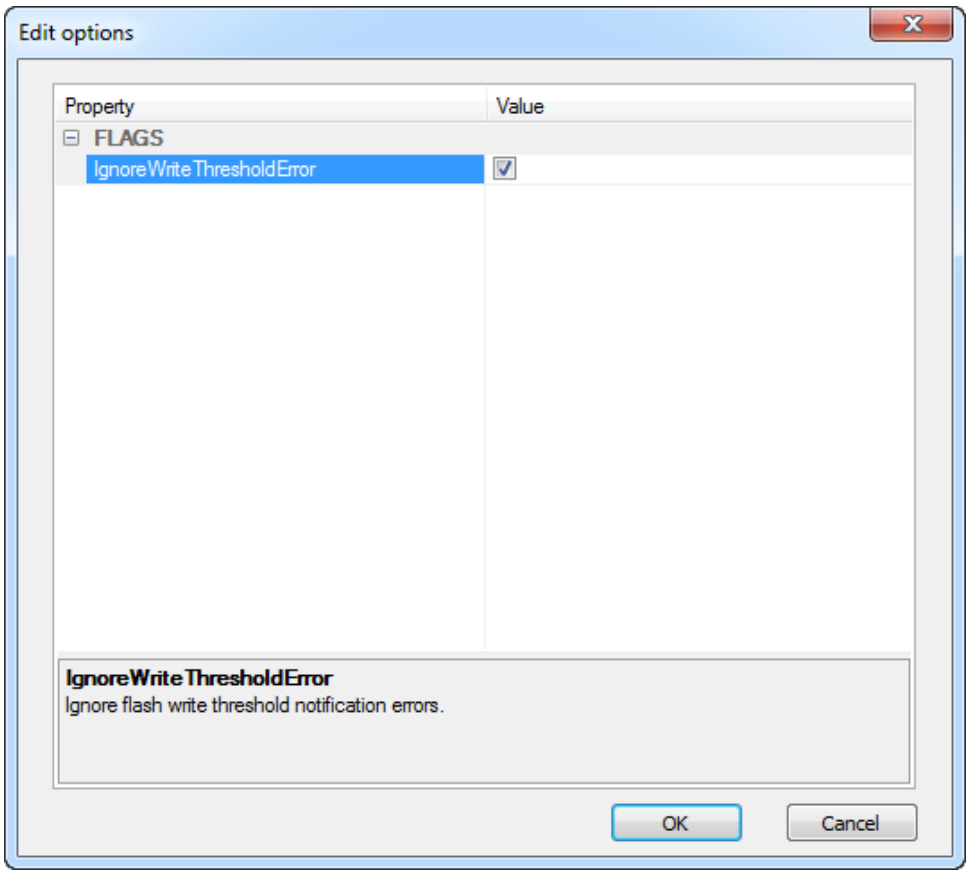
10 Microsemi SmartFusion2

10.1 Flash Configuration Field

SmartFusion2 devices have threshold detection for number of writes performed on FLASH memory. When this threshold is reached flash programming API returns an error status.

Selecting **'IgnoreWriteThresholdError'** will prevent winIDEA to abort flash programming operation when CPU returns this notification.

winIDEA can perform a verify after download, to detect any data not correctly written to flash memory.



FLASH Setup -> Configuration, Flags

11 Troubleshooting

This section addresses Cortex-M specific issues only. For common Cortex issues refer to the Cortex troubleshooting section and for tips and hints on problems, which are architecture independent refer to the general troubleshooting section.

11.1 Error 311: CoreSight initialization failed

When winIDEA reports error 311 during the initial debug connection double check that a specific microcontroller family is selected and not a generic core selection (Cortex-M0, Cortex-M3…) under the CPU list. For instance, for NXP LPC13xx target device, select LPC13xx selection; for STM32 target device select STM32 selection, etc. Selection of a generic core (e.g. Cortex-M3 for specific Cortex-M3 target device) will most likely report an error since every single microcontroller family requires some specific adjustments for the initial debug connection. In case the target device does not belong to any listed microcontroller family, please contact iSYSTEM technical support for assistance.

Make sure also that the particular target microcontroller is selected in the 'CPU variant' combo box.