Freescale MPC6xx & MPC7xx Family On-Chip Emulation

This document is intended to be used together with the CPU reference manual provided by the silicon vendor. This document assumes knowledge of the CPU functionality and the terminology and concepts defined and explained in the CPU reference manual. Basic knowledge of winIDEA is also necessary. This document deals with specifics and advanced details and it is not meant as a basic or introductory text.

Contents

1 Introduction

There are three fundamentally different debug interfaces in the MPC family. The MPC5xx/8xx uses the BDM style debug interface, while the PPC/MPC4xx/6xx/7xx use a debug interface based on the JTAG protocol. A newly developer Nexus interface on the MPC56x provides faster development access and, more importantly, instruction and data trace.

The debug interface of the MPC core uses the BDM, the JTAG or Nexus development port, which is a dedicated port that needs none or minimal of the regular system interfaces. System activity can be controlled from the development port when the core is in debug mode. The development port is a relatively inexpensive interface that allows the development system to operate at a lower frequency than the core's frequency (except for Nexus that operates at the core's frequency). In debug mode, the core fetches all instructions from the development port. Data can be read from or written to the development port. This allows memory and registers to be read and modified by a development tool (emulator) connected to the development port.

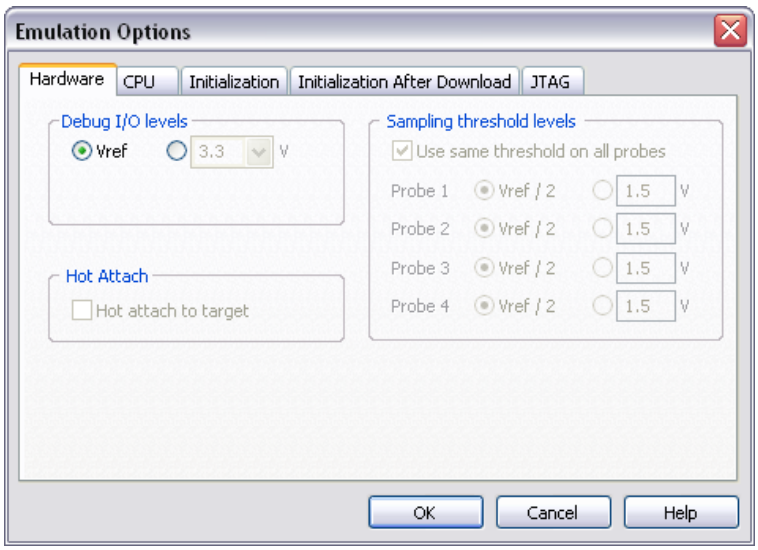
The JTAG development port of the 6xx/7xx, also known as COP, is scan chain based. The debug port of the MPC4xx cores and processors is not scan chain based, instead they have built a special debug module, similar to the MPC BDM.

Debug Features

- 1 or 2 hardware breakpoints
- Unlimited number of software breakpoints
- Fast flash programming
- No real-time access

2 Emulation Options

2.1 Hardware Options



Emulation Options, Hardware page

Debug I/O levels

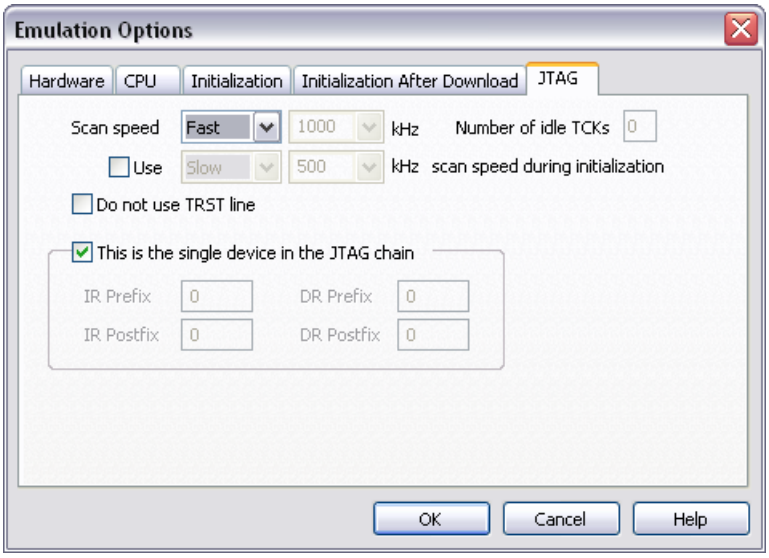
The development system can be configured in a way that the debug JTAG signals are driven at 3.3V target voltage level (Vref).

When 'Vref' Debug I/O level is selected, a voltage applied to the belonging reference voltage pin on the target debug connector is used as a reference voltage for voltage follower, which powers buffers, driving the debug
JTAG signals. The user must ensure that the target power supply is connected to the Vref pin on the target JTAG connector and that it is switched on before the debug session is started. If these two conditions are not meet, it is highly probably that the initial debug connection will fail already. However in some cases it may succeed but then the system will behave abnormal.

2.2 Initialization Sequence

Before the flash programming or download can take place, the user must ensure that the memory is accessible. This is very important since there are many applications using memory resources (e.g. external RAM, external flash), which are not accessible after the CPU reset. In that case, the debugger must execute after the CPU reset a so called initialization sequence, which configures necessary CPU chip selects and then the download or flash programming can actually take place. The user must set up the initialization sequence based on his application. Detailed information may be found in the Initialization Sequence help topic.

2.3 JTAG Scan Speed



JTAG Scan Speed definition

Scan speed

The JTAG chain scanning speed can be set to:

- Slow - long delays are introduced in the JTAG scanning to support the slowest devices. JTAG clock frequency varying from 1 kHz to 2000 kHz can be set.
- Fast – the JTAG chain is scanned with no delays.
- Burst – provides the ability to set the JTAG clock frequency varying from 4 MHz to 100 MHz.

- Burst+ - provides the ability to set the JTAG clock frequency varying from 4 MHz to 100 MHz

Slow and Fast JTAG scanning is implemented by means of software toggling the necessary JTAG signals. Burst mode is a mixture of software and hardware based scanning and should normally work except when the JTAG scan frequency is an issue that is when the JTAG scan frequency used by the hardware accelerator is too high for the CPU. In general, selecting an appropriate scan frequency usually depends on scan speed limitations of the CPU. In Burst+ mode, complete scan is controlled by the hardware accelerator, which poses some preconditions, which are not met with all CPUs. Consequentially, Burst+ mode doesn't work for all CPUs.

In general, Fast mode should be used as a default setting. If the debugger works stable with this setting, try Burst or Burst+ mode to increase the download speed. If Fast mode already fails, try Slow mode at different scan frequencies until you find a working setting.

Note: Burst and Burst+ modes are implemented for PowerPC and ARM CPUs, including XScale.

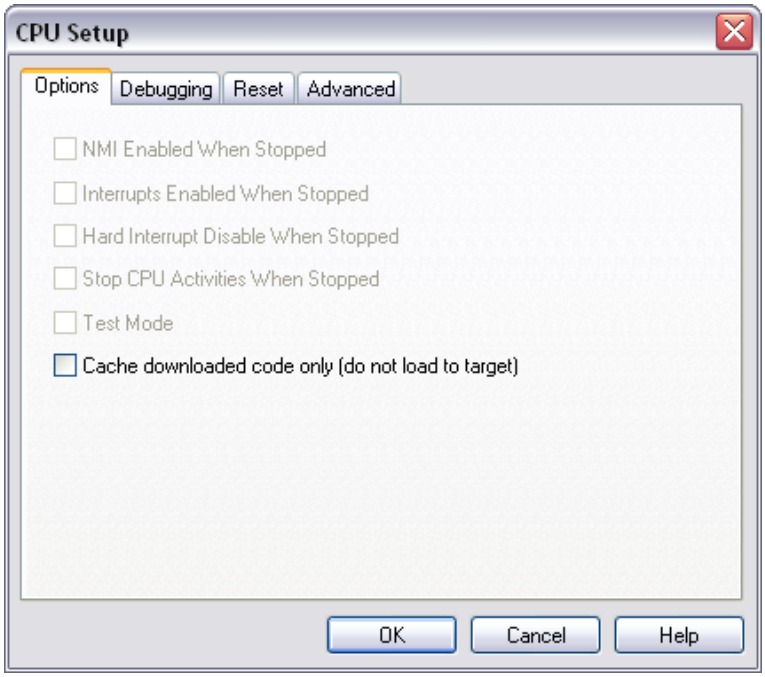
Use – Scan Speed during Initialization

On some systems, slower scan speed must be used during initialization, during which the CPU clock is raised (PLL engaged) and then higher scan speeds can be used in operation. In such case, this option and the appropriate scan speed must be selected.

3 CPU Setup

3.1 General Options

The CPU Setup, Options page provides some emulation settings, common to most CPU families and all emulation modes. Settings that are not valid for currently selected CPU or emulation mode are disabled. If none of these settings is valid, this page is not shown.



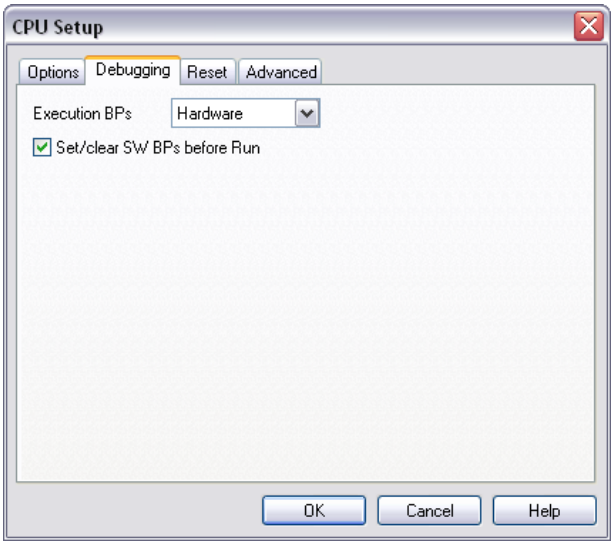
PowerPC, General Options

Cache downloaded code only (do not load to target)

When this option is checked, the download files will not propagate to the target using standard debug download but the Target download files will.

In cases, where the application is previously programmed in the target or it's programmed through the flash programming dialog, the user may uncheck 'Load code' in the 'Properties' dialog when specifying the debug download file(s). By doing so, the debugger loads only the necessary debug information for high level debugging while it doesn't load any code. However, debug functionalities like ETM and Nexus trace will not work then since an exact code image of the executed code is required as a prerequisite for the correct trace program flow reconstruction. This applies also for the call stack on some CPU platforms. In such applications, 'Load code' option should remain checked and 'Cache downloaded code only (do not load to target)' option checked instead. This will yield in debug information and code image loaded to the debugger but no memory writes will propagate to the target, which otherwise normally load the code to the target.

3.2 Debugging Options



PowerPC Family Debugging Options

Execution Breakpoints

Hardware Breakpoints

Hardware breakpoints are breakpoints that are already provided by the CPU. The number of hardware breakpoints is limited to four. The advantage is that they function anywhere in the CPU space, which is not the case for software breakpoints, which normally cannot be used in the FLASH memory, non-writeable memory (ROM) or self-modifying code. If the option 'Use hardware breakpoints' is selected, only hardware breakpoints are used for execution breakpoints.

Note that the debugger, when executing source step debug command, uses one breakpoint. Hence, when all available hardware breakpoints are used as execution breakpoints, the debugger may fail to execute debug step. The debugger offers 'Reserve one breakpoint for high-level debugging' option in the Debug/Debug Options/Debugging' tab to circumvent this. By default this option is checked and the user can uncheck it anytime.

Software Breakpoints

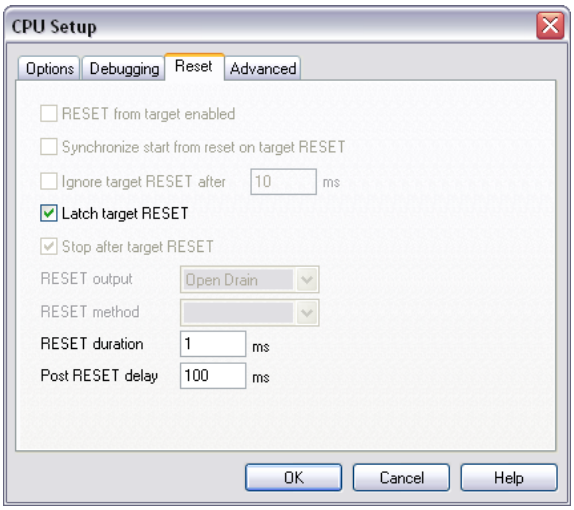
Available hardware breakpoints often prove to be insufficient. Then the debugger can use unlimited software breakpoints to work around this limitation.

When a software breakpoint is being used, the program first attempts to modify the source code by placing a break instruction into the code. If setting software breakpoint fails, a hardware breakpoint is used instead.

Set/clear SW BPs before Run

When the option is checked, then a software breakpoint is not set/cleared immediately, but is just remembered. Only when the CPU is set to running are the breakpoints committed. This way several breakpoints can be changed but only one re-FLASH operation takes place. This is especially noticeable in testIDEA operation with many stubs and also during a regular debugging session when several breakpoints are set/cleared within the same flash erase block.

4 Reset Options



PowerPC Family Reset Options

Latch target RESET

When the option is checked (default), the debugger latches active target reset until it gets processed. This yields a delay between the target reset and restart of the application from reset. If this delay is not acceptable for a specific application, the option should be unchecked. An example is an application where the CPU is periodically set into a power save mode and then waken up e.g. every 6ms by an external reset circuit. In such case, a delay introduced by the debugger would yield application not operating properly.

When the option is unchecked, it may happen that the debugger does not detect the target reset although the CPU gets reset. The debugger polls the CPU status ~3 times per second while the target reset can occur in between.

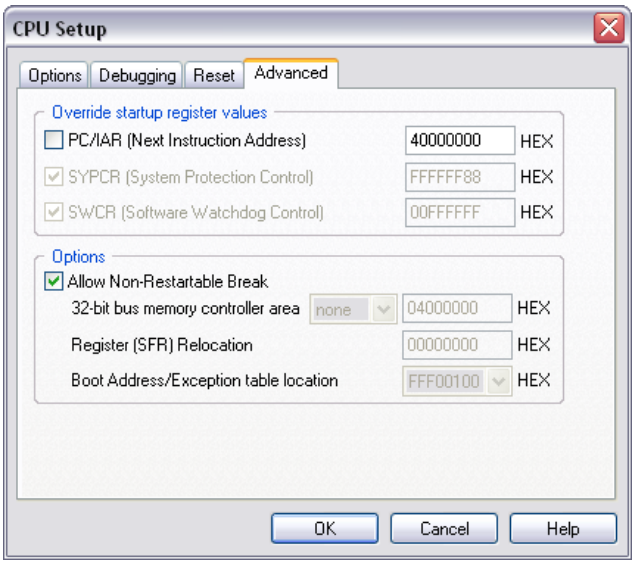
Reset Duration

The width of the RESET pulse is specified here.

Post RESET Delay

Typically, the on-chip debug module is reset concurrently with the CPU. After the CPU reset line is released from the active state, the on-chip debug module requires some time (delay) to become operational. The default delay value normally allows the debugger to gain the control over the CPU. If a first debug connection fails already try different delay values to establish the debug connection.

5 Advanced Options



PowerPC Family Advanced Options

Override Startup Register Values

If required, the Emulator can change the checked register after the CPU is released from reset. These settings have typically to do with setting the default program starting point and disabling the watchdog.

Caution: If in Hardware/Emulation options/Initialization the one of the Reset&Run options is used the run after reset will be started from this address. A Run attempt will fail, if there is no valid code there right after reset. It is recommended that the two options, Override PC and Initalization/Reset&Run, are not used at the same time.

Allow Non-Restartable Break

Normally, for stopping the CPU a so-called soft-stop (halt) debug command is used. All clocks are stopped after all pending instructions have executed, including all bus interface actions.

If the soft-stop is not successful and this option is checked then a hard-stop (freeze) command is issued. In this case, all clocks on the device are immediately stopped. Due to the asynchronous nature of the JTAG TAP controller, the stopping of the clocks will be indeterminate. Thus, restartability is not guaranteed.

32-bit bus memory controller area

Some MPC6xx PowerPC implementations have proprietary bus controllers or possible such bus configurations that may have limited functionality of not supporting standard 64-bit access for memory reads and writes. With this option it is possible to define a memory area in which a 32-bit access width is forced.

Register (SFR) Relocation

For some MPC6xx PowerPC implementations the software must know the location of SFRs after reset in order to correctly execute an initialization sequence. The address offset entry is also used later for SFR access during a debug session.

Boot Address/Exception Table Location

The location of boot address or exception table can be specified here – required for some MPC6xx implementations.

6 Getting Started

Before powering on check the Power supply setting in Emulation Options/Hardware. Default is 3.3V, note however that certain newer devices operate at lower voltages and may no longer be 3.3V-tolerant. Use ‘Target’ setting in such case. Target board should provide its I/O reference voltage on the debug connector.

Disable watchdog with the option described above. Usually the watchdog timer will timeout after just a few seconds after reset is released, leaving very little time for user intervention. At the same time, don’t forget to disable or adequately service the watchdog timer in your software.

To enable code download user should provide some minimum memory access configuration. This can be trivial by merely setting the appropriate chip-select registers for flash or SRAM, or complicated and lengthy for DRAM. Note that there’s no need for that with the 5xx/8xx that have some internal RAM sufficient for small test code.

See notes above for setting debug clock and reset delay.

In case the debugger won’t start, disable initialization and all configurable options except disabling the watchdog. Check if the CPU stops at reset vector 0xFFF00100, or 0x100 if MSR_{IP} is 0.

It is recommended that CPU clock PLL is not modified in the initialization sequence as this may cause CPU debug port to fail. Instead, user code should do this in any case.

Under Emulation Options/CPU Setup check all exceptions that don’t have a handler in your code. This is a good means to detect early when your code strays away.

7 Emulation Notes

Ensure that the emulator /HRESET connector pin is connected directly to the /HRESET signal of the processor. This will provide the ability for the emulator to drive and sense the status of /HRESET. The target design should only drive the /HRESET with open collector or open drain type devices.

To guarantee that JTAG Mode is not accidentally invoked, connect a pull down (~10K) resistor on the /TRST signal and a pull-up (~10K) on the TMS signal. When the MPC’s development port (BDM) is used, JTAG functionality is disabled. Designs that require both should have a Reset configuration scheme to support the two modes. We use 10k pull-down on a '245 buffer that drives DBPC=00 and enables BDM functionality. And if a JTAG tester is connected it overcomes the pull-down to a logic high and the DBPC=11 enabling JTAG pins.

8 Troubleshooting PPC750CX/Cxe

Because the CXe operates on 2.5V, the 'Vref' debug I/O levels must be selected in the ‘Hardware/Emulation Options/Hardware’ pane. The CX will work with both settings, 3.3V and Vref.