Freescale Semiconductor
Application Note

Document Number: AN2614
Rev. D, 18/March/2010

MPC553x, MPC555x, and MPC556x Family Nexus Interface Connector

MICTOR-38, Robust, and JTAG Connector Definitions

by: Randy Dees
Applications Engineering
Austin,Tx
USA

1 Introduction

The IEEE-ISTO 5001™-2003 Nexus standard (referred to as the Nexus standard) interface is an industry standard that crosses CPU boundaries and allows industry-standard tools to support multiple CPU architectures. It allows advanced debug capabilities by providing high-speed access to the microcontroller core. These advanced debug capabilities include trace, without requiring extensive external circuitry to monitor an external address bus. The Nexus standard defines a minimum feature set that must be implemented. To simplify discussion of debug capabilities between different requirements/capabilities, the minimum feature set requirements are divided into four classes.

- Nexus Class 1 defines basic run control capabilities that must be implemented.
- A Nexus compliant Class 2 or greater device allows non-intrusive instruction and ownership trace on a microcontroller in single-chip mode or a microcontroller without external data and address buses.
- A Nexus Class 3 or greater device also allows for data trace and read/write access to memory while the microcontroller is running.
- Class 4 supports some advanced debug capabilities, but Freescale currently has implemented only a few Class 4 features and does not support all of them. Therefore there are no devices or cores that support the full Class 4 requirements.

Contents

NOTE

In general, Freescale has implemented the Nexus Read/Write Access (RWA) mechanism in all devices using the Nexus interface, regardless of whether the device supports Nexus Class 1, Class 2, or Class 3, even though RWA is a Class 3 feature. In addition, all e200z*x* Nexus clients that support trace also implement the Nexus Class 4 feature of enabling and disabling of trace upon occurrence of a watchpoint. As these are features that are above the basic requirements of a given class, generally, these are referred to as a "+" feature. In other words, for a client that support all Nexus class 2 features and supports RWA, the implementation is referred to as Class 2+.

The MPC5554 microcontroller was the first device in the MPC5500 family that contained multiple Nexus clients that communicate over a single IEEE-ISTO 5001™-2003 Nexus Combined JTAG IEEE 1149.1/Auxiliary Out interface. When combined, all of the Nexus clients are referred to as the Nexus Development Interface (NDI). Class 3 Nexus allows for Program, Data, and Ownership Trace of the MPC5500 device execution without access to the external data and address buses.

This application note addresses the connector options and recommendations for the MPC5554 and other members of the MPC5500 family, including the MPC5533, MPC5534, MPC5553, MPC5561, MPC5565, MPC5566, and MPC5567. While the connector definitions for the MPC5510 family of devices or any newer MPC56xx families of devices may be the same, some operations may be slightly different and have different requirements.

2 MPC5500 Nexus Architectural Overview

The MPC5500 devices implement a varying number of internal Nexus clients. Collectively, all of the JTAG and Nexus clients are known as the Nexus Development Interface (NDI). The typical interconnections between the different Nexus clients are shown in the following figure. Tools can communicate with each of these clients by selecting the client through the JTAG Controller (JTAGC) and the Nexus Port Controller (NPC).

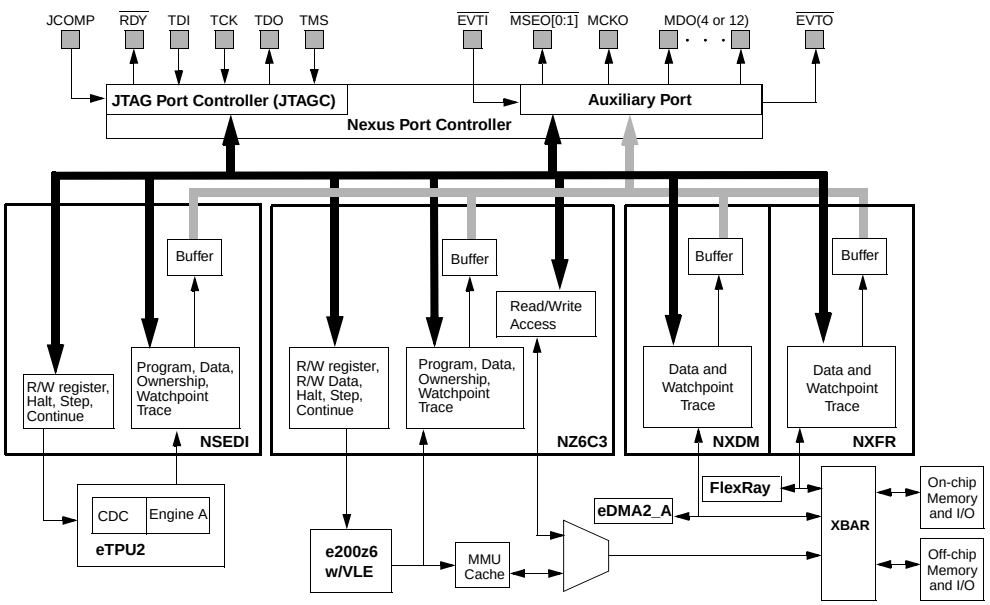


Figure 1. Typical MPC5500 Nexus Development Interface (MPC5567 shown)

The Nexus clients can consist of many types, but the typical clients are:

- e200z*x* core - In the MPC5500 devices, the primary processing cores are Power Architecture based cores. Current available e200z*x* cores are the e200z0, e200z1, e200z3, e200z4, e200z6, e200z6 with VLE, and the e200z7. Some devices implement multiple cores, some as true dual (homogeneous) cores and some as a primary core and a secondary core. The secondary

core is a smaller core intended for off-loading the primary core. The Nexus client is usually named for the core that it supports plus the Nexus class that the client supports. For example, the Nexus client for the e200z4 that supports Class 3 functionality is called the NZ4C3.

- eTPU - The Enhanced Timing Processor (eTPU or eTPU2) is a task-specific computing engine that is designed primarily for dealing with timing functions and includes some mathematics capabilities (including multiplication and division). The eTPU Nexus client is implemented as either a dual eTPU engine (Nexus Dual eTPU Development Interface - NDEDI) or a single eTPU engine client (Nexus Single eTPU Development Interface - NSEDI). In addition to the eTPU engine interface, the NDEDI/NSEDI interfaces with a secondary Nexus client, the Coherent Data-Parameter Controller (CDC). The CDC supports the shared memory and logic between the dual eTPU engines. On NDEDI devices, the eTPU Nexus interface appears to have three sub-clients (the two eTPU engines and the CDC). On NSEDI devices there are two (one engine and the CDC).
- Bus Trace Clients - Another trace client available is a Nexus client that traces data accesses made through a Crossbar bus port. Usually, these have been implemented on the master side of the crossbar switch for the eDMA module and the FlexRay Communication Interface. This allows master accesses made by these bus masters to peripherals or memory accessed though the Crossbar switch bus (XBAR). Freescale has chosen to implement these interfaces on the master side of the crossbar for most devices, but this module can also be instantiated on the slave side of the crossbar switch on future devices, allowing trace of all accesses into a particular slave port of the crossbar switch bus. Bus trace clients are referenced by the the type of interface they support. For example, the Nexus eDMA Master XBAR client is referred to as the NXDM; the FlexRay Master XBAR trace client is the NXFR.

2.1 Nexus Class 1 JTAG-Only Operation

For Nexus class 1 (run control only) operation, the JTAG pins can be used for standard, stopped mode debug operations. In this mode, Nexus does not have to be enabled and the auxiliary port pins can be ignored by the debugger. Since there is no JTAG standard connector pin-out definition, Freescale recommends that class 1 only debuggers use the same connector options as full Nexus debuggers. This allows customers to avoid putting two separate connectors on boards for Nexus versus JTAG-only debug tools. However, this document includes a definition for a 14-pin BERG style connector (2 x 7, 0.1" pin centers) for extremely low-cost applications that cannot use one of the full Nexus connectors.

NOTE

The Nexus Class 3 feature read/write access is accessible through a JTAG connection. Other Class 3 features are not available unless a full Nexus connector is used.

NOTE

It is recommended that all the full Nexus signals be made available (somewhere) on production boards to facilitate debugging new boards and analysis of errors in software, even on boards that have restricted space and provide a JTAG-only connection. If all of the Nexus signals are available on the production board, an adapter could be built to provide a Nexus connection on boards that do not have a complete footprint for one of the standard Nexus connectors; likewise, the JTAG connector does not have to be populated on production boards. It could even use a smaller connector footprint. This could be used with an adapter to the standard debug connections.

3 Nexus Connector Recommendation and Pin-Out Definitions

For the MPC5500 family, Freescale recommends that a single 38-pin MICTOR connector be used on all boards for both reduced- and full-port Nexus modes (M38C). In cases where a more robust connector is required, Freescale recommends that the 51-pin Glenair connector be used (R51C).

The table below shows the possible connectors that can be used for the MPC5500 family devices.

Table 1. Recommended Nexus Connectors for the MPC5500 Family

Connector	Designation	Part number (for target system)	Total number of pins	Maximum message data out signals (per Nexus standard)
3M MICTOR	M38C	767054-1[1]	38 (+ 5)	8 (12 by extension)[2]
3M MICTOR	M38-2C	767054-1[1] X 2	76 (+ 10)	16
Glenair MicroD Robust	R51C	MR7580-51P2BNU	51	8 (12 by extension)[2]

1. Other compatible part numbers are 2-5767004-2 (RoHS compliant), 2-767004-2, 767061-1, and 767044-1.
2. Although the Nexus standard definition normally only allows eight MDO signals, Freescale has extended the definition to include an extra four MDO signals.

Since full-port mode requires twelve MDO pins and both of these connectors support only eight MDO pins, some of the Vendor_IO and Tool_IO pins must be defined for MDO signals (MDO[8:11]). The MICTOR option has five vendor-defined pins (four for the extra MDO signals and then one extra pin), but the 51-pin robust connector only has three Vendor_IO pins. Therefore, to support the full twelve MDO signals, two Tool_IO pins had to be defined as MDO signals. The table below shows the recommended signal usage for the vendor-defined I/O pins, as well as tool vendor-defined I/O pins. Freescale has worked with tool vendors to insure minimal impact to tool needs on these pins.

Table 2. Extended MDO definition to the Nexus Standard Connector using IO for additional MDO Signals

Nexus pin designation	MPC5500 MICTOR definition	MPC5500 robust definition
VEN_IO0	MDO9/GPIO80	MDO9/GPIO80
VEN_IO1	MDO11/GPIO82	MDO11/GPIO82
VEN_IO2	BOOTCFG1/$\overline{\text{IRQ3}}$/GPIO212	$\overline{\text{RSTOUT}}$[1]
VEN_IO3	MDO8/GPIO70	-[2]
VEN_IO4	MDO10/GPIO81	-
TOOL_IO0	-[3]	-
TOOL_IO1	-	
TOOL_IO2	-	MDO8/GPIO70
TOOL_IO3	$\overline{\text{RSTOUT}}$	MDO10/GPIO81

1. This pin has been redefined since the original definition in 2004. Customers may want to make this signal selectable (via a 0 ohm resistor option) between $\overline{\text{RSTOUT}}$ and BOOTCFG1/$\overline{\text{IRQ3}}$/GPIO212.
2. This signal is not available on the 51-pin robust connector
3. This pin is defined for use by tool vendors and has no defined connection to the MPC5500 device for the MICTOR connector option.

3.1 Port Width Considerations

The 5001-2003 Nexus standard allows the Nexus Auxiliary Output Port to have 0[1]-16 message data outputs (MDO) pins. The MPC5500 family have two options for Nexus port widths; reduced-port mode has four MDO pins and full-port mode has twelve MDO pins. All levels (Class 1, 2, and 3) of Nexus trace can be supported over either port width, but the reduced port width limits the bandwidth available for trace information.

1. The 5001-2003 allows for a zero-width auxiliary output port to support Class 3 features by embedding the auxiliary output data into a specially formatted JTAG message. The MPC5500 devices do not support embedding trace information in a JTAG message.

Table 3. MPC5500 Family Nexus Port Width

Port Mode	Number of MDO Signals
Reduced Port Mode (RPM)	4
Full Port Mode (FPM)	12

The table below shows the port widths available in each of the possible package types for the currently defined members of the MPC5500 family.

Table 4. MPC5500 Family Nexus Port Width Options

Package Size (Balls)	Reduced Auxiliary Port Available /Size	Full Auxiliary Port Available/Size
208	Yes / 4	No / 0
324	Yes / 4	Yes / 12
416	Yes / 4	Yes / 12
496[1]	Yes / 4	Yes / 12

1. The 496 package is not a standard production package that can be purchased by customers. It is used in the VertiCal emulation device.

3.2 MPC5500 JTAG Connector

The figure below shows the pinout of the recommended JTAG connector to support the MPC5500 devices. If there is enough room allowed in the target system, a full Nexus connector is preferred over the simple 14-pin JTAG connector since it allows a higher degree of debug capability. It can be used as a minimum debug access or for BSDL board testing.

The recommended connector for the target system is the Tyco part number 2514-6002UB.

NOTE

This pinout is similar to the Freescale MCORE and DSP JTAG/OnCE connector definitions.

Table 5. Recommended JTAG Connector Pinout

Description	Pin	Pin	Description
TDI	1	2	GND
TDO	3	4	GND
TCK	5	6	GND
$\overline{\text{EVTI}}$[1]	7	8	-
$\overline{\text{RESET}}$	9	10	TMS
VREF	11	12	GND
$\overline{\text{RDY}}$[2]	13	14	JCOMP

1. $\overline{\text{EVTI}}$ is optional and was not included in the original (very early) definitions of the JTAG-only connector.
2. The $\overline{\text{RDY}}$ signal is not available on all packages or on all devices. Check the device pinout specification. In general it is not available in packages with 208 signals or less.

NOTE

Freescale recommends that a full Nexus connector be used for all tool debug connections regardless of whether Nexus trace information is needed. Adapters for a JTAG Class 1 14-pin connector (tool side) to the full Nexus MICTOR connectors (board side) are available from P&E Microcomputer Systems (http://www.pemicro.com), part number PE1906, and from Lauterbach (http://www.lauterbach.com), order number LA-3723

(CON-JTAG14-MICTOR). Lauterbach also has an adapter that will connect a MICTOR connector (tool side) to a 14-pin JTAG connector (board side). This adapter is order number LA-3725 (CON-MIC38-J14-5500).

3.3 MICTOR Connector Definition for the MPC5500 Family

The following table shows the complete signal usage for the MPC5500 full-port mode MICTOR connector. This uses the Vendor_IO pins 1–4 as MDO[11:8]. This connector may also be used for reduced-port mode (which only uses MDO[3:0]). While only one MICTOR is recommended, some tools may not support this configuration. For maximum tool compatibility, a second MICTOR may need to be added for the upper four MDO signals (MDO[11:8]). If the second connector is used, the layout of the two connectors should conform to the dimensions in the figure following the tables.

Table 6. MPC5500 Family MICTOR Connector M38C

MPC5500 signal	Combined M38C or M38-2C						Combined M38C or M38-2C	MPC5500 signal
-	Reserved[1]	-	1	Ground	2	-	Reserved[1]	-
-	Reserved[1]	-	3		4	-	Reserved[1]	-
MDO9/GPIO80	VEN_IO0	Out	5		6	Out	CLOCKOUT	CLKOUT
BOOTCFG1/$\overline{\text{IRQ3}}$/GPIO212	VEN_IO2	In	7		8	Out	VEN_IO3	MDO8/GPIO70
$\overline{\text{RESET}}$	/RESET	In	9		10	In	/EVTI	$\overline{\text{EVTI}}$
TDO	TDO	Out	11		12		VTREF	VDDE7
MDO10/GPIO81	VEN_IO4	Out	13		14	Out	/RDY	$\overline{\text{RDY}}$
TCK	TCK	In	15		16	Out	MDO7	MDO7/GPIO78
TMS	TMS	In	17	Ground	18	Out	MDO6	MDO6/GPIO77
TDI	TDI		19		20	Out	MDO5	MDO5/GPIO76
JCOMP	/TRST	In	21		22	Out	MDO4	MDO4/GPIO75
MDO11/GPIO82	VEN_IO1	Out	23	Ground	24	Out	MDO3	MDO3
$\overline{\text{RSTOUT}}$	TOOL_IO3	Out	25		26	Out	MDO2	MDO2
-[2]	TOOL_IO2		27		28	Out	MDO1	MDO1
-[2]	TOOL_IO1		29		30	Out	MDO0	MDO0
12 volts	UBATT		31		32	Out	/EVTO	$\overline{\text{EVTO}}$
12 volts	UBATT		33		34	Out	MCKO	MCKO
-[2]	TOOL_IO0		35		36	Out	/MSEO1	$\overline{\text{MSEO1}}$
VSTBY	VALTREF		37		38	Out	/MSEO0	$\overline{\text{MSEO0}}$

1. Pins 1 through 4 should be considered "reserved" and may be used by some logic analyzers as ground connections. If care is taken (and the proper cables are used), these pins could be used for customer I/O signals. However, check with the tool vendors used.
2. This optional Nexus signal is defined for use by tool vendors and has no defined connection to the MPC5500 family device.

Table 7. MPC5500 Family MICTOR Connector M38-2C (Second Half, if needed)

MPC5500 Signal	Combined M38-2C						Combined M38-2C	MPC5500 Signal
-	Reserved	-	1	Ground	2	-	Reserved	-
-	Reserved	-	3		4	-	Reserved	-
-	VEN_IO5	In/Out	5		6	-	-	-
-	VEN_IO6	In/Out	7		8	In/Out	PORT15	-
-	VEN_IO7	In/Out	9		10	In/Out	PORT14	-
-	VEN_IO8	In/Out	11		12	In/Out	PORT13	-
-	VEN_IO9	In/Out	13		14	In/Out	PORT12	-
-	TOOL_IO7	In/Out	15		16	In/Out	PORT11	-
-	TOOL_IO6	In/Out	17	Ground	18	In/Out	PORT10	-
-	TOOL_IO5	In/Out	19		20	In/Out	PORT9	-
-	TOOL_IO4	In/Out	21		22	In/Out	PORT8	-
	MDO15	Out	23	Ground	24	In/Out	PORT7	-
	MDO14	Out	25		26	In/Out	PORT6	-
	MDO13	Out	27		28	In/Out	PORT5	-
	MDO12	Out	29		30	In/Out	PORT4	-
MDO11/GPIO82[1]	MDO11	Out	31		32	In/Out	PORT3	-
MDO10/GPIO81[2]	MDO10	Out	33		34	In/Out	PORT2	-
MDO9/GPIO80[3]	MDO9	Out	35		36	In/Out	PORT1	-
MDO8/GPIO79[4]	MDO8	Out	37		38	In/Out	PORT0[5]	-

1. This pin duplicates the signal available on pin 23 of the first MICTOR connector.
2. This pin duplicates the signal available on pin 13 of the first MICTOR connector.
3. This pin duplicates the signal available on pin 8 of the first MICTOR connector.
4. This pin duplicates the signal available on pin 7 of the first MICTOR connector.
5. The MPC5500 family does not support port replacement.

The following figure shows the recommended layout for the M38-2C connectors. This spacing should be adhered to if the two MICTOR connector option is implemented in the target system.

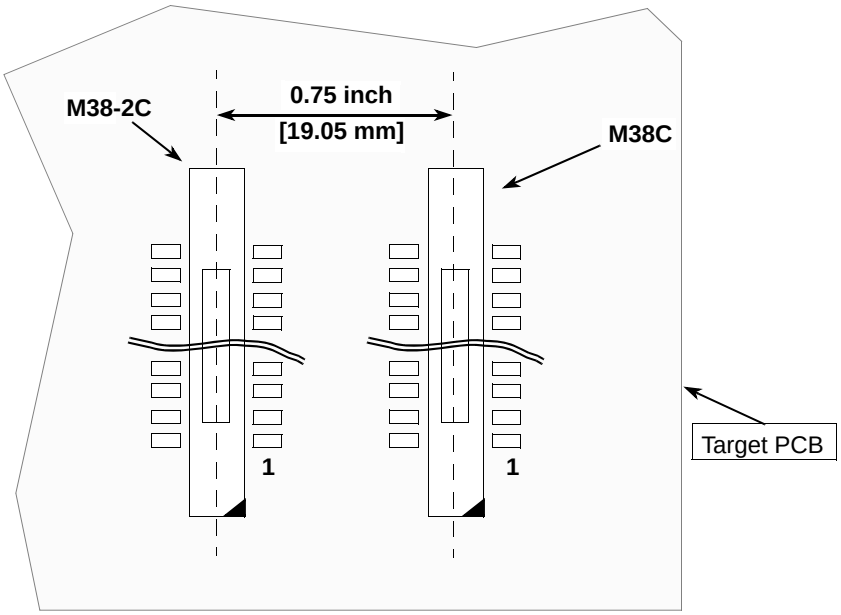


Figure 2. Nexus M38-2C Connector Layout

3.4 Robust Connectors for the MPC5500

The robust connectors are designed to be used in harsh environments such as under the hood of an automobile. The robust connectors listed in Table 3 are manufactured by Glenair, Inc. and feature TwistPin connections for highly reliable contact under all conditions in a Micro-D configuration. The figure below is an illustration of a typical sub-D configuration and a close-up of the TwistPin, and is provided courtesy of Glenair, Inc (http://www.glenair.com).

Figure 3. Details of the Glenair Micro-D TwistPin Connectors

Normally, a 100-pin Glenair Micro-D connector should be defined for a Nexus device with more than eight MDO pins. However, due to board area and cost, Freescale recommends the 51-pin Glenair Micro-D connector be used with the MPC5500. The table below shows the recommended pinout for the robust connector option. It is similar to the R51C connector (which supports up to eight MDO pins), but it redefines the Vendor_IO and Tool_IO pins for use as four additional MDO signals (see Figure 5).

Table 8. MPC5500 Family 51-Pin robust connector definition

	19 MDO0	**1** UBATT
36 GND	**20** GND	**2** UBATT
37 MDO4	**21** MCKO	**3** VSTBY (VALTREF)
38 GND	**22** GND	**4** TOOL_IO0
39 MDO5	**23** $\overline{\text{EVTO}}$	**5** TDO
40 GND	**24** GND	**6** $\overline{\text{RDY}}$
41 MDO6	**25** $\overline{\text{MSEO0}}$	**7** $\overline{\text{RESET}}$
42 GND	**26** MDO9 (VEN_IO0)	**8** VDDE7 (VREF)
43 MDO7	**27** MDO1	**9** $\overline{\text{EVTI}}$
44 GND	**28** GND	**10** GND
45 MDO8 (TOOL_IO2)	**29** MDO2	**11** JCOMP ($\overline{\text{TRST}}$)
46 GND	**30** GND	**12** GND
47 MDO10 (TOOL_IO3)	**31** MDO3	**13** TMS
48 GND	**32** GND	**14** GND
49 MDO11 (VEN_IO1)	**33** TOOL_IO1	**15** TDI
50 GND	**34** GND	**16** GND
51 $\overline{\text{RSTOUT}}$[1]	**35** $\overline{\text{MSEO1}}$	**17** TCK
		18 GND

1. For maximum system flexibility, a 0 Ω resistor option could be used to select between the $\overline{\text{RSTOUT}}$ function and the BootCFG1/$\overline{\text{IRQ3}}$/GPIO212 for this pin for additional debug capabilities.

NOTE

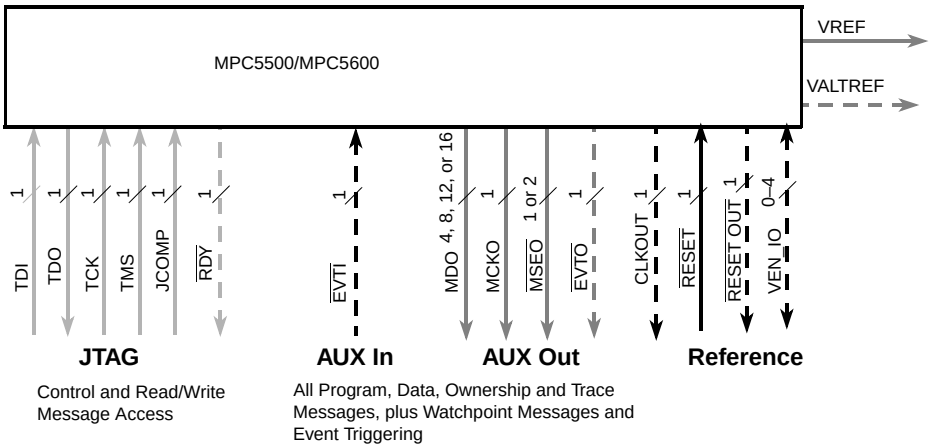
CLKOUT is not available on the robust 51-pin connector. This may limit some capabilities of tools. Tools cannot directly determine the microcontroller's clock speed and must determine the processor speed based on the MCKO frequency and the programmed MCKO divider ratio.

NOTE

If $\overline{\text{RSTOUT}}$ is on pin 51 of the connector, then the capability of the tool triggering an interrupt or DMA access will not be available. However, if BOOTCFG1/$\overline{\text{IRQ3}}$/GPIO212 is on pin 51, then the tool cannot determine exactly when an MCU-initiated reset was asserted.

4 Nexus Signals for the MPC5500 Family

The figure below shows a block diagram of the MPC5500 family Nexus signals. The pins are divided between the JTAG/OnCE port, the auxiliary input port, the auxiliary output port, and reference signals. The $\overline{\text{RDY}}$ pin is grouped with the JTAG pins, but it is used for Nexus block transfers done through the JTAG port (read/write messaging).



Note: Dashed signals are optional in the Nexus standard. Tool_IO pins do not have any connectivity to the MCU.

Figure 4. Pin interface for an MPC5500 Nexus combined JTAG/AUX port

The auxiliary output port can be implemented with four, eight, twelve, or sixteen Message Data Output signals (MDO) and either one or two Message Start/End Output signals ($\overline{\text{MSEO}}$).

Some of the auxiliary output port functions on the MPC5500 family are shared with other pin functions. The following sections show the JTAG and Nexus auxiliary port signals versus the MPC5500 pins and signal descriptions.

4.1 MPC5500 JTAG Signals

The following table shows the JTAG signals for the MPC5500 families.

Table 9. JTAG Connector Signal Descriptions

JTAG Signal	Description
TDO	**JTAG Test Data Output (TDO)**. TDO provides the serial test data output for the on-chip test logic.
TDI	**JTAG Test Data Input (TDI)**. TDI provides the serial test instruction and data input for the on-chip test logic.
TCK	**JTAG Test Clock Input (TCK)**. TCK is the clock input to the JTAG TAP controller and should be a maximum of one-fourth of the system clock frequency.
TMS	**JTAG Test Mode Select Input ($\overline{\text{TMS}}$)**. $\overline{\text{TMS}}$ controls test mode operations for the on-chip test logic for boundary scan and debug access.
$\overline{\text{EVTI}}$	**Nexus Event In ($\overline{\text{EVTI}}$)**. After reset, the $\overline{\text{EVTI}}$ pin is used to initiate program and data trace synchronization messages or generate a breakpoint. If asserted during reset, upon negation of $\overline{\text{RESET}}$, the device will enter debug mode and not begin code execution.
$\overline{\text{RDY}}$	**Nexus Ready Output ($\overline{\text{RDY}}$)**. $\overline{\text{RDY}}$ is an output that indicates to the development tools that the data is ready to be read from or written to the Nexus read/write access registers.
$\overline{\text{RESET}}$	**Reset ($\overline{\text{RESET}}$)**. This is the reset input to the microcontroller. It should be driven by tools to reset the microcontroller. It should be driven by an open drain device. The tool should also monitor this pin to determine if other devices have forced a reset to the microcontroller. It does not reset the JTAG state machine.
JCOMP	**JTAG TAP Controller Enable/ JTAG Compliancy (JCOMP)**. JCOMP is used to enable the TAP controller for communication to the JTAG state machine for boundary scan and for debug access. A high on this pin enables the TAP controller.[1]

JTAG Signal	Description
VREF	**Nexus VREF (VREF)**. Provides a reference for the signal levels of the Nexus device. All input high and low voltages should be referenced to this pin. The Nexus specification defines the input voltages as VIL = 0.3xVREF and VIH = 0.7xVREF. **Note:** This is a reference for the signal level and may not be the actual power supply for the debug pins. See the device definition for the debug pins in the device reference manual.
GND	**Ground**. This is the negative reference (return) for the interface.

1. The IEEE 1149.1 specifies a Test Reset ($\overline{\text{TRST}}$) pin; however, the standard defines that by default the pin should be high (enabled) on the device. Since normal operation of the device does not require the JTAG port to be enabled, Freescale uses the JTAG Compliancy pin (JCOMP), which performs the same basic function, but is asserted (low) by default to disable the JTAG boundary scan/debug port (for normal operation).

The recommended voltage of the JTAG connector is a nominal 3.3 volts for the MPC5500 family devices. For the MPC5600 family and the MPC5510 family, see the definition in each device's reference manual for the recommended voltage. Some devices recommend a nominal 3.3 volts and some devices are 5.0 volts.

4.2 Nexus Auxiliary Port Signals

Some of the auxiliary output port functions on the MPC5500 family are shared with other pin functions. The table below shows the Nexus signal versus the MPC5500/ pins and signal descriptions.

Table 10. Nexus Auxiliary Port Signal Descriptions

Signal Type	Nexus Signal	MPC5500 Signal	Full Signal Name	Description
Auxiliary Output Port	/EVTO	$\overline{\text{EVTO}}$	Nexus Event Out	$\overline{\text{EVTO}}$ is an output that provides timing to a development tool for a single watchpoint or breakpoint occurrence.
	/MSEO[1:0]	$\overline{\text{MSEO}}$[1:0]	Nexus Message Start/End Output	The $\overline{\text{MSEO}}$[1:0] signals indicate when an MDO packet has started and when an MDO packet has ended.
	MCKO	MCKO	Nexus Message Clock Out	MCKO provides timing for the MDO and MSEO[0:1] signals.
	MDO[11:0]	MDO11_GPIO82 MDO10_GPIO81 MDO9_GPIO80 MDO8_GPIO70 MDO7_GPIO78 MDO6_GPIO77 MDO5_GPIO76 MDO4_GPIO75 MDO3 MDO2 MDO1 MDO0	Nexus Message Data Output	MDO[3:0] provide the message data out of the auxiliary output port in the reduced-port configuration. The full-port configuration uses MDO[11:0].

Signal Type	Nexus Signal	MPC5500 Signal	Full Signal Name	Description
Auxiliary Input Port	/EVTI	$\overline{\text{EVTI}}$	Nexus Event Input	After reset, the $\overline{\text{EVTI}}$ pin is used to initiate program and data trace synchronization messages or generate a breakpoint. If asserted during reset, upon negation of $\overline{\text{RESET}}$, the device will enter debug mode and not begin code execution.

4.3 Other Nexus Connector Signals

In addition to the JTAG and Nexus auxiliary port signals, there are additional signals included on the connector for reference and for use by tools or other purposes. The following table shows the definition of these signals.

Table 11. Other Nexus Connector Signal Descriptions

Signal Type	Nexus Signal	MPC5500 Signal	Full Signal Name	Description
Vendor IO	VEN_IO2	BOOTCFG1_$\overline{\text{IRQ3}}$_GPIO212	Vendor Defined IO 2	Nexus Vendor_IO0 (VEN_IO2). The MPC5500 signal BOOTCFG1_$\overline{\text{IRQ3}}$_GPIO212 gives the Nexus tool access to $\overline{\text{IRQ3}}$. This signal can either generate an interrupt to the interrupt controller or can be used to initiate a DMA transfer via DMA channel 3. This allows the Nexus port to request a DMA transfer. It can also invoke the DMA gather data feature. The BOOTCFG1 function can be used during reset to change the default boot mode of the device. On the robust connector, this pin is used for $\overline{\text{RSTOUT}}$.

Signal Type	Nexus Signal	MPC5500 Signal	Full Signal Name	Description
Tool IO	TOOL_IO0		Tool Defined IO 0	Nexus Tool_IO0 (TOOL_BR). This pin is reserved for use as a bus request signal for arbitration between different tools on the Nexus bus. It has no connection to the MPC5500 device. **Note:** Tool arbitration is an informal extension of the Nexus IEEE-ISTO 5001-2003 specification and has not been approved by the Nexus Consortium.
	TOOL_IO1		Tool Defined IO 1	Nexus Tool_IO1 (TOOL_BG). This pin is reserved for use as a bus grant signal for arbitration between different tools on the Nexus bus. It has no connection to the MPC5500 device. **Note:** Tool arbitration is an informal extension of the Nexus IEEE-ISTO 5001-2003 specification and has not been approved by the Nexus Consortium.
	TOOL_IO3		Tool Defined IO 3	Nexus Tool_IO3 (Tool_IO3) This pin is defined on the MICTOR 38 pin connector as $\overline{\text{RSTOUT}}$ of the MPC5500. This allows a tool to monitor when the MPC5500 has caused an internal reset. **Note:** The MCU can drive $\overline{\text{RSTOUT}}$ low via software to reset external devices, but not cause a reset of the MCU. This pin is defined for use as MDO10 on the robust connector option.
	TOOL_IO[4:7]		Tool Defined IO [4:7]	Tool_IO pins (TOOL_IO[4:7]). Designated for Nexus tool use. At this time, there are no defined connections from these pins to the MPC5500 family device. They may be used for communication between multiple tools on the Nexus connector. These pins are not available on the single MICTOR connector option or on the 51-pin robust connector.

Signal Type	Nexus Signal	MPC5500 Signal	Full Signal Name	Description
Reference Signals	/RESET	$\overline{\text{RESET}}$	/Reset	Reset (/RESET). This is the reset input to the MPC5500 microcontroller and should be driven by tools to reset the microcontroller. It should be driven by an open drain device. The tool should also monitor this pin to determine if other devices have forced a reset to the microcontroller. It does not reset the JTAG state machine.
	CLOCKOUT	CLKOUT	System Clock Output	Clock Out (CLKOUT). Clock out provides a reference clock for determining the system clock frequency. The user can disable this signal.
	VREF	VDDE7	Signal Power Reference	Nexus VREF (VREF). Provides a reference for the signal levels of the Nexus device. All input high and low voltages should be referenced to this pin. The Nexus specification defines the input voltages as VIL = 0.3 × VREF and VIH = 0.7 × VREF.
	VALTREF	VSTBY	Alternate Reference	Nexus VALTREF (VSTBY). Provides a monitor for the backup power supply for theMPC5500 family internal SRAM. This pin is monitored by the Nexus tool to determine if power is lost to the standby supply of the system. This pin should be grounded if standby operation is not allowed in a particular system.
	GND	GND	Ground Reference	Ground. This is the return (ground) reference point for the Nexus connectors. Many ground signals are included to provide shielding for the high-speed Nexus signals. In the MICTOR connector, the center pins should be grounded.
	UBATT	12V	Unregulated Battery Voltage	Battery/unregulated module voltage. This is a supply voltage output from the target system and should have reverse-bias protection. This allows tools to have circuitry powered by the target system. The IEEE-ISTO 5001-2003 defines this as a 5 to 20 volt supply of up to 300 mA of current.

5 Nexus Connector Voltages for the MPC5500

All of the Nexus pins on the MPC5500 family are supplied by VDDE7. VDDE7 can be used with any voltage from 1.62 to 3.6 volts. The user determines the voltage level of the Nexus pins. See the following table for the common voltages that could be used. Most commonly, this should be set to 3.3 V, which is also the most common bus voltage used on the MPC5500 family.

Table 12. Nexus Port Voltages Options

System Bus Voltage	Recommended Nexus Voltage
1.8 volts	1.8 volts
2.5 volts	2.5 volts
3.3 volts	3.3 volts[1]

1. A 3.3 V supply is the preferred option. 1.8 V and 2.5 V are not recommended, although they are allowed within the specification.

On the MPC5500, the $\overline{\text{RESET}}$ input should be driven by devices with an open drain. The target system is required to have a pullup device to the device power supply. This can be up to 5 V for the MPC5500. Care must be taken to insure that all circuitry connected to the $\overline{\text{RESET}}$ signal is 5 V tolerant. Tools need to be able to drive $\overline{\text{RESET}}$, but they also need to sense $\overline{\text{RESET}}$ to determine if another device in the target system has requested the MCU be reset.

NOTE

$\overline{\text{RSTOUT}}$ is negated 2400 clocks after $\overline{\text{RESET}}$ is negated or the system PLL is locked (whichever occurs last) in normal clock operating modes (crystal or external reference). $\overline{\text{RSTOUT}}$ is negated 16000 clocks after $\overline{\text{RESET}}$ is negated or the PLL is locked in bypass or 1:1 clock modes.

NOTE

$\overline{\text{RESET}}$ and $\overline{\text{RSTOUT}}$ are powered by VDDEH6. VDDEH6 can have a nominal voltage of 3.3 or 5.0 volts. Other than the UBATT pins, the IEEE-ISTO 5001-2003 standard does not allow voltages greater than VREF. If VDDEH6 is connected to 5.0 volts, then the tool must have protection for the $\overline{\text{RESET}}$ and $\overline{\text{RSTOUT}}$ signals.

NOTE

Consult the hardware specifications of the debug interface to insure that the voltages are all compatible with the module design.

6 JTAG Clock Speed for MPC5500

The JTAG/OnCE interface on the MPC5500 devices has a maximum operating frequency of one-fourth of the system frequency. Immediately after reset, the MCU system frequency will be 1.5 times the crystal input (or input frequency). The JTAG/OnCE clock frequency should therefore be set to a maximum of 3 MHz (assuming an 8-MHz crystal, the operating frequency will be 12 MHz) until the tool detects that the user has changed the operating frequency of the device. Clock Out (CLKOUT) may be monitored by the tool initially to determine the system frequency. This clock should not be a requirement for the normal operation of the Nexus interface since the user has the option of disabling this clock.

NOTE

CLKOUT is not available on the robust connector.

7 Minimum External Circuitry

In general, other than the connector, no additional circuitry is required for the Nexus/JTAG debug circuitry. The MPC5500 devices include internal pull devices that insure the pins remain in a safe state; however, if there is additional circuitry connected to the Nexus/JTAG pins or long traces that could affected by other signals (due to crosstalk from high current or high speed signals), a minimum number of external pull resistors can be added to insure proper operation under all conditions.

Table 13. Optional Minimum Debug Port External Resistors

Nexus/JTAG Signal	Resistor direction and value	Description
JCOMP	10K Ω pull down	Holds debug port in reset and prevents any debug commands from interfering with the normal operation of the MCU.
$\overline{\text{RESET}}$	4.7K Ω pull up	The $\overline{\text{RESET}}$ input should be driven from an open collector output; therefore, it requires a pullup resistor for the MCU.
TD/WDT[1]	10K Ω pull down	With no tool attached, this signal should be held low and may or may not be connected to a pin of the MCU, depending on the system definition.

Nexus/JTAG Signal	Resistor direction and value	Description
$\overline{\text{EVTI}}$	10K Ω pull up	A pullup prevents debug mode from being forced after reset if debug mode is enabled (JCOMP = high). It also prevents breakpoints from being forced if debug mode is enabled. **Note:** In almost all situations, a resistor is not required on this signal.

1. This is an optional signal and is not actually required for the MCU.

In addition to the pull resistors, some systems may want to use buffers between the Nexus/JTAG connector inputs and the MCU. This will prevent over-voltage conditions from causing damage to the MCU signals. Normal systems should not require this circuitry, but it is helpful in systems that can be exposed to improper connections that provide voltages that are outside the operating conditions of the MCU. A common circuit to use is the Texas Instruments SN74CBTLV3861[2]. This device is a bus switch that implements a bidirectional interface between two terminals with less than 5 Ω of resistance. It should be powered by the same supply that powers the debug port. The device enable should be connected to ground for the interface to be enabled whenever the debug port on the MCU is powered. This circuit provides a high impedance to the tool when the debug port is powered off.

NOTE

It is recommended that at least the reduced port configuration Nexus signals be made available (somewhere) on production boards. This facilitates debugging of new boards and analysis of errors in software, even on boards that have restricted space and normally provide a JTAG-only connection. If the Nexus signals are available on the production board, an adapter could be built to provide a Nexus connection on boards that do not have a complete footprint for one of the standard Nexus connectors. Likewise, the JTAG connector does not have to be populated on production boards and could even utilize a smaller connector footprint that could be used with an adapter to the standard debug connections.

8 Nexus Port Configuration Register

The Nexus Port Controller Port Configuration Register (NPC_PCR) controls the operation and configuration of the Nexus debug interface for the entire device. This register overrides any control settings (MCKO_DIV, FPM, and MCKO_EN) that may be available in individual Nexus client (NDEDI/NSEDI, NZ6C3/NZ3C3, NXDM, and NXFR) Debug Control (DC) or Port Configuration Registers (PCR).

During $\overline{\text{RESET}}$, the MPC5500 will assert (high) MDO0 until the PLL is locked after the initial internal power-on reset. Once MDO0 is negated (low), access to the JTAG controller is enabled and registers can be written by tools. The nexus port should be enabled in the NPC_PCR while the processor is in reset. After the port has been configured, the NPC will transmit a device identification (DID) message through the Nexus auxiliary output port.

NOTE

The numbering of the bits in the NPC_PCR are set per the IEEE-ISTO 5001 standard, as it is accessed only by tools and the CPU cannot write to any of the bits in the register. All bits are controlled by the external tool.

2. SN74CBTLV3861-Q1 is automotive qualified if required.

Table 14. NPC Port Configuration Register (NPC_PCR)

NPC Register Index 127		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	FPM	MCKO_GT	MCKO_EN	MCKO_DIV			EVT_EN[1]	DDR_EN[2]	NEXCFG[3]	0	0	0	0	0	0	0
	W																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	LP_DBG[4]	0	0	0	0	0	LP2_SYN[4]	LP1_SYN[4]	0	0	0	0	0	0	0	PSTAT_EN[5]
	W																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. This bit is not implemented on all devices. It is not implemented on any of the MPC5500 devices.
2. The Double Data Rate feature does not conform to the Nexus 5001 standard and may not be supported.
3. Not supported on all devices. This use of this bit is determined by the MCU definition.
4. Not supported on devices that do not support true low-power modes (where some power supplies can be removed from the MCU).
5. PSTAT is an internal Freescale monitor mechanism and its use is not supported. This bit should always be set to 0.

8.1 MPC5500 Nexus Port Size (NPC_PCR[FPM])

The configuration of the full-port mode or reduced-port mode is selected via the JTAG/OnCE port by setting of the Full Port Mode (FPM) bit in the Nexus Port Controller port configuration register (NPC_PCR). Setting NPC_PCR[FPM] selects full-port mode. Clearing NPC_PCR[FPM] selects reduced-port mode. In RPM, MDO[4:11] are defined to the GPIO function; FPM automatically disables the GPIO function.

Table 15. Nexus Port Controller Full Port Mode bit Settings

Set-ting	Description
0b0	Reduced port mode is selected.
0b1	Full port mode is selected.

Table 16. MPC5500 Family Nexus Port Sizes Across Packages

Device	Package	Reduced Port Size	Full Port Size
MPC5534	208 MAPBGA	4	NA[1]
	324 PBGA	4	12
MPC5553	324 PBGA	4	12
	416 PBGA	4	12
MPC5554	416 PBGA	4	12
MPC5561	324 PBGA	4	12
MPC5565	324 PBGA	4	12
MPC5566	416 PBGA	4	12

Device	Package	Reduced Port Size	Full Port Size
MPC5567	324 PBGA	4	12
	416 PBGA	4	12

1. Not available.

8.2 Nexus Clock Speed (NPC_PCR[MCKO_DIV])

The Nexus auxiliary clock prescaler should be programmed in the NPC_PCR. For systems with a maximum operating frequency of over 80 MHz, the auxiliary clock prescaler should be set to divide by 2 or more. MCKO_DIV should not be changed after the auxiliary output port is enabled (NPC_PCR[MCKO_EN]). The ability to run the MCKO at the same frequency as the system frequency is available on some devices if the system frequency is 80 MHz or less. In all cases, see the device electrical specification to determine if the signals have any other speed limitations.

Table 17. Nexus MCKO Clock Divider Options

MCKO_DIV	MCKO Frequency
0b000	System Frequency[1]
0b001	System frequency ÷ 2
0b010	Reserved
0b011	System frequency ÷ 4
0b100	Reserved
0b101	Reserved
0b110	Reserved
0b111	System frequency ÷ 8

1. The 1X clock mode is not supported on all devices. In addition, MCKO is limited to 80 MHz maximum on all devices.

The speed of the Nexus port limits the amount of trace information that will be available to tools. Enabling too much trace will cause overflows of the trace buffers included on the device. In general, trace of the core program execution will not cause overflows with the MCKO frequency set to divide by two. A divide of four may also support full instruction trace if branch history mode is used. Enabling data trace may cause overruns if too much information is requested. Care should be taken when enabling data trace to limit the information that should be traced. Watchpoints can be used to enable trace upon an event occurrence and then disabled by a second event to limit the data trace bandwidth requirements.

8.3 Nexus Clock Gating (NPC_PCR[MCKO_GT])

The Nexus auxiliary output clock (MCKO) can be gated off when the MPC5500 is not transmitting trace information. Setting NPC_PCR[MCKO_GT] gates the auxiliary clock off when auxiliary messages are not being transmitted. If this bit is cleared, then the Nexus auxiliary clock is always present when the Nexus auxiliary port is enabled.

NOTE

Tools may require that the MCKO not be gated. Check with your tool vendor to determine if the tool can work with a gated MCKO.

8.4 Unsupported Bits in the NPC_PCR

None of the MPC5500 devices covered by this application note support low power modes. Therefore the Low Power Debug Enable (LP_DBG) and the Low Power Mode Synchronization (LP2_SYN and LP1_SYN) bits in the NPC_PCR are not supported. In addition, none of the devices implement or use the Nexus Configuration Select (NEXCFG) bit in the NPC_PCR.

The Event Enable (EVT_EN - for enabling/disabling of the $\overline{\text{EVTI}}$ and $\overline{\text{EVTO}}$.

The Dual Data Rate (DDR) enables a non-standard signal format that clocks two bits per clock cycle. The MDO bits are valid approximately halfway between the falling and rising edges of MCKO and then again between the rising and falling edges of MCKO. This convention is not covered by the IEEE-ISTO5001 standard and is not expected to be supported by the standard.

The Nexus Configuration (NEXCFG) bit is not supported on any of the MPC5500 devices.

The Processor Status Enable (PSTAT_EN) is not supported on any devices. This is an internal mode for Freescale use only.

Appendix A MPC5500 Available Debug Adapters

A number of adapters are available to ease the task of having full debug capabilities available with a minimum board footprint impact to the target system. The following table lists adapters that are available from Freescale and third-party manufacturers.

Table A-1. Available Debug Connector Adapters

Part Number	Description	Manufacturer
PE1906	14-pin JTAG debug tool to MICTOR 38-pin target board connector adapter	P&E Microcomputer Systems
LA-3723 CON-JTAG14-MICTOR	14-pin JTAG debug tool to MICTOR 38-pin target board connector adapter	Lauterbach GmbH
LA-3725 CON-MIC38-J14-5500	38-pin MICTOR debug trace tool to 14-pin JTAG target board connector adapter	Lauterbach GmbH
LFVDBJR	VertiCal Debug Board with JTAG / MICTOR Connector Interface, Rigid PCB (Lead Free) VertiCal Lead Free Debug Interface board which consists of a VertiCal Male interface on the secondary side and a VertiCal Female interface on the primary side, allowing the user to connect to any VertiCal-based development tool. Debug tool interface is facilitated by both a 2x7 pin JTAG header and MICTOR connector interface on the primary side. The LFVDBJR is implemented with a rigid PCB.[1]	Freescale Semiconductor
LFVDBJF	VertiCal Debug Board with JTAG / MICTOR Interface, Flexible PCB (Lead Free) VertiCal Lead Free Debug Interface board which consists of a VertiCal Male interface on the secondary side and a VertiCal Female interface on the primary side, allowing the user to connect to any VertiCal-based development tool. Debug tool interface is facilitated by both a 2x7 pin JTAG header and MICTOR connector interface on the primary side. The LFVDBJF is implemented with a flexible PCB.[1]	Freescale Semiconductor
LFVDBGR[2]	VertiCal Debug Board with Robust Nexus 51-pin Connector Interface, Rigid PCB (Lead Free) VertiCal Lead Free Debug Interface board which consists of a VertiCal Male interface on the secondary side and a VertiCal Female interface on the primary side, allowing the user to connect to any VertiCal-based development tool. Debug tool interface is facilitated by a Robust Nexus connector interface on the primary side. The LFVDBGR is implemented with a rigid PCB.[1]	Freescale Semiconductor

Part Number	Description	Manufacturer
LFVDBGF[2]	VertiCal Debug Board with Robust Nexus 51-pin Connector Interface, Flexible PCB (Lead Free) VertiCal Lead Free Debug Interface board which consists of a VertiCal Male interface on the secondary side and a VertiCal Female interface on the primary side, allowing the user to connect to any VertiCal-based development tool. Debug tool interface is facilitated by a Robust Nexus 51-pin connector interface on the secondary side. The LFVDBGF is implemented with a Flexible PCB.[1]	Freescale Semiconductor

1. Works with LFVCALSRAM or any tool with a Male Vertical interface.
2. This adapter only supports a maximum of 12 MDO signals.

Additional information is available at the web links in the following table.

Table A-2. Adapter manufacturer contact information

Company	Web Link
Freescale Semiconductor	http://www.freescale.com
Lauterbach GmbH	http://www.lauterbach.com
P&E Microcomputer Systems	http://www.pemicro.com

Appendix B Core Memory Access versus Nexus Read/Write Access

There seems to be much confusion about accessing memory via either the e200z*x* core or via Nexus Read/Write Access. The Nexus Read/Write Access feature is sometimes referred to as the DMA access. This itself causes confusion because the e200zx devices implement an eDMA module, and on the high end devices any transfers via the eDMA module can also be traced using the Nexus data trace features of the device.

Core access (by a debugger) is performed by forcing instructions into the e200z*x* core. Accesses to memory, therefore, are through the cache and through the Memory Management Unit (MMU). Memory accesses use the virtual addresses. (Virtual addresses are translated by the MMU to the physical address.)

Nexus Read/Write Accesses do not access memory through the core. Nexus R/W accesses bypass both the cache and the MMU. All accesses utilize physical addresses. Nexus R/W access can be performed while the core is running. This allows a debugger to perform real-time access of variables without stopping the core.

NOTE

On devices with cache, Nexus R/W Access will not see changes made by the core to variables or data in the cache, until the cache contents are flushed to memory. Conversely, changes to memory by Nexus R/W Access may not be seen by the core, if that area of memory is cached by the CPU.

Nexus Read/Write access provides a much faster and much more streamlined transfer of blocks of data. A tool initializes Read/Write start address (RWA), the word transfer size (RWCS[SZ]), and the number of the transfers (RWCS[CNT]) to be performed.

The two different paths from the debugger to memory are shown in the following figure. This MCU architecture shown in this figure is based on the MPC5567.

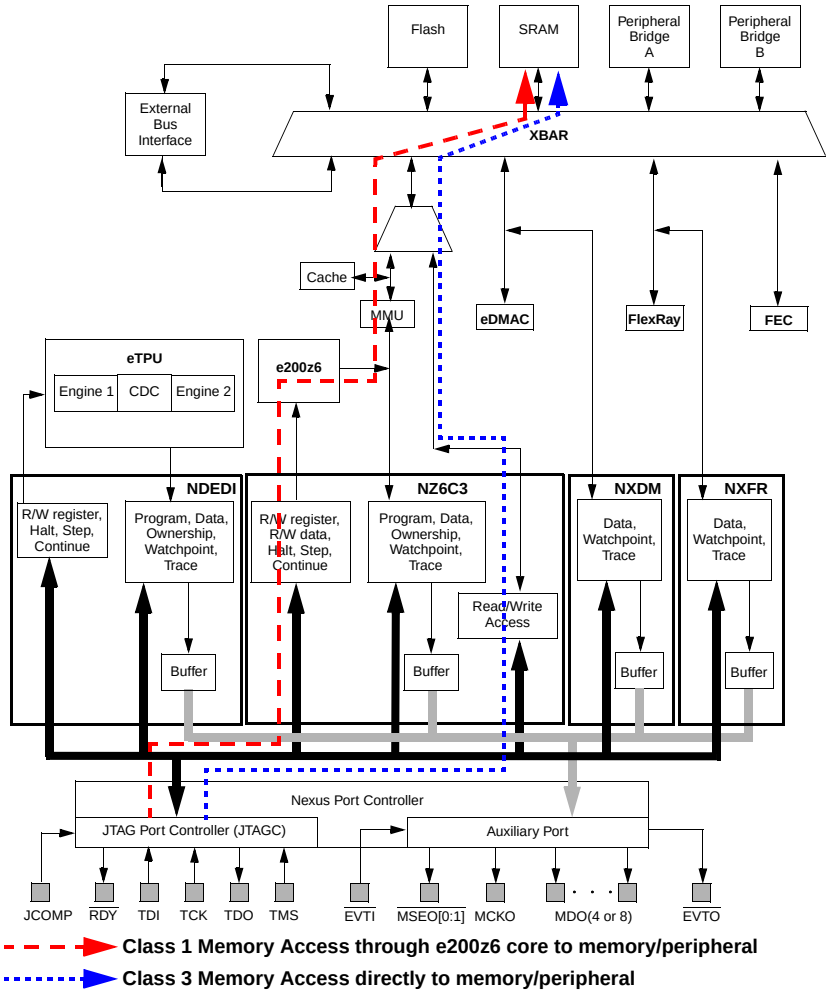


Figure B-1. Core (Class 1) versus Nexus Class 3 Memory Access

The sequence for performing a Nexus read/write access is as follows.

1. Enable access to e200z*x* Nexus.
2. Write address to Nexus Read/Write Access Address register.
3. Write data to Nexus Read/Write Access Data register.
4. Write to Nexus Read/Write Control/Status register to set the size and length of the transfer.
5. Poll the $\overline{\text{RDY}}$ pin (if available) to determine if a single access (when performing a block read or write operation) is complete. If the $\overline{\text{RDY}}$ pin is not available, the Read/Write Access Control register should be polled for the status of the operation.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN2614
Rev. D, 18/March/2010