Freescale Semiconductor
Application Note

Document Number: AN4591
Rev. 2, August 2015

Lauterbach MPC57xx Nexus Trace Tools

Including both parallel and serial (Aurora) trace protocols

by: Randy Dees

Contents

1 Introduction

This application note describes the different trace solutions that are available for the MPC57xx family of devices. Different trace strategies are currently defined for the MPC57xx devices depending on the Nexus trace requirements of the end system. Some of the MPC57xx devices utilize the Nexus high-speed serial trace interface, based on the Xilinx® Aurora protocol, that is included in the IEEE-ISTO 5001-2012™ Nexus standard. Other devices continue to use the existing Nexus parallel trace interface (MPC55xx and MPC56xx). In addition, some of these devices also supports trace to an internal trace memory. Not all devices in the MPC57xx family support the high-speed Nexus Aurora trace or the trace-to-memory features.

This application note lists the Lauterbach tool options that support the Nexus high-speed serial (Aurora) trace protocol on the MPC57xx devices. It also shows the tools that are available for the Nexus parallel trace and the Nexus trace-to-memory feature for devices that support these options.

This application note also lists the type of trace support included on each of the MPC57xx devices. The table below summarizes the trace and basic debug support for the MPC57xx devices.

Table 1. MPC57xx family trace support summary

Device	IEEE 1149.7 support	Nexus parallel trace[1]	Nexus serial (Aurora) trace[2]	Nexus trace to memory[3]	Highest Nexus class support[4]
MPC574xB/MPC574xD	Yes	Yes[5] / 12 or 16	No	No	Class 3+
MPC574xG/MPC574xC	Yes	Yes[5] / 12 or 16	No	No	Class 3+
MPC574xP	No	Yes / 4[6]	Yes / 2[7]	No	Class 3+
MPC5746R	Yes	No	No	Yes / 16 KB	Class 3+
MPC5746R emulation device	Yes	No	Yes/4	Yes / 1 MB	Class 3+
MPC577xK	Yes	No[8] / 16	Yes / 4	No	Class 3+
MPC5777C	Yes	Yes / 12 or 16	No	No	Class 3+
MPC5777M	Yes	No	No	Yes / 16 KB	Class 3+
MPC5777M emulation device	Yes	No	Yes / 4	Yes / 2 MB	Class 3+

1. Number of (parallel) Message Data Outputs is shown if supported.
2. Maximum number of Aurora lanes is shown, if supported. Devices that support 4-lanes also support 2-lane operation.
3. Maximum trace memory. This memory can be split between trace use and overlay use.
4. For some devices, all Nexus clients may not implement the full Nexus Class 3+ features. Some clients may only support Class 1 or Class 2+. See the complete device documentation for the Nexus class supported for each of the clients. In this table, if Class 3+ is listed, then at least one Nexus client supports Nexus Class 3+ features.
5. 324 MAPBGA package only
6. Available in all packages.
7. 257 MAPBGA package only
8. Internally, 16 parallel Nexus Message Data Outputs are available, however, these signals are not available in the standard 356 Molded Array Process Ball Grid Array (MAPBGA) device package.

The remainder of this application note provides an overview of the different trace interfaces (Nexus parallel, Nexus high-speed serial, or Aurora, and trace to memory) and the tools available from Lauterbach to support each of these Nexus interfaces.

NOTE

For additional overview information about the general Nexus features, see a description of the Nexus classes in Appendix "Nexus Class Definitions" and types of Nexus trace messages in Appendix "Nexus trace overview".

NOTE

Run control operations are controlled by the JTAG Controller of the Nexus Debug Interface (NDI). The trace is information is output through the Nexus auxiliary port of the NDI. In this document, the Nexus auxiliary port is referenced as the Nexus parallel trace interface, the Nexus high-speed serial (Aurora) trace interface, or the Nexus trace-to-memory interface.

2 Nexus Trace Options Overview

The Nexus high-speed serial (Aurora) interface, as defined for the MPC57xx devices, supports a serial interface of up to four lanes at 1.25 Gbit/s, providing a much higher bandwidth of trace data with fewer pins required on the device.[1]

1. Some devices only support two lanes, but other devices support both two-lane and four-lane configurations.

Additionally, some devices in the MPC57xx family support the Nexus parallel trace interface as described in both the IEEE-ISTO 5001-2003™ and the IEEE-ISTO 5001-2012™ standards. The Nexus parallel trace interface can support up to 16 parallel Message Data Output (MDO) pins and is limited to a maximum of approximately 80 MHz.[2] One lane of Nexus serial (Aurora) trace is approximately equal to a 16-bit MDO Nexus parallel trace solution.

Some devices in the MPC57xx family support trace to memory. This allows a limited amount of trace data to be collected on-chip and decoded by an external tool. This requires only a JTAG interface connection and the addition of a trace license for the external debugger.

All MPC57xx devices support the IEEE 1149.1 interface for run control and initialization of the Nexus trace modules of the device. In addition, some devices also support the IEEE 1149.7 interface, which supports the 2-wire JTAG mode. This mode allows the Test Data Input (TDI) and Test Data Output (TDO) JTAG signals to be eliminated from the debug interface and frees the signals for other use in the system.

The table below shows the MPC57xx blocks that generate the JTAG and Nexus signals described in the following sections.

Table 2. Nexus trace and debug blocks shown in block diagrams

Name	Description
JTAG Port Controller (JTAGC)	Acts as a gateway to the JTAG interfaces of other debug clients located inside the MCU.
IEEE 1149.7 Interface (CJTAG)	Powers up in the IEEE 1149.1 compatible mode, but it can be enabled by the debugger to support features available in the IEEE-1149.7 specification. It can also provide a conversion of the 2-wire 1149.7 interface to the full 4-wire standard JTAG interface.
Nexus Port Controller (NPC)[1]	Acts as a switch between the different Nexus clients located inside the MCU and an external parallel trace interface.
Nexus Aurora Router (NAR)[1]	Buffers Nexus trace data from Nexus clients and routes the trace data either to internal memory or to an external high-speed Nexus Aurora physical interface.
Nexus Aurora Link (NAL)	Provides the Aurora 8b10b protocol encoding of the parallel Nexus messages. The NAL splits the data into the defined number of lanes and provides the 10-bit formatted symbols to the NAP.
Nexus Aurora Physical Interface (NAP)	Provides the parallel to serial conversion from the internal system clock frequency to the Nexus Aurora clock frequency.

1. Devices will implement either an NPC or a NAR, both cannot be implemented on a single device.

2.1 Nexus parallel trace interface

Some MPC57xx support the Nexus parallel trace interface, which is also supported in the existing generation of Nexus Automotive devices (MPC55xx and MPC56xx families), as defined in the IEEE-ISTO 5001-1999 and IEEE-ISTO 5001-2003 standards. The Nexus parallel trace solution includes a parallel interface with Message Data Outputs (MDO), either one or two Message Start/End Output(s) (MSEO), and a Message Data Clock (MCKO). The number of MDO signals depends on the definition of the device (port widths of four, eight, twelve, or sixteen are defined on MPC55xx, MPC56xx, and MPC57xx devices). A JTAG interface (IEEE 1149.1 and, in some cases, IEEE 1149.7) is also included for controlling the initialization of the trace modules. This figure below shows a block diagram of a typical Nexus parallel trace interface showing the signals available external to the device.

2. The Nexus parallel trace is supported by most of the devices in the MPC5500 and MPC5600 families. However, some devices may limit the maximum speed of the Nexus parallel trace interface to a lower frequency.

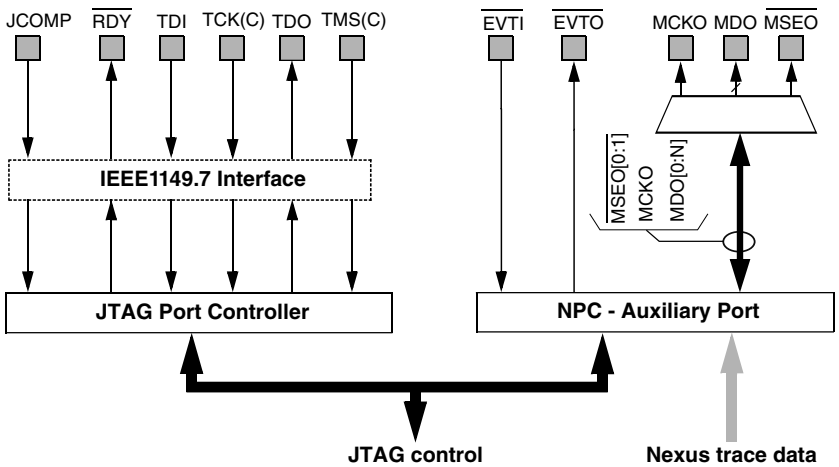

Figure 1. Parallel trace signals

2.2 Hybrid Nexus trace (parallel and high-speed serial) interface

Some devices in the MPC57xx family support the Nexus parallel trace and the Nexus high-speed serial over a Xilinx Aurora physical interface. The Nexus parallel solution allows a limited amount of trace data and the Nexus high-speed serial interface is available when a larger amount of trace information is required. These may be segregated by package type. Smaller packages may only support the Nexus parallel interface, whereas the larger package supports the Nexus high-speed serial interface and the Nexus parallel interface. The parallel interface includes Message Data Outputs (MDO), either one or two Message Start/End Output(s) (MSEO), and a Message Data Clock (MCKO). The Nexus high-speed serial interface uses low-voltage differential signals (LVDS) and includes multiple Aurora Transmit lanes and an LVDS clock input lane.[3] A JTAG interface (either IEEE 1149.1 and in some cases, IEEE 1149.7) is also included for controlling the initialization of the trace modules. The figure below shows a high level view of the Nexus blocks with the externally available signals.

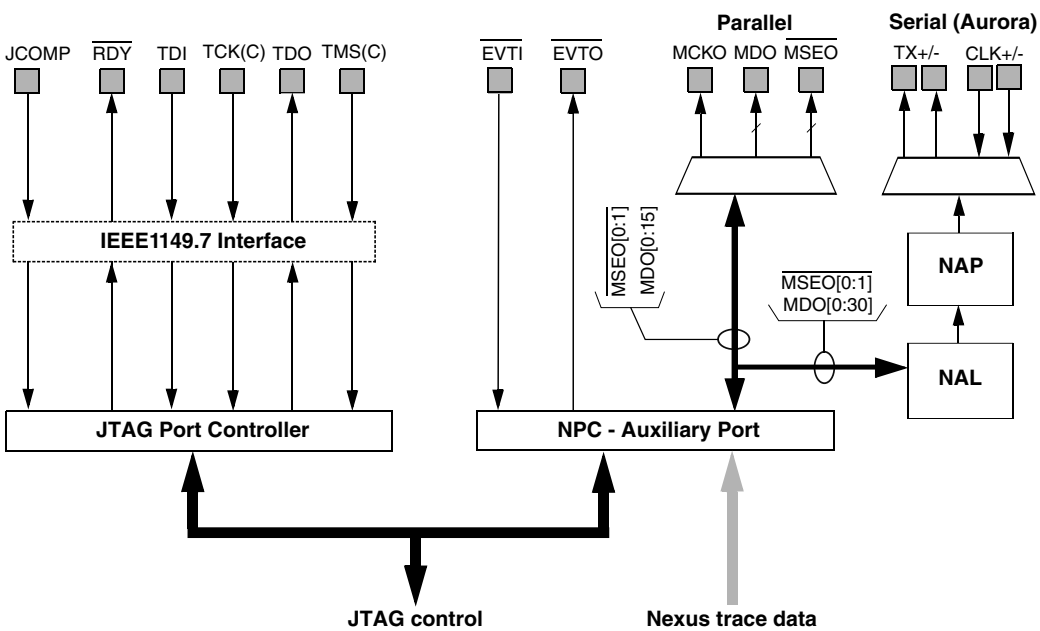

Figure 2. Hybrid parallel/serial trace signals

3. A lane consists of two signals, a positive signal and a negative signal.

2.3 Nexus high-speed serial trace interface

To allow for the largest bandwidth of trace information to be transmitted from the microcontroller (MCU), a new Nexus high-speed serial trace option was added to the IEEE-ISTO 5001-2012 standard. This interface formats the parallel Nexus trace data into a serial Xilinx Aurora physical interface for transmission from the MCU. The high-speed serial Nexus interface uses low-voltage differential signals and includes multiple Aurora transmit lanes and an LVDS clock input lane.[4] A JTAG interface (IEEE 1149.1 and, in some cases, IEEE 1149.7) is also included for controlling the initialization of the trace modules. The figure below shows a high level block diagram of the Nexus serial interface with the externally visible signals.

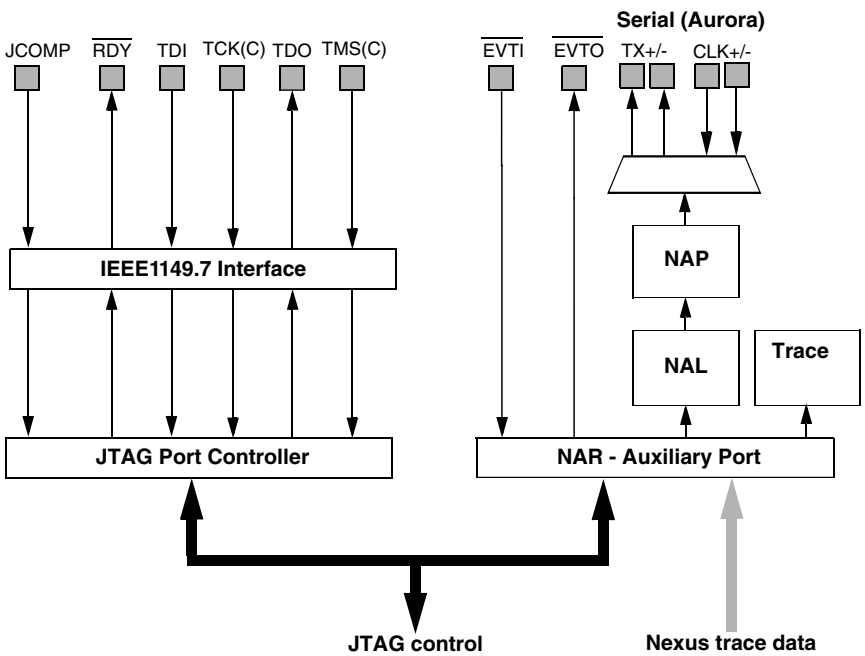

Figure 3. Nexus serial (Aurora) trace signals

3 Lauterbach TRACE32 MPC57xx Nexus Aurora Trace Tools

Lauterbach has implemented several options for supporting the high-speed serial trace that is provided on devices that implement the Nexus Aurora interface. The first solution is intended for high performance to deal with more trace information and maximum future compatibility. Below is a list of the components that must be ordered for a complete Lauterbach TRACE32 Nexus Aurora tool. There are currently three options for trace memory size (1 GB, 2 GB, or 4 GB) in the trace interface.

Table 3. Complete Nexus Serial (Aurora) trace solution for the MPC57xx devices

Description	Order/Part Number	Name	Requirement
Host interface[1]	LA-7699[2]	PowerDebug II	Only one of these options is required.
	LA-3505	PowerDebug PRO	

Table continues on the next page...

4. A lane consists of two signals, a positive signal and a negative signal.

Table 3. Complete Nexus Serial (Aurora) trace solution for the MPC57xx devices (continued)

Description	Order/Part Number	Name	Requirement
Trace module[1]	LA-7692	PowerTrace II (1 GB)	Only one of these options is required.
	LA-7693	PowerTrace II (2 GB)	
	LA-7694	PowerTrace II (4 GB)	
Debug cable (JTAG interface)	LA-7753[2]	JTAG Debug Cable for MPC5500/ MPC5600 (OnCE)	Only one of these two options is required.
	LA-3736	Universal Automotive JTAG Debug Cable for MPC5xxx (AUTO26)[3]	
Aurora trace preprocessor	LA-3911	MPC57xx Aurora Preprocessor	One required.
JTAG and Nexus Aurora adapter	LA-3871	OnCE JTAG or AUTO26 + Nexus Aurora adapter (includes Samtec 34-pin flex extension cable)	One required.
Multi-core license[4]	LA-7960X	Software license for multi-core operation	One required.
Software warranty	LA-8017	Additional year of software maintenance	Not required for first year, but needed for second year

1. Can also be used for parallel Nexus trace with the LA-7630 preprocessor.
2. This interface is out of production, but is still supported by Lauterbach.
3. The Automotive JTAG Debug cable supports additional signals that are not available on the standard (LA-7753) MPC5xxx debug cable ($\overline{\text{RSTOUT}}$, $\overline{\text{EVTI}}$, $\overline{\text{EVTO}}$, and external Watchdog Disable).
4. Required for devices with multiple cores. Debug of the enhanced Timing Processing Unit (eTPU), the Generic Timer Module (GTM), or Signal Processing Toolbox (SPT) in addition to the Power Architecture® core, does not require a multi-core license.

The second solution is targeted to lower end performance or for existing installations that already include the smaller, more limited trace interface.

Table 4. Low-end Nexus Serial (Aurora) trace solution for the MPC57xx devices

Description	Order/Part Number	Name	Requirement
Host Interface and Trace module[1]	LA-7707[2]	PowerTrace 256 MB[3, 4]	Only one of these two options is required.
	LA-7690[2]	PowerTrace 512 MB[3, 4]	
Debug cable (JTAG interface)	LA-7753[2]	JTAG Debug Cable for MPC5500/ MPC5600 (OnCE)	Only one of these two options is required.
	LA-3736	Universal Automotive JTAG Debug Cable for MPC5xxx (AUTO26)[5]	
Aurora trace preprocessor	LA-3911	MPC57xx Aurora Preprocessor	One required.
JTAG and Nexus Aurora adapter	LA-3871	OnCE JTAG or AUTO26 + Nexus Aurora adapter (includes Samtec 34-pin flex extension cable)	One required.
Multi-core license[6]	LA-7960X	Software license for multi-core operation	One required.
Software warranty	LA-8017	Additional year of software maintenance	Not required for first year, but needed for second year.

1. Can also be used for parallel Nexus trace with the LA-7630 preprocessor.
2. This interface is out of production, but is still supported by Lauterbach.
3. The PowerTrace I is limited in bandwidth. One lane of 5 GBPS, two lanes of 2.5 GBPS, or four lanes of 1.25 GBPS data can be supported.
4. This feature is only available on PowerTrace units manufactured in 2004 or later (serial numbers C05xx or higher).
5. The Automotive JTAG Debug cable supports additional signals that are not available on the standard (LA-7753) MPC5xxx debug cable ($\overline{\text{RSTOUT}}$, $\overline{\text{EVTI}}$, $\overline{\text{EVTO}}$, and external Watchdog Disable).
6. Required for devices with multiple cores. Debug of the enhanced Timing Processing Unit (eTPU), the Generic Timer Module (GTM), or Signal Processing Toolbox (SPT) in addition to the Power Architecture® core, does not require a multi-core license.

4 Lauterbach TRACE32 MPC55xx/MPC56xx/MPC57xx Nexus Parallel Trace Tools

Lauterbach offers two ranges of products that support Nexus parallel trace on the MPC55xx, MPC56xx, and some of the MPC57xx devices that support Nexus parallel trace.[5] The first solution is intended for situations in which larger amounts of trace information, and, therefore, larger trace memories included in the interface, are required. Below is a list of the components that must be ordered for a complete parallel Nexus trace solution. There are currently three options for trace memory size (1 GB, 2 GB, or 4 GB) in the trace interface. This solution can be upgraded to support the Nexus serial (Aurora) trace, see section Lauterbach TRACE32 MPC57xx Nexus Aurora Trace Tools for details on the components required for the high-speed serial trace solution.

Table 5. Complete Nexus parallel trace solution for the MPC55xx, MPC56xx, and MPC57xx devices

Description	Order/part number	Name	Requirement
Host interface[1]	LA-7699[2]	PowerDebug II Ethernet	One required
	LA-3505	PowerDebug PRO	
Trace module[1]	LA-7692	PowerTrace II (1 GB)	Only one of these options is required
	LA-7693	PowerTrace II (2 GB)	
	LA-7694	PowerTrace II (4 GB)	
Nexus trace interface (Auto-Focus)[3]	LA-7630	Nexus-MPC5500-AF	One required
Connector adapter	LA-7631	Mictor38 connector on target	Only one of these options is required.[4]
	LA-7636	SAMTEC 50 connector on target	
Multi-core license[5]	LA-7960X	Software license for multi-core operation	One required
Software warranty	LA-8017	Additional year of software maintenance	Not required for first year, but needed for second year

1. Can also be used for Nexus Aurora trace with the LA-3911 preprocessor.
2. This interface is out of production, but is still supported by Lauterbach.
3. Includes support of the JTAG signals. Supports 2- to 16-bit parallel Message Data Outputs.
4. Generally, the Mictor38 connector is used for Nexus parallel ports that implement 12 or fewer Message Data Output (MDO) pins. The SAMTEC 50 is recommended for 12 or more MDO signals. 12-bit implementations may use either connector.

5. See Table 1 for the MPC57xx devices that support Nexus parallel trace.

5. Required for devices with multiple cores. Debug of the enhanced Timing Processing Unit (eTPU), the Generic Timer Module (GTM), or Signal Processing Toolbox (SPT) in addition to the Power Architecture® core, does not require a multi-core license.

The second is a low-end solution with a limited amount of trace memory (256 MB or 512 MB only) implemented in the tool. This option is shown in the table below.

Table 6. Nexus parallel trace solution for the MPC55xx, MPC56xx, and MPC57xx devices (256 MB/512 MB trace capacity)

Description	Order/part number	Name	Requirement
Host interface[1]	LA-7690[2]	PowerTrace 512 MB	One required
	LA-7707[2]	PowerTrace 256 MB	
Nexus trace interface (Auto-Focus)[3]	LA-7630	Nexus-MPC5500-AF	One required
Connector adapter	LA-7631	Mictor38 connector on target	Only one of these options is required[4]
	LA-7636	SAMTEC 50 connector on target	
Multi-core license[5]	LA-7960X	Software license for multi-core operation	One required
Software warranty	LA-8017	Additional year of software maintenance	Not required for first year, but needed for second year

1. This host interface can also be used for JTAG-only connections
2. This interface is out of production, but is still supported by Lauterbach.
3. Includes support for the JTAG signals. Supports 2- to 16-bit parallel Message Data Outputs
4. Generally, the Mictor38 connector is used for Nexus parallel ports that implement 12 or fewer Message Data Output (MDO) pins. The SAMTEC 50 is recommended for 12 or more MDO signals. 12-bit implementations may use either connector.
5. Required for devices with multiple cores. Debug of the enhanced Timing Processing Unit (eTPU), the Generic Timer Module (GTM), or Signal Processing Toolbox (SPT) in addition to the Power Architecture® core, does not require a multi-core license.

5 Lauterbach TRACE32 MPC57xx Trace-to-Memory Tools

In addition to trace to an external interface, devices that include the Nexus Aurora router allow trace to memory. This trace memory can be decoded by JTAG-only tools with the addition of a memory trace license. Below is a list of the components that must be ordered for a complete JTAG debugging solution with trace-to-memory support. This configuration also supports just JTAG debugging.

Table 7. Trace to Memory or JTAG only solutions for the MPC57xx devices

Description	Order/part number	Name	Requirement
Host interface	LA-7699[1]	PowerDebug II Ethernet[2]	One required
	LA-7708[1]	PowerDebug USB2	
	LA-3505	PowerDebug PRO[2]	
	LA-3500	PowerDebug USB3	
Debug cable (JTAG Interface)	LA-7753[1]	JTAG debugger for the MPC5500/ MPC5600 (14-pin)	One required

Table continues on the next page...

Table 7. Trace to Memory or JTAG only solutions for the MPC57xx devices (continued)

Description	Order/part number	Name	Requirement
	LA-3736	Automotive JTAG debug cable for MPC5xxx (Auto26)[3]	
Multi-core license[4]	LA-7960X	Software license for multi-core operation	One required
Trace license	LA-7968X	Trace license for MPC57xx	Optional, but required to support trace to memory
Software warranty	LA-8017	Additional year of software maintenance	Not required for first year, but needed for second year

1. This interface is out of production, but is still supported by Lauterbach.
2. This interface can be extended to add a trace interface (see previous sections).
3. The Universal Automotive JTAG Debug cable supports additional signals that are not available on the standard (LA-7753) MPC5xxx debug cable ($\overline{\text{RSTOUT}}$, $\overline{\text{EVTI}}$, $\overline{\text{EVTO}}$, and external Watchdog Disable)
4. Required for devices with multiple cores. Debug of the enhanced Timing Processing Unit (eTPU), the Generic Timer Module (GTM), or Signal Processing Toolbox (SPT) in addition to the Power Architecture® core, does not require a multi-core license.

6 References

For more information on Nexus and the Nexus implementation on the MPC57xx families of devices, see the device reference manual, the core reference manuals, and the additional documents listed in the table below.

Table 8. Nexus References

Document	Title	Availability
AN4088	MPC5500/MPC5600 Nexus Support	freescale.com
IEEE-ISTO 5001-1999	The Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface, Version 1	nexus5001.org
IEEE-ISTO 5001-2003	The Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface, Version 2.0	
IEEE-ISTO 5001-2012	The Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface, Version 3	
training_nexus.pdf	Lauterbach Nexus Training	lauterbach.com/manual.html
debugger_mpc55xx.pdf	Lauterbach MPC5xxx Debugger	

Appendix A Lauterbach support for IEEE 1149.7 operation

Support for the recently-released IEEE 1149.7 standard is being added to many of the MPC57xx devices. Not all of the Lauterbach TRACE32 hardware interfaces support the IEEE 1149.7 Class T4 (2-wire mode) or higher interface. The table below shows the different TRACE32 interfaces that support the MPC55xx, MPC56xx, and MPC57xx devices.

Table A-1. TRACE32 IEEE 1149.7 hardware interface support

TRACE32 hardware interface	Manufacturer date[1]	IEEE 1149.7 T4 (2-wire mode) support
LA-7610	All	Does not support IEEE 1149.7 operation.
LA-7753	October 2009 and newer	Supports full IEEE 1149.7 operation.
	September 2009 and older	Does not support IEEE 1149.7 operation.
LA-7630	January 2013 and newer	Supports full IEEE 1149.7 operation.
	June 2009 through December 2012	IEEE 1149.7 operation is supported, however, the TDO signal must not be connected to the trace connector when IEEE 1149.7 is used.
	May 2009 and older	Does not support IEEE 1149.7 operation
LA-3736	All	Supports full IEEE 1149.7 standard

1. The manufacturer date is encoded into the serial number for the interface: CYYMMxxxxxx (YY = year MM = month).

Appendix B Nexus Class Definitions

To understand the level of debug support for a device, the Nexus standard divides the debug features into four classes. The four feature classes are Class 1 Basic Run Control, Class 2 Dynamic Debug (Instruction) Trace, Class 3 Data Trace, and Class 4 Advanced Debug features. In the Nexus 5001 standard, some features are optional but most features are required. In addition, the standard defines a minimum set of resources required for some features.

In general, when features from a higher class are included in a lower class device, (but not the full higher class features), Freescale refers to the extra features as a "+" feature. For example, a Class 2+ device includes all Class 2 features plus some features from either Class 3 or Class 4; however, all of the Class 3 or 4 features are not supported. Freescale implements the Class 3 feature of read/write access (access to memory in real-time while the core is executing) on all MPC55xx/MPC56xx/MPC57xx devices that support Nexus. The Class 4 feature of enabling and disabling of trace via a watchpoint is included on all devices that support trace.

An overview of the Nexus classes is summarized in the table below.

Table B-1. Nexus feature/class overview

Class	Type	Required features	Optional features
1	Static debug (basic run control)	Read/write registers and memory	—
		Start and stop program execution, including single instruction execution	
		Control entry and exit debug mode from both reset and user modes	
		Set breakpoint and watchpoints	
		Read Nexus device identification	
2	Dynamic debug	Real-time ownership trace	Port replacement
		Real-time program trace	
		Watchpoint messaging	
3	Data collection	Data trace of memory/peripheral writes	Data trace of memory/peripheral reads
		Dynamic memory read and write	Data acquisition
4	Advanced debug	Complex triggering control of trace	Start/stop memory substitution on watchpoint
		Memory substitution	

B.1 Nexus Class 1—basic runtime control

Nexus Class 1 provides the minimal debug capability and includes primarily Stop mode debug features (static debug), but some dynamic debug features are required. Most features require that the device is stopped, either at a breakpoint or immediately after reset. Required capabilities include the ability to start and stop the target system, set breakpoints, (at least two are required), reset the target system, read and write memory, and execute single instructions. All of these features are also supported in the higher classes (Class 2, Class 3, and Class 4). The table below summarizes the required static debug features for Class 1 devices.

Table B-2. Class 1 (and higher) required static debug features

Development feature	Description	Required (R) or Optional (O) feature	Class 1	Class 2	Class 3	Class 4[1]
Register read/write	Ability to read and write user registers in Debug (stopped) mode	R	√[2]	√	√	√
Memory read/write	Ability to read and write user memory in Debug (stopped) mode	R	√	√	√	√
Debug entry from reset	Enter Debug mode from reset	R	√	√	√	√
Debug entry from running	Enter Debug mode from user mode	R	√	√	√	√
Exit to Run mode	Exit a Debug mode to a user mode	R	√	√	√	√
Single step	Single step instruction in user mode and re-enter Debug mode	R	√	√	√	√
Breakpoint	Stop program execution on instruction/data breakpoint and enter Debug mode (a minimum of two breakpoints are required)	R	√	√	√	√

1. Most Class 4 features are not supported on the MPC55xx or MPC56xx devices.
2. √ indicates that the feature is implemented on all devices.

The table below shows the required dynamic debug features for a Class 1 device.

Table B-3. Class 1 (and higher) dynamic debug features

Development feature	Description	Required (R) or Optional (O) feature	Class 1	Class 2	Class 3	Class 4[1]
Breakpoints/ watchpoints	Ability to set breakpoints and watchpoints	O	√[2]	√	√	√
Device identification message[3]	Access to a device identification either through a JTAG ID register or an automatic message	R	√	√	√	√

1. Most Class 4 features are not supported on the MPC55xx or MPC56xx devices.

2. Since Class 1 does not require a Nexus Auxiliary Output port, watchpoints are signaled by asserting the Event Out ($\overline{\text{EVTO}}$) signal.
3. The Device ID message is optional on JTAG-based devices.

B.2 Nexus Class 2—dynamic debug (instruction trace)

Nexus Class 2 requires that all Class 1 features be supported and adds support for watchpoints messages, ownership trace messages, and program trace messages. An auxiliary output port is required to transmit this data outside of the device. All Class 2 or higher cores and modules support all features of Class 2.

NOTE

The Nexus standard allows trace data to be stored in an internal memory space and read via the JTAG interface. This option is currently not supported in the MPC55xx/MPC56xx devices. Trace to memory is supported on some MPC57xx devices.

Table B-4. Class 2 (and higher) dynamic debug features

Development feature	Description	Required (R) or Optional (O) Feature	Class 1	Class 2	Class 3	Class 4[1]
Watchpoint messages	Capability to signal the occurrence of watchpoint with a signal ($\overline{\text{EVTO}}$) and/or a watchpoint (an address was accessed, either from execution or by a load/store access)	R	—	√	√	√
Ownership trace messages	Capability to monitor the process identifier in real-time with a message	R[2]	—	√	√	√
Program trace messages	Monitor program flow in real-time	R[2]	—	√	√	√
Port replacement	Low-speed input/output port replacement and high-speed I/O port sharing	O	—	Not supported	Not supported[3]	Not supported[3]

1. Most Class 4 features are not supported on the MPC55xx or MPC56xx devices.
2. Required for Class 2 and above.
3. This optional feature is not implemented.

B.3 Nexus Class 3—data trace

Nexus Class 3 requires all Class 2 features and adds the capability to perform the data trace of writes by the MCU core. The ability to trace reads is an optional feature that is not required, but it is supported on current devices and modules that support Nexus Class 3 features.

Table B-5. Class 3 (and higher) dynamic debug features

Development feature	Description	Required (R) or Optional (O) feature	Class 1	Class 2	Class 3	Class 4[1]
Data trace (writes only)	Monitor data writes while process runs in real-time	R	—	—	√	√
Read/Write Access (RWA)	Read and write memory locations without stopping the core CPU.	R[2]	—	Supported[3]	√	√
Data Trace (reads)	Monitor data reads while the CPU runs in real-time.	O	—	—	Supported	Supported[4]
Data Acquisition	Transmit data values for acquisition by the tool	O	—	—	Supported	Supported[5]

1. Most Class 4 features are not supported on the MPC55xx or MPC56xx devices.
2. Required to be supported for Class 3 and above.
3. The Nexus read/write access feature is supported by all Freescale automotive devices that support Class 2 and above.
4. This is an optional feature that is implemented.
5. This is an optional Nexus feature that is supported on devices that use either the e200z4 or the e200z7 core.

B.4 Nexus Class 4—advanced features

Nexus Class 4 requires all Class 3 features and adds some very advanced features such as memory substitution and the ability to enable or disable trace upon a watchpoint occurrence. The ability to start or stop trace (either program or data trace depending on the class device) based on a watchpoint is supported on all MPC55xx and MPC56xx devices that support Class 2 or Class 3.

Table B-6. Class 4 (and higher) dynamic debug features

Development feature	Description	Required (R) or Optional (O) Feature	Class 1	Class 2	Class 3	Class 4[1]
Memory substitution	Allows program execution directly from the Nexus port upon either a reset or an exception.	R	—	—	—	√
Start/stop trace on watchpoint occurrence	Supports the ability to start or stop ownership, program, or data trace upon a watchpoint occurrence.	R[2]	—	Supported	Supported[3]	√[4]
Start/stop memory substitution based on watchpoint occurrence	Ability to start or stop memory substitution upon a watchpoint occurrence.	O	—	—	—	O

1. Most Class 4 features are not supported on the MPC55xx or MPC56xx devices.
2. Required to support Class 4.
3. This feature is not required or optional, but it is implemented.
4. This is a required Class 4 feature, but it is supported on devices that support either Class 2 or Class 3 features.

Appendix C Nexus trace overview

The Nexus auxiliary (AUX) port transmits trace information out of the microcontroller (MCU). This information can be used to reconstruct events or operations that occurred inside the MCU. Nexus supports multiple trace clients within the MCU which all transmit information through a single Nexus AUX port. Each Nexus message is tagged with the source client identifier and the type of message. The actual trace information that is available depends on the Nexus client and can generally be enabled individually(both the type of messages and which clients).

Table C-1. Types of Nexus messages generated by Freescale automotive MCUs

Message type	Description
Program Trace	Transmits any discontinuities in the program flow, such as branches and interrupts. Multiple types of messages can be generated.
Ownership Trace	Provides information on when the process identification is changed.
Data Trace	Allows reads or writes performed by the Nexus client to be traced.
Watchpoint Trace	Generated any time a watchpoint match occurs in the program flow. These can be programmed for many types of events within the MCU.
Device Identification	Allows information about the MCU to be transmitted on start-up to allow tools to identify the target system MCU type.
Debug Status	Provides additional information that may be needed to reconstruct software execution; information such as trace being enabled or disabled, or if the device has entered debug mode.
Data Acquisition	Optional feature that allows software control of information to be transmitted over the auxiliary bus.
Read/Write Access	Provides access to memory-mapped resources over the auxiliary port. These messages are only supported on full bidirectional auxiliary ports and are not used on combined JTAG/ auxiliary ports.
In-Circuit Trace	Allows for information that does not conform to a standard program or data trace protocol. These messages can be used to monitor internal signals. Support of this type of message is optional.
Error	Transmitted when an error condition occurs, such as internal buffer overflows.

The Nexus trace information can be output directly from the Nexus Port Controller (NPC) or Nexus Aurora Router (NAR) in a parallel manner (similar to previous generations) for storage in internal memory; the parallel information can also be serialized for transmission over a serial interface through the Nexus Aurora interface.

Appendix D Revision history

The table below shows the complete revision history of this document.

Table D-1. Revision history

Revision	Date	Author	Changes
0	November 2012	Randy Dees	Initial customer release.
1	July 2013	Randy Dees	Introduction : removed MPC574xK from summary table; added MPC5775K and MPC574xG
2	August 2015	Randy Dees	Introduction : added MPC5746R, MPC5746R Emulation Device, and

Table D-1. Revision history

Revision	Date	Author	Changes
			MPC5777C to summary table; Added PowerTrace I option; changed UAD references to AUTO26. Added new Lauterbach interfaces, marked discontinued interfaces as out of production. Added MPC574xB, MPC574xD. Changed MPC5744P to MPC574xP. Labeled the JTAG bus and the Nexus trace bus in the figures in section Nexus Trace Options Overview

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number AN4591
Revision 2, August 2015