

MPC5644A to MPC5746R Migration Guide

1 Introduction

This document provides a summary of the significant differences in the MPC5644A and the MPC574xR family of microcontrollers. It may be used as an aid for planning a migration to the MPC574xR.

The MPC5644A is fabricated in a 90 nm wafer process, and the MPC574xR is fabricated in a 55 nm process. The MPC574xR incorporates many new features when compared to the 90 nm family devices. As such, there are some changes in the device architecture that require software and hardware modifications. Note particularly the sections labeled ‘Migration Tip’ throughout the document for important details about migration to the MPC574xR. The following table shows the different devices in the MPC574xR family of microcontrollers (MCUs).

Table 1. MPC574xR family of MCUs

Device
MPC5743R
MPC5745R
MPC5746R

The MPC574xR is available in 144 LQFP, 176 LQFP, and 252 MAPBGA packages and the functions available of the MCU may vary according to the package chosen. A 292

Contents

1	Introduction.....	1
2	Overview.....	2
3	MPC5746R block diagram.....	4
4	Reset, boot, and initialization sequences.....	6
5	Memory.....	9
6	Core.....	12
7	Clock generation and distribution.....	13
8	Other common peripherals.....	14
9	New modules and peripherals.....	20
10	Functional safety.....	24
11	Removed peripherals.....	26
A	References.....	26
B	Revision history.....	27



Overview

MAPBGA emulation devices is also available that includes many extra features for development. Where applicable these differences are noted in this application note.

2 Overview

Table 2 provides a summary of the feature differences and feature comparison between the MPC5644A and MPC574xR devices.

Table 2. MPC5644A/MPC5746R feature comparison

Feature	MPC5644A	MPC5746R	MPC5745R	MPC5743R
Process	90 nm	55 nm	55 nm	55 nm
Number of cores	1	2 (+1 lock step)	2 (+1 lock step)	1 (+1 lock step)
Core	e200z446	e200z425 (multiple)	e200z425 (multiple)	e200z425
<ul style="list-style-type: none"> Tightly coupled local memories 	No	32 KB D-MEM, 16 KB I-MEM (each core)	32 KB D-MEM, 16 KB I-MEM (each core)	32 KB D-MEM, 16 KB I-MEM
<ul style="list-style-type: none"> Signal Processing Extension (SPE) 	Yes	No		
<ul style="list-style-type: none"> Lightweight Signal Processor (LSP) 	No	Yes		
<ul style="list-style-type: none"> Book E instructions 	Yes	No		
<ul style="list-style-type: none"> Cache 	8 KB instruction cache	8 KB instruction cache per core		8 KB instruction cache
<ul style="list-style-type: none"> Non-maskable interrupt (NMI) 	Yes	Yes		
<ul style="list-style-type: none"> MMU 	Yes	No		
<ul style="list-style-type: none"> Operating frequency (maximum) 	150 MHz	200 MHz ¹		
Memory Protection Unit (MPU)	Yes	Dual (SMPU0, SMPU1), plus CMPU on core		
XBAR	Single	Dual (fast and slow)		
Core Nexus	Class 3+	Class 3+		
Nexus trace capability	4 or 12-bit parallel Nexus port	16K on-chip trace memory on all devices, 4 lane Nexus Serial Aurora interface with 1M trace memory (Emulation device only)		
SRAM	192 KB ²	320 KB including D-MEM (I-MEM adds another 32 KB) ²	256 KB including D-MEM (I-MEM adds another 32 KB) ²	160 KB including D-MEM (I-MEM adds another 16 KB) ²
Flash memory	4,096 KB	4,256 KB	3,232 KB	2,208 KB
Flash memory fetch accelerator	Yes	Yes		
External bus	16-bit non-multiplexed/32-bit multiplexed	No		

Table continues on the next page...

Table 2. MPC5644A/MPC5746R feature comparison (continued)

Feature	MPC5644A	MPC5746R	MPC5745R	MPC5743R
Calibration support	16-bit non-multiplexed/32-bit multiplexed external bus	Production device—16 KB calibration/trace memory, Emulation device—1 MB SRAM		
DMA	64 channels			
Serial ports	3 (eSCI)	6 (4 + 2 MSB) LINFlex	6 (4 + 2 MSB) LINFlex	5 (3 + 2 MSB) LINFlex ³
FlexCAN	3	4		
DSPI	3	7 (5 + 2 MSB)	7 (5 + 2 MSB) ⁴	6 (4 + 2 MSB) ⁵
Micro-Second Bus (uSB/MSB)	Yes	Yes, enhanced LVDS MSB		
FlexRay	Yes	No		
Ethernet	No	Yes		
Timers	5 x PIT, 4 x STM, 1 x SW watchdog, RTI	PIT0—eight 32-channel; PIT1—two 32-bit channels w/64-bit chain; Two STM, 4 channels each, 3 SW watchdog, RTI		
Enhanced Modular Input/Output System (eMIOS)	24 channels	32 channels (2 x 16 channels)		
Single Edge Nibble Transmission (SENT)	No	6 channels		
Enhanced Timing Processing Unit (eTPU)	32 channels	64 channels (2 x 64 channels)		
• Reaction channels	6 channels	10 channels		
• Code memory	14 KB	24 KB		
• Data memory	3 KB	6 KB		
Sigma Delta Analog to Digital Converter ($\Sigma\Delta$ ADC)	No	3 16-bit		
Analog to Digital Converter	4 12-bit (RSD ⁶)	4 12-bit (SAR ⁷)		
Temperature sensor	Yes	Yes		
Variable gain amp	Yes	No		
Decimation filter	2			
Sensor diagnostics	Yes			
Frequency Modulated Phase Lock Loop (FMPLL)	Yes (1 PLL)	Yes (2 PLL ⁸)		
Low (1.2v) Voltage Regulator Controller (VRC)	Yes			
Supplies	5 V, 3.3 V, 1.2 V internal regulator controller	Single 5 V, 1.25 V internal regulator controller		

Table continues on the next page...

Table 2. MPC5644A/MPC5746R feature comparison (continued)

Feature	MPC5644A	MPC5746R	MPC5745R	MPC5743R
Low power modes	Stop mode, Slow mode	Stop mode, Slow mode		

1. Some device options support only 150 MHz operation. See the device Data Sheet.
2. Includes 32K of standby SRAM
3. LINFlex 3 is not supported.
4. DSPI 4 is not supported on the 176 LQFP device.
5. DSPI 4 is not supported. In addition, DSPI M1 is not supported on the 144 LQFP device.
6. Redundant Signed Digit
7. Successive Approximation Register
8. One PLL (PLL1) supports FM.

The following table shows a comparison of the package availability of the MPC5644A and the MPC574xR MCUs.

NOTE

Although the MPC5745R and MPC5743R are available in LQFP176 packages, the pin-out between them and the MPC5644A are not compatible.

Table 3. MPC5644A/MPC574xR package comparison

Package	MPC5644A	MPC5746R	MPC5745R	MPC5743R
LQFP144	—	—	—	•
LQFP176	•	—	•	•
MAPBGA208	•	—	—	—
MAPBGA252	—	•	•	—
MAPBGA292 Emulation Device	—	•	—	—
BGA324	•	—	—	—

NOTE

The MPC5644A allows some of the Input/Output (I/O) supplies to be a nominal 3.3 V or 5.0 V (complete range is 3.0 V to 5.25 V). The I/O supplies of the MPC574xR, must be a nominal voltage of 5.0 V (3.5 V to 5.5V). 3.5 V allows for power supply droop during high current conditions.

3 MPC5746R block diagram

The figure 1 below is a block diagram of the full MPC5746R device, showing the cores and the partitioning of the system between computational shell (processing cores) and the peripheral CrossBar interface. This partitioning allows the cores to have high-speed access to the flash and SRAM, while providing a separate bus structure for the Direct Memory Access (eDMA) and the two clusters of peripherals (figure 2 shows a list of the peripherals included in clusters A and B). In addition, it shows the safety lakes for Core 0 and the eDMA. The safety lake guarantees that the redundant safety circuits are completely independent from and isolated from the main core 0 and the primary eDMA interface. The redundant circuitry provides a mechanism that instruction execution and eDMA operations are performed by separate circuitry to identify failures of those operations due to a latent defect in the MCU. It provides a mechanism to shut down the MCU in an orderly manner in the remote possibility of an MCU failure. The MPC5644A did not have redundant operations and was not intended for safety-critical applications.

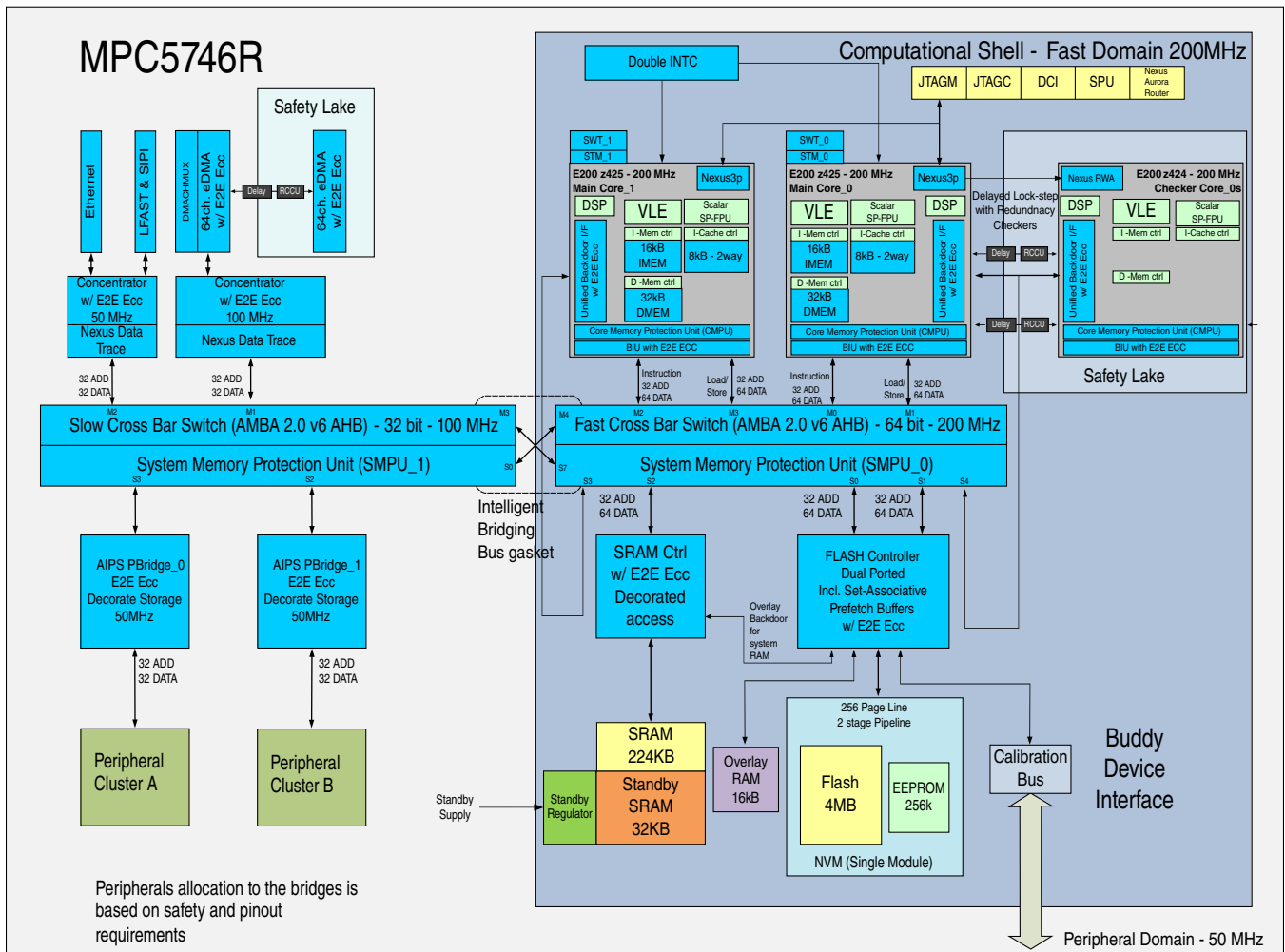


Figure 1. MPC5746R block diagram

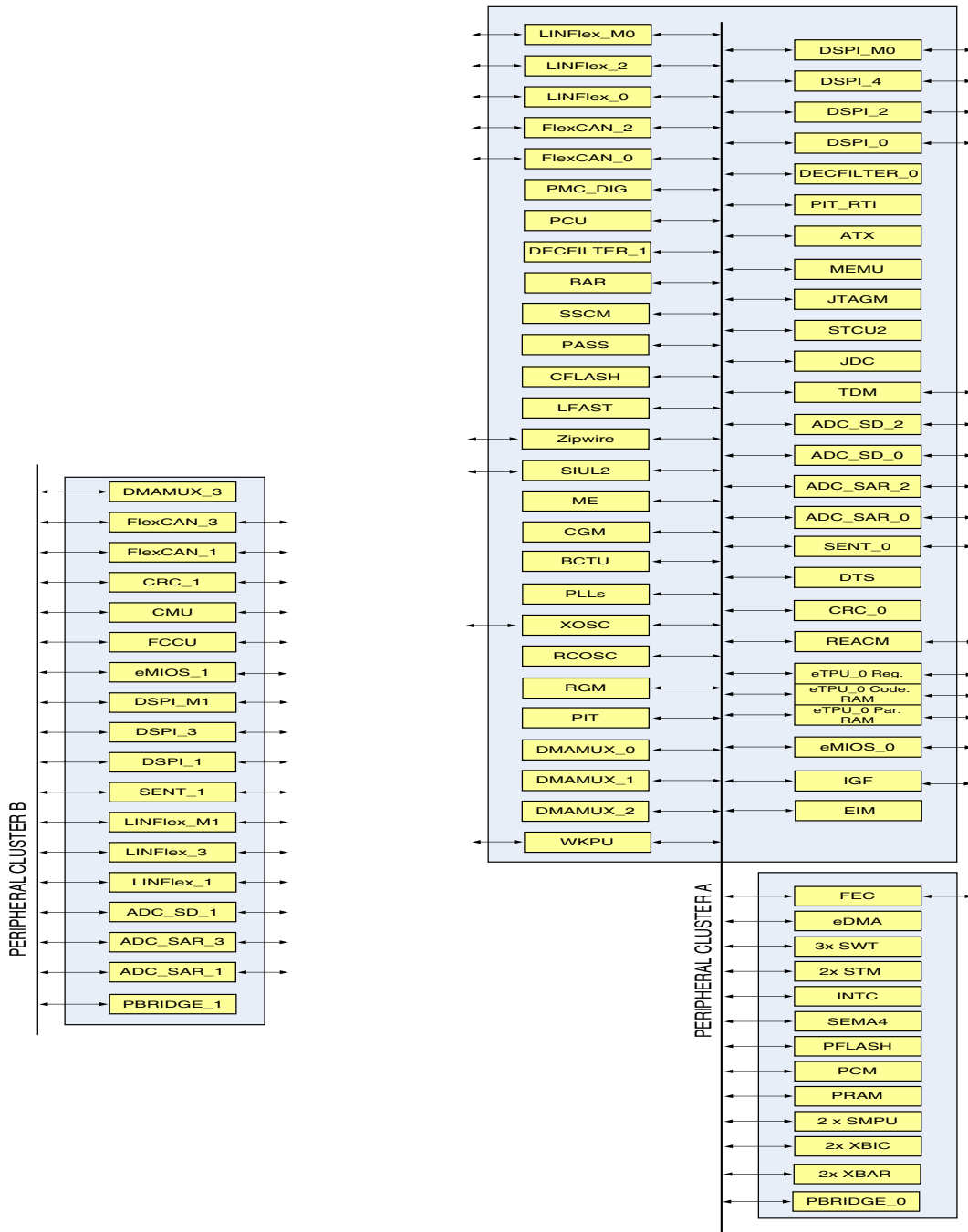


Figure 2. Peripheral allocation for MPC5746R

4 Reset, boot, and initialization sequences

The reset, boot and initialization sequences for the MPC5746R are different from those on the MPC5644A. The MPC5746R platform has been re-architected to provide significant new safety and security features for automotive microcontrollers and to provide additional flexibility in controlling the initial configuration of the device after reset.

- The MPC5644A reset is quite straightforward, with a single reset process and minimal pre-configuration before the application runs.
- The MPC5746R should be considered as a complete system that must be brought out of reset with each of the system components configured for correct operation.
 - Multiple processing cores, configuration and start-up
 - Lock step core preparation and start-up
 - Use of the Device Configuration Format (DCF) records
 - Clock tree and PLL configuration and start-up
 - Configuring and entering the correct run mode

4.1 MPC5644A flash memory boot overview

The MPC5644A incorporates the Boot Assist Module (BAM), a nonvolatile executable ROM program. At reset, the chip begins execution at the reset vector, which is the first instruction of the BAM code. The BAM does a basic setup of the chip's MMU and attempts to locate a valid "reset configuration halfword" (RCHW) at specific predetermined memory locations in flash memory. If a valid RCHW is found, the BAM locates the start address of user initialization code and branches to that address. The user initialization code then completes the device setup and passes control to the main application.

If a valid RCHW is not found, the BAM begins an external serial boot sequence from either eSCI or FlexCAN.

4.2 MPC574xR flash memory boot overview

The MPC574xR boot process uses several different blocks as it progresses. These blocks are the Power Management Controller (PMC), the Reset Generation Module (RGM), the Mode Entry Module (MC_ME), the System Status and Configuration Module (SSCM), the Boot Assist Flash (BAF), and the Self Test Configuration Unit (STCU). The reset sequence and interaction of the various blocks is illustrated at a very high level in Figure 3.

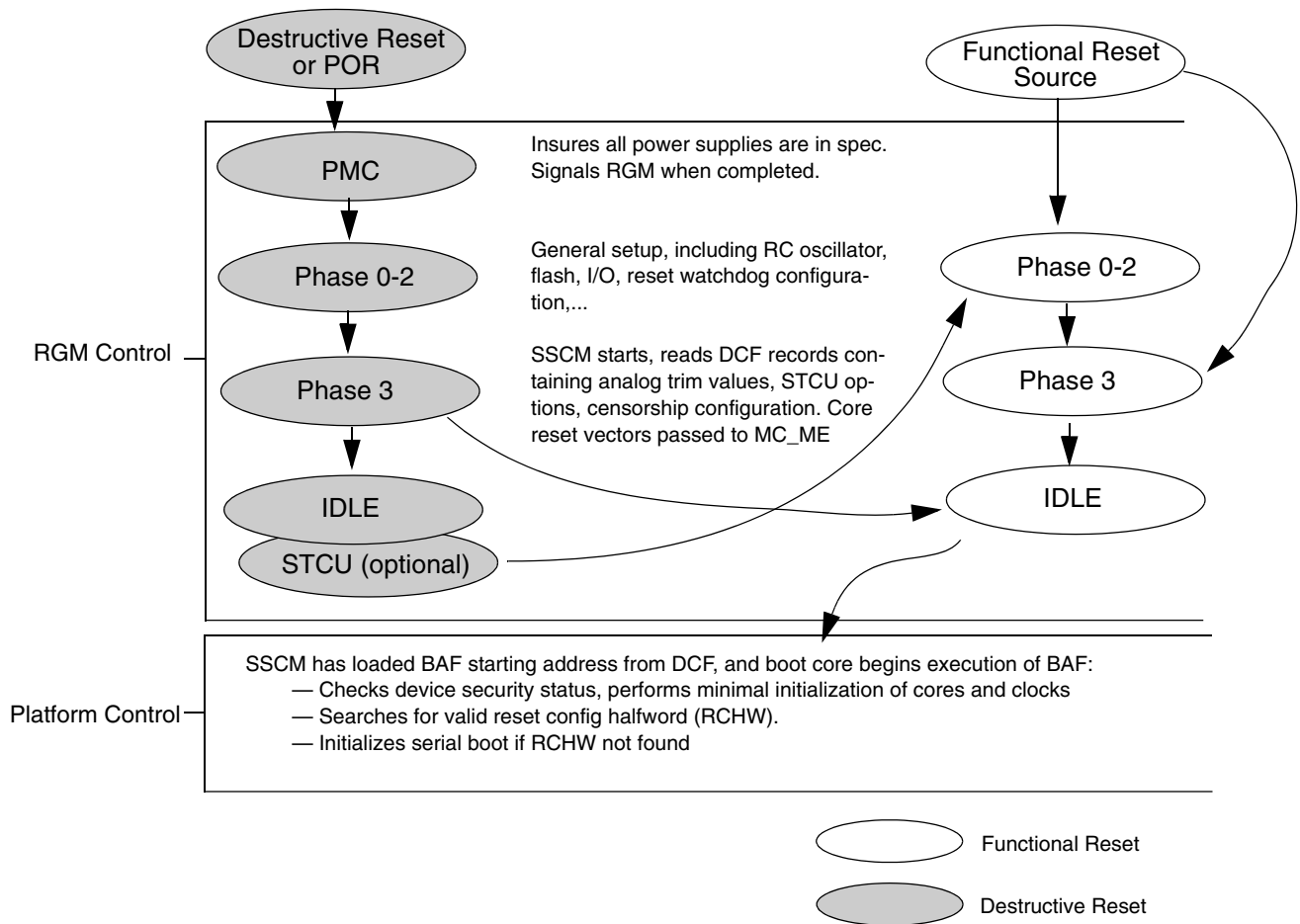


Figure 3. Boot sequence (high level)

Similar to the Boot Assist Module (BAM) in the MPC5644A, the MPC574xR include a Boot Assist Flash (BAF) that assists in configuring the MCU after reset negates. The BAF is preprogrammed at NXP and cannot be changed by the user. The BAF searches for the Reset Configuration Half-Word (RCHW) to begin execution of user code. If a valid RCHW is not found, the BAF initializes for serial boot operations.

Note that the MPC574xR has both destructive and functional resets. This feature allows a full system reset that is typical of a power on reset (POR), or a less destructive functional reset. The functional reset allows reset and initialization of a subset of the system resources, as opposed to a full system reset.

Migration tip

Due to the differences in the boot sequences and architectures of the MPC5644A and the MPC5746R, the application’s initialization software will require modifications or a complete re-coding, especially if the application can benefit from the functional vs. destructive reset feature. Refer to AN4670, *MPC5746R Software Initialization* and the MPC5746R Reference Manual for complete details and example code.

4.3 Device Configuration Format (DCF) records

The MPC5644A relies on external pin states on the BOOTCFG[0] and BOOTCFG[1] to select certain options during the boot process, such as boot from flash memory, boot from external memory, or boot from FlexCAN or the enhanced Serial Communication Interface (eSCI).

The MPC574xR uses a new method for controlling certain options at boot time. A special area of flash memory (UTEST) contains programmed information in Device Configuration Format (DCF) records. During the boot process the BAF and SSCM modules use these pre-programmed values to control certain options for device configuration. Some of these DCFs are preprogrammed at the factory and may not be modified, while others can be programmed by the end user.

The flash memory locations of these records are called the TEST and UTEST flash areas. Only the UTEST area is accessible to the user. Complete details of the DCF records and how they are programmed are available in the MPC5746R Reference Manual.

Migration tip

Though you may prefer to modify some DCF records to suit the specific requirements of the application, the chip is delivered with default DCF records that enable basic reset, boot, and normal device operation without user changes. The application-specific DCF modifications that are desired can be evaluated and tested during development.

5 Memory

This section provides details of the differences in the memories (SRAM, flash and overlay SRAM), including information that is relevant for porting applications to the MPC574xR.

5.1 SRAM

The MPC5746R has 320 KB of SRAM which includes 64 KB of tightly coupled data RAM (D-MEM), plus an additional 16 KB of tightly coupled instruction RAM (I-MEM). The MPC5644A had 192K of SRAM. The following table shows a comparison of the SRAM availability across the different available devices.

Table 4. SRAM memory definition

Start Address	End Address	Allocated Size [KB]	Description	Actual size	MPC5644A 192 KB	MPC5746R 256 KB + 96 KB	MPC5745R 192 KB + 96 KB	MPC5743R 128 KB + 48 KB
System RAM								
0x4000_0000	0x4000_7FFF	32	System Standby RAM	32 KB	•	•	•	•
0x4000_8000	0x4001_FFFF	96	System RAM	96 KB	•	•	•	•
0x4002_0000	0x4002_FFFF	64	System RAM	64 KB	•	•	•	
0x4003_0000	0x4003_FFFF	64	System RAM	64 KB		•		
0x4004_0000	0x4FFF_FFFF	261888	Reserved	—				

Migration tip

The application initialization code must initialize all SRAM up to 256 KB, and the tightly coupled memories. Additionally, any linker directive files must be updated to reflect the change in the memory map. The MPC574xR also provides extensive

Memory

SRAM ECC monitoring and correction features that are not available on the MPC5644A. Refer to the MPC5746R Reference Manual for a complete description of these features and how they are used.

5.2 Flash memory

The MPC574xR and the MPC5644A both have 4 MB of flash memory. However, block sizes and configurations are different. Since the addresses and sizes of the flash blocks in the MPC5746R are different, the compiler linker file will need to be updated. A new strategy may be required for allocating flash blocks for different user use cases (such as areas of the flash used for EEPROM Emulation).

Migration tip

The MPC5746R requires a different flash programming algorithm that is provided by NXP as a software driver. Any proprietary flash programming algorithms or drivers must be re-coded for the MPC5746R. Any PC tools used for programming flash will need to be updated. The flash block sizes have changed as shown in [Table 5](#).

Table 5. Flash comparison

Flash block type	Block size	MPC5644A	MPC574xR
Low address space	16K blocks	2 (32K)	5 (80K)
	32K blocks	0 (0K)	2 (64K)
	48K blocks	2 (96K)	0 (0K)
	64K blocks	2 (128K)	4 (256K)
Mid Address space	16K blocks	0 (0K)	8 (128K)
	64K blocks	0 (0K)	2 (128K)
	128K blocks	2 (256K)	0 (0K)
High address space	128K blocks	20 (2.56M)	0 (0K)
256K block Space	256K blocks	0 (0K)	14 (3.584M)

Migration tip

The flash block addresses and allocations used in the MPC5746R are different from the MPC5644A

5.3 Local memories (MPC574xR only)

The MPC574xR incorporates local Data (D-MEM) and Instruction (I-MEM) memories, that are not available on the MPC5644A. These memories are for data and instructions, respectively, and are memory mapped and tightly coupled to the core, allowing single-cycle accesses. Both computational cores incorporate their own discretely mapped local memory regions. Both cores can access the other core's local memory if required.

Table 6. Local memory definition

Start Address	End Address	Allocated Size [KB]	Description	Actual size	MPC5644A [†]	MPC5746R 96 KB	MPC5745R 96 KB	MPC5743R 48 KB

Table continues on the next page...

Table 6. Local memory definition (continued)

0x5000_0000	0x5000_3FFF	64	I-MEM CPU0	16 KB		•	•	•
0x5000_4000	0x5000_FFFF		Allocated for I-MEM CPU0	[48 KB]				
0x5001_0000	0x507F_FFFF	8128	Reserved I-MEM CPU0	—				
0x5080_0000	0x5080_7FFF	64	D-MEM CPU0	32 KB		•	•	•
0x5080_8000	0x5080_FFFF		Allocated D-MEM CPU0	[32 KB]				
0x5081_0000	0x50FF_FFFF	8128	Reserved D-MEM CPU0	—				
0x5100_0000	0x5100_3FFF	64	I-MEM CPU1	16 KB		•	•	
0x5100_4000	0x5100_FFFF		Allocated for I-MEM CPU1	[48K]				
0x5101_0000	0x517F_FFFF	8128	Reserved I-MEM CPU1	—				
0x5180_0000	0x5180_7FFF	64	D-MEM CPU1	32 KB		•	•	
0x5180_8000	0x5180_FFFF		Allocated D-MEM CPU1	[32 KB]				
0x5181_0000	0x51FF_FFFF	8128	Reserved D-MEM CPU1	—				

1. The MPC5644A does not support local memories.

Migration tip

Any application data or code that requires fast access and/or execution times can be relocated to these memory areas via linker file options. Putting the core stack in the local D-MEM speeds up execution of subroutines calls that stores parameters on the stack. A possible good use of the local I-MEM is exception handlers or frequently called routines. This is an alternative to locking routines in the cache.

5.4 Overlay SRAM

Overlay SRAM is included in the MPC574xR family of controllers as part of a comprehensive set of calibration and debug features. Calibration re-mapping is supported using overlay SRAM. Overlay SRAM can be mapped over specific regions of on-chip flash memory so that any access to an overlaid flash address is routed to the overlay SRAM instead. This enables the calibration of constant data without requiring additional external RAM and calibration memory interfaces. The calibration remap function supports three different types of overlay SRAM. Refer to the MPC5746R Reference Manual for a complete description of these features and how they are used.

Table 7. Different Overlay sections used for remapping

Overlay Section Start Address	Overlay Section End Address	Size	Overlay Section
0x0D000000	0x0D003FFF	16Kbytes	Internal overlay SRAM
0x0C000000	0x0C0FFFFFF	1024Kbytes	Extended overlay SRAM ¹

1. ED devices only.

Also, a portion of the system RAM can be used as overlay SRAM.

6 Core

The MPC5644A incorporates the e200z446n3 core while the MPC574xR incorporates an e200z425n3 core. With the exception of the SPE APU, and Book E instruction support, the two instruction sets are the same. No modifications to compiler switches or options are required if using the Variable Length Encoded instruction set and not using SPE on the MPC5644A.

Hand coded SPE instructions will require modification (see [Signal Processing Lite \(LSP\)](#)), and any assembly code that contains Book E instructions (classic 32-bit Power Architecture) will need to be ported to the VLE instruction set for use on the e200z425n3 core.

6.1 Core Memory Protection Unit (CMPU)

A new core memory protection unit (CMPU) has been added to each of the e200z425n3 cores. The CMPU performs a similar function as the MPU or SMPU (see [Memory Protection Unit \(MPU/SMPU\)](#)); however, it has a total of 24 region descriptor entries, and allows additional control of accesses based on the current mode of the core (user or supervisor). This provides a multi-level approach to memory protection, individual core resource protection versus system level protection. This is a new feature for the e200z4 core subsystem (e200z4 core plus the local memories) and is designed to satisfy additional safety requirements that are required by some applications.

Migration tip

The CMPU, SMPU0, and SMPU1 are designed to be used together as a complete memory protection system. When partitioning the access protection, consider that access control enforced by an SMPU consumes fewer core resources than those controlled by the CMPU. Refer to the MPC5746R Reference Manual for a complete description of how the CMPU and SMPUs can be configured.

6.2 Signal Processing Lite (LSP)

The e200z425 core used in the MPC5746R supports the Signal Processing Light extension, rather than the SPE2 extension used in the MPC5644A.

Migration tip

Since the upper 32-bit registers are not implemented in LSP (GPRs are no longer 64-bit with upper and lower halves), 64-bit vector operations are not supported in the same hardware model for the MPC5746R. Any user code that employs SPE instructions should be examined for compatibility with LSP due to the change in the core register model and instruction mnemonics.

6.3 Cache

The core used in the MPC5644A and MPC5746R has 8 KB of instruction cache. The cache implementation is identical and no changes are required to cache configuration software.

Migration tip

If code is locked in the cache, or if the cache is being used for the software stack, the configuration code and/or linker file may require modification. Alternately, you may consider using the local memories for this purpose instead of the cache. See [Local memories \(MPC574xR only\)](#) and the MPC5746R Reference Manual for more information on local memories.

6.4 Lock step core

The MPC5746R incorporates a delayed lock step core that executes in parallel with the computational core. This lock step core compares all functional and operational behavior with the computational core and can generate an exception in the case of any incorrect operation or result.

Migration tip

When the cores exit reset and begin executing, the state of the delayed lock step core and any calculations or results is constantly compared to the state of the computational core for correctness. In order to avoid state mismatches that would generate an exception, software must ensure that both the computational and lock step cores are in identical states (including values in certain registers) before execution begins. Refer to AN4670, *MPC5746R Software Initialization* for a code example that can be executed in the low-level start-up routines for the e200z4 cores prior to the start of the application.

7 Clock generation and distribution

In typical applications the MPC5644A incorporates a single Frequency Modulation Phase-Lock Loop (FMPLL) for generation of the system clock. This clock is distributed throughout the platform and, where required, peripheral modules (LinFLEX, FlexCAN, and so on) have prescaler functions that divide the system clock to the correct frequency for the particular module.

The MPC574xR incorporates a much more flexible clock generation architecture with the following new features:

- Additional clock source selection options at some module levels—16 MHz IRC, system clock, or crystal/ external oscillator
- Integrated Progressive Clock Switching (PCS) module—steps PLL frequency during start-up to control instantaneous current demands
- Dual PLLs with associated divider networks—one for the core clocks (with FM), and a second (without FM) for peripheral clocks
- Multiple auxiliary clock trees with dedicated dividers—allow for more flexible clock frequency and routing control
- Important features of the MPC5644A clocking system such as clock monitoring are retained and/or enhanced

The following sections provide highlights of some of the key new modules in the MPC5746R clock generation architecture.

7.1 Clock Generation Module (MC_CGM)

The Clock Generation Module (MC_CGM) is the user programmable module that controls configuration of the clocks for the chip. This includes initializing and enabling the dual PLLs, configuring clock routing, and setting frequency dividers for auxiliary clock trees.

The modules that require specific clock frequencies (LinFlex, FlexCAN, DSPI) still retain module level prescalers and dividers.

Migration tip

While many of the MC_CGM blocks are essentially register compatible with similar modules on the MPC5644A, each module's initialization code will require inspection for potential changes due to the new clock distribution architecture.

7.2 Internal 16 MHz RC Oscillator (IRC)

An internal 16 MHz RC oscillator is provided on the MPC574xR. This oscillator is designed to provide a reliable clock source during the boot sequence, and a backup clock source in the event of PLL failure. Additionally, it may be optionally routed as a clock source to various auxiliary clock trees.

Migration tip

Since the MCU boots from power on reset (POR) using the IRC clock, user initialization code must configure and enable the PLLs for full speed device operation after all destructive resets. Some functional reset sources will leave the FMPLL configuration unmodified, and reinitialization of the clocks is not required. See [MPC574xR flash memory boot overview](#) and the MPC5746R Reference Manual for additional information on destructive and functional resets.

7.3 Dual PLL architecture

The MPC574xR incorporates two PLL modules, one with FM capability (PLL0) and one without FM (PLL1). The architecture is designed so that PLL0 is used to generate the core clocks and can also provide the input to PLL1 (when FM is not used), which is used to generate the peripheral and auxiliary clocks. This scheme allows FM modulation on the cores where higher frequency operation is desired (PLL0), thereby reducing EMI emissions without affecting the clock provided to the peripherals (PLL1) where a non-FM clock is preferred.

Migration tip

While the PLL0 and PLL1 blocks are similar in function and programmability when compared to the MPC5644A, the blocks are not completely register/bit compatible. Any PLL initialization code will need modification to support the differences in the register model. Additionally, the enhanced clock tree routing and divider features require new MC_CGM code for correct implementation.

8 Other common peripherals

The following sections highlight any differences in peripherals common to both devices that may require software modifications. Additionally feature changes that can allow improved system performance are identified. Note that due to the changes in the architecture of the MPC574xR, many new peripherals are added. See [MPC574xR flash memory boot overview](#) for details of the new modules and peripherals.

8.1 Power Management Controller (PMC)

While the expected voltage monitoring capabilities are retained, the MPC574xR PMC is different from that implemented on the MPC5644A. Additionally, the architecture of the MPC574xR is implemented in a way that provides various run modes that allow low-power operation under certain conditions. This level of control is not available on the MPC5644A.

Migration tip

When compared to the MPC5644A, the overall power supply hardware design for the system will likely require modifications. However, the implementation of the PMC on the MPC574xR provides the opportunity for cost reduction within the ECM supply design. A single 5V supply may be used along with a power transistor and internal regulator to generate the 1.25V VDD supply.

Although functionally similar, the PMC register map is different and software will require modification. See the MPC5746R Reference Manual for complete information on the new PMC implementation.

8.2 Software Watchdog Timer (SWT)

The SWT implemented on the MPC574xR has a similar register model and operation when compared to the MPC5644A. However new features are added including additional SWT service modes.

Migration tip

Existing software may require modification as the bit mappings have changed slightly. Since the MPC574xR is targeted for a different operating frequency, the watchdog timeout value may need to be recalculated accordingly.

8.3 System Timer Module (STM)

The MPC574xR incorporates the same STM as implemented by MPC5644A.

Migration tip

Any existing driver code does not require modification. However, the two devices may be operated at different frequencies so any timeout values may need to be recalculated using the frequency selected for the MPC574xR peripherals.

8.4 Interrupt Controller (INTC)

The interrupt controller (INTC) used in the MPC574xR is similar in design and operation to the interrupt controller implemented in the MPC5644A. Hardware and software vector modes are still supported, and the register model is similar. However, because the MPC574xR has two cores instead of a single core, there are dual interrupt controllers, one for each computational core.

The interrupt vector offset register (IVOR4) is writable on the MPC5644A. The application writes an offset to IVOR4, which, in conjunction with the interrupt vector prefix register (IVPR), forms the address of the interrupt handler for the external (peripheral) interrupts. In the e200z425 cores implemented by the MPC574xR, there are no IVORn registers, only the IVPR register and fixed offsets for each of the interrupt types, including IVOR4, the vector for external interrupts.

Migration tip

Because each core may be interrupted independently by the same peripheral, the INTC strategy should be redesigned to efficiently allocate resources and processor time within the application, providing maximum system performance. Additionally, because of the change from IVOR registers to fixed offsets, any interrupt setup and configuration requires modifications in order to operate correctly on the MPC574xR.

8.5 enhanced Direct Memory Access (eDMA)

From an operational standpoint, eDMA is essentially unchanged from the MPC5644A; however, some new features have been added to the eDMA module. The number of DMA channels remains at 64, but the MPC574xR incorporates a new DMAMUX that allows up to 64 DMA request 'slots' to be routed to 16 DMA channels.

The MPC574xR also includes a second lock step DMA engine to meet safety requirements. The second lock step DMA is transparent to user operation and requires no change in how the DMA is configured or used in the application.

Migration tip

DMA channel assignments are changed and the register model, while similar, is not completely identical, particularly the Transfer Control Descriptor (TCD). Software must be inspected and modified as necessary to ensure that correct DMA channels are being accessed and triggered.

Refer to the MPC5746R Reference Manual for full details on using the eDMA and DMAMUX modules.

8.6 Temperature Sensor

The MPC5644A and MPC574xR both incorporate Temperature Sensors. These sensors are similar in design and operation; however, the temperature sensor on the MPC574xR is designed to provide additional accuracy.

Migration tip

Due to improvements in the Temperature Sensor module, any existing code should be inspected for compatibility.

8.7 Periodic Interrupt Timer (PIT)

The basic PIT timer block used on the MPC5644A is reused on the MPC574xR; however, the following enhancements have been added:

- 1 real time interrupt (RTI) channel and 8 PIT channels, instead of 1 RTI channel and 4 PIT channels. All channels remain 32-bit.
- A method for chaining PIT channels 0 and 1 together to create a single 64-bit timer.

Migration tip

Any code that configures the PIT and RTI timers may require modification due to different operating frequencies. Additionally, if a 64-bit timer is required, code must be written to configure and start the timer. Due to the additional channels, the module register map has been modified.

8.8 System Integration Unit Light (SIUL/SIUL2)

The SIUL incorporated in the MPC5644A is different than the SIUL2 incorporated in the MPC574xR. The SIUL is used to configure the pins of the MPC5644A device by controlling which peripheral has access to the physical pin or whether the pin is configured as a GPIO. The SIUL also controls the configuration of the actual pad: input enable, output enable, slew rate and hysteresis.

The SIUL2 performs similar functions; however, the SIUL2 has been enhanced to also provide the capability to route internal signals to peripherals as well as external pins.

Migration tip

All SIUL code should be rewritten to support SIUL2. The programming model remains similar, particularly the pad configuration registers (PCRs), but they are not identical, and the PCR assignments (which peripherals are assigned to which PCRs) is completely different. Additionally there are now two types of multiplexed signal configuration registers (MSCRs), those that support actual physical pins and those that support routing of internal signals. Depending on the application, it may be necessary to redesign the SIUL scheme.

8.9 Enhanced Time Processing Unit (eTPU2/eTPU2+)

The MPC574xR includes the eTPU2+ module which has several enhancements over the eTPU2 that is incorporated in the MPC5644A. The most significant new feature is the inclusion of a performance monitor. This provides a mechanism for obtaining performance and channel servicing information. Other features have also been added, including alternate angle counter inputs.

The MPC574xR allows for much greater flexibility in signal routing of the device pins to the eTPU module, including routing the pins through an Input Glitch Filter. See [Input Glitch Filter \(IGF\)](#).

Migration tip

By default after reset, the IGF is in Configuration Mode which disables signals from the pins to the eTPU. The IGF must be initialized to allow signals to the eTPU. See [Input Glitch Filter \(IGF\)](#).

Migration tip

The eTPU2+ has been specifically designed to be 100% code compatible with legacy eTPU2 code. Any existing eTPU2 code may be used without modifications. Using the new eTPU2+ features will require additional code. If any of the new features are added, some legacy code may also require modifications.

8.10 Analog-to-Digital Controllers (ADCs)

The MPC5644A implements a single enhanced Redundant Signed Digit Queued Analog to Digital converter eQADC module consisting of two ADCs with multiple command and result queues. The MPC574xR implements two $\Sigma\Delta$ ADCs¹, and four Successive Approximation Register (SAR) ADCs instead of the eQADC. These new ADC modules have a different method of operation and a new programming model. In addition, the Variable Gain Amplifier on the analog inputs is no longer supported.

Generally, all functions associated with the DMA and interrupts (commonly conversion command and result queues) are retained on the two new ADC types.

Migration tip

The SAR ADC incorporates direct hardware connectivity with the Reaction Channel Module (see [Reaction module](#)) and the cross trigger unit (BCTU) that is new to the MPC574xR device. Additional software must be written to enable and use the ADC in conjunction with the Reaction Channel Module and BCTU; however, the new architecture provides precise reaction channel control with very low CPU overhead. Note that DMA and interrupt numbers and assignments are different, requiring software modifications.

8.11 FlexCAN

The MPC574xR implements the FlexCAN3 rather than the FlexCAN2 module that is implemented on the MPC5644A. Additionally, there are four FlexCAN instantiations on the MPC5746R instead of three. The FlexCAN3 module has some new features that are not available on FlexCAN2, including more message buffers on some of the modules.

On the MPC5644A, both of the FlexCAN2 modules implement 64 message buffers. FlexCAN_0 and FlexCAN_1 on the MPC574xR implements 96 message buffers. (FlexCAN_3 only implements 64 message buffers.)

Table 8. FlexCAN module message buffer size

Device	FlexCAN_0	FlexCAN_1	FlexCAN_2	FlexCAN_3
MPC5644A	64 Message Buffers	64 Message Buffers	64 Message Buffers	—
MPC574xR	96 Message Buffers	96 Message Buffers	96 Message Buffers	64 Message Buffers

The message buffers in the MPC574xR FlexCAN modules implement ECC that requires initialization. See [SRAM initialization](#).

Migration tip

1. Also called Sigma Delta ADCs.

Other common peripherals

The basic driver code for the FlexCAN2 module should be usable on FlexCAN3 with few or no modifications; however, using the new features may require additional code. Refer to the MPC5746R Reference Manual for complete information on the FlexCAN3 module.

8.12 Memory Protection Unit (MPU/SMPU)

The MPC5644A incorporates a memory protection unit (MPU) that provides a software-configurable means of controlling accesses to peripherals and/or specific memory locations. The MPU provides 16 region descriptors that each control the address range and access controls for a specific memory region.

The MPC574xR includes two instances of the next generation MPU, SMPU0 and SMPU1. The SMPU modules provide 12 descriptors in each instance and perform the same functions as the descriptors in the MPU on the MPC5644A. SMPU0 is located on the fast XBAR associated with the computational shell, and SMPU1 is located on the slow XBAR associated with the peripheral shell. See [CrossBar \(XBAR\)](#) for more information on the XBAR switches.

Migration tip

Because the MPC574xR has two system shells, a peripheral shell and the computational shell, there is an SMPU unit associated with each one, located on the slave side of each XBAR. Additionally, there is a CMPU unit associated with the core (see [Core Memory Protection Unit \(CMPU\)](#)). When partitioning and designing the memory protection schemes, a typical use case is for the CMPU to provide access control for the local memories in the cores, and to rely on the SMPU units to control accesses to the peripherals on the slow XBAR and memory on the fast XBAR switch (flash memory and SRAM).

8.13 enhanced Modular Input/Output System (eMIOS)

The eMIOS incorporated in the MPC574xR is essentially the same eMIOS200 module used on the MPC5644A; however, there are two instantiations instead of one, for a total of 32 channels. No software driver modifications are required, but channel mapping may need to be adjusted. The MPC5644A supports 24 channels (channels 0 to 23) and the MPC574xR supports 32 channels (channels 8-23 on each eMIOS instantiation).

The MPC574xR allows for much greater flexibility in signal routing of the device pins to the eMIOS module, including routing the pins through an Input Glitch Filter. See [Input Glitch Filter \(IGF\)](#).

Migration tip

By default after reset, the IGF is in Configuration Mode which disables signals from the pins to the eMIOS. The IGF must be initialized to allow signals to the eMIOS. See [Input Glitch Filter \(IGF\)](#).

8.14 Reaction module

The MPC574xR incorporates a new version of the reaction module (Reaction Module 2). This new version offers feature enhancements over the version in the MPC5644A, including more channels and additional modulation modes. Additionally, the trigger and delivery of the ADC samples from the SAR converters to the reaction module is now managed by the ADC cross-triggering unit (BCTU); see [Cross Trigger Unit \(BCTU\)](#) for further details.

Migration tip

While the operation of the reaction module in the MPC5644A is functionally similar, the architecture of the reaction module in the MPC574xR and how it is interfaced at the system level is different. The change to SAR ADCs and the addition of the BCTU dictate that the reaction channel strategy and software be updated to operate in the context of the new modules.

8.15 Decimation filter

The decimation filter itself is essentially the same as that implemented in the MPC5644A. The MPC5644A provides two decimation filters and a parallel side interface (PSI) where ADC conversion results may be sent directly to decimation filter A. The MPC567xR also provides two decimation filters; however, the PSI is not included.

Migration tip

Because there is no PSI in the MPC574xR, if ADC results are to be sent directly to the decimation filter, user code must set up either DMA or CPU/interrupts for data transfers. Both decimation filters can provide DMA triggers or generate interrupts.

8.16 De-serial Serial Peripheral Interface (DSPI)

The DSPI module incorporated on the MPC5644A includes four DSPI channels. Two (DSPI B and DSPI C) of the four channels also support microsecond channel (MSC) operation. The MPC574xR includes seven DSPI channels. Five of the DSPI modules are for standard DSPI communications, and two are for MSC operation.

Migration tip

Standard DSPI communications are essentially unchanged; however, the MSC communications function has been enhanced with new options. See the MPC5746R Reference Manual for details on DSPI module operation.

8.17 CrossBar (XBAR)

The CrossBar (XBAR) switch implemented on the MPC5644A is 5×4 (5 master and 4 slave ports). The MPC574xR incorporates dual XBAR switches, a 200 MHz computational shell XBAR switch (5×6) and a 100 MHz peripheral shell XBAR switch (3×3).

Migration tip

Basic operation of the XBAR switches is similar, with control for priorities on a master-by-master basis, arbitration methods, slave parking, and so on. However, there is a different number of master and slave ports, and master and slave port numbering has changed. Existing code will need to be revised to correctly implement the desired XBAR setup. See the MPC5746R Reference Manual for details on XBAR operation.

8.18 Debug, trace, calibration, and development options

Both the MPC5644A and MPC574xR support IEEE 1149.1 (JTAG) operation for Boundary Scan board test and debug run control. The MPC574xR also supports the IEEE 1149.7 for run control and basic debug functionality. The IEEE 1149.7 allows for a "2-pin" JTAG mode². In addition, both device families support the IEEE-ISTO 5001 Nexus debug standard. However, the MPC5644A supports the IEEE-ISTO 5001-2003 Nexus Class 3+ standard and includes a parallel Nexus Auxiliary port on the device. The MPC5746R supports IEEE-ISTO 5001-2012 Nexus Class 3+ standard and does not support a physical trace port on the production device (PD). The PD does support trace and calibration features utilizing the on-chip 16K trace/calibration memory. Additionally, an Emulation Device (ED) is available that implements a 1 MByte trace/calibration memory and a physical Nexus High Speed Serial (based on Aurora) trace port. The ED is available from NXP in a 292 MAPBGA package or the ED is available in a small Trace Adapter board that contains the ED and fits the footprint of the production packages on the bottom of the adapter. For more information on the Trace Adapter, see AN5181, *Introduction to the MPC5746R Trace Adapter*.

2. The 2-pin JTAG mode actually uses 3 pins: JCOMP, TMSC, and TCKC.

Table 9. MPC5644A/MPC574xR Development Support

Device	IEEE 1149.1 support	IEEE 1149.7 support	IEEE-ISTO 5001 Nexus parallel trace support	IEEE-ISTO 5001 Nexus serial (Aurora) trace	IEEE-ISTO 5001 Nexus trace to memory
MPC5644A	Yes	No	Yes / 4 or 12-bit	No	No
MPC574xR Production Device	Yes	Yes	No	No	Yes / 16 KByte
MPC574xR Emulation Device	Yes	Yes	No	Yes / 2- or 4-lanes	Yes / 1 MByte

The MPC5644A supported VertiCal, which is similar to the Trace Adapter, except is more modular, requiring additional boards to support an external calibration SRAM and a Nexus trace connector.

8.19 Low-power modes

Both the MPC5644A and the MPC574xR implement low-power mode schemes; however, the implementation of these schemes is device-specific.

Migration tip

Low-power mode operation should be re-evaluated. While some mechanisms for managing low-power modes are similar between the two devices, the actual implementation is not the same. See the MPC5746R Reference Manual for details about various low-power mode options.

8.20 JTAG security

The MPC5644A supported a 64-bit JTAG password to optionally prevent unauthorized access to the MCU. It requires a Censorship enable word and the password to be programmed into fixed locations in the shadow block of the flash. The MPC574xR supports a 256-bit password to prevent unauthorized access. The censorship is enabled via a DCF client, the password is also defined as a DCF record. See [Device Configuration Format \(DCF\) records](#).

9 New modules and peripherals

This section highlights new modules and peripherals included on the MPC574xR that are not found on the MPC5644A. As an introduction, a brief description of each new module is provided in the following sections. Refer to the MPC5746R Reference Manual for complete details on the configuration and use of these new features.

9.1 Cross Trigger Unit (BCTU)

The Cross Trigger Unit (BCTU) is a new hardware module that provides efficient control of SAR ADC triggering, as well as data collection and routing to the reaction modules. Minimal CPU intervention is required.

9.2 Tamper Detect Module (TDM)

The Tamper Detect Module provides a type of flash memory write protection mechanism that forces software to write a record associated with one or more blocks in a Tamper Detection Region (TDR) before the block(s) can be erased.

9.3 Semaphores2 (SEMA42)

The SEMA42 module is new for the MPC574xR microcontroller. Since the MPC574xR is a multi-core platform, the SEMA42 module allows the coherent management of resources that are shared by multiple cores. See the MPC5746R Reference Manual for details on SEMA42 operation.

9.4 Zipwire (Interprocessor Bus)

The Zipwire Interprocessor Bus is a serial communications protocol that allows high-speed data communication between a microprocessor and the memory-mapped peripherals or memory space of a second processor.

The Zipwire interface consists of the LVDS Fast Asynchronous Serial Transmission (LFAST) and Serial Interprocessor Interface (SIPI) modules.

See the MPC5746R Reference Manual for details on Interprocessor Bus Zipwire operation. In addition, see AN5134 *Introduction to the Zipwire Interface*.

9.4.1 Serial Inter-Processor Interface (SIPI)

The Serial Inter-Processor Interface (SIPI) is an application layer protocol which runs on top of the LVDS Fast Asynchronous Serial Transmission (LFAST) module. It is used by the local device to access the shared memory of a remote device. SIPI defines point to point full duplex communication between two devices.

9.4.2 LVDS Fast Asynchronous Serial Transmission Interface (LFAST)

LFAST works as a physical medium of communication between both of the devices. LFAST Consists of 2 LVDS pairs (Transmit and Receive) and clock. It is the physical interface that transports the SIPI frames from one device to another.

9.5 Fast Ethernet Controller (FEC)

The Fast Ethernet Controller (FEC) is a new feature for the MPC574xR. It is a communication controller that supports 10 and 100 Mbps Ethernet/IEEE 802.3 networks. An external transceiver interface and transceiver function are required to complete the interface to the media. See the MP5746R Reference Manual for complete details of the configuration and operation of the FEC module.

9.6 Password and Device Security module (PASS)

For safety and security, the MPC574xR implements the Password and Device Security module (PASS). The PASS receives password challenges and determines their validity. It also maintains chip security and access states. The PASS features include life cycle status, password comparison, JTAG password comparison, and production disable control.

9.6.1 Life Cycle

For safety and security, the MPC574xR implements a device Life Cycle. The Life Cycle is used to enable and disable features of the device, such as debug, serial booting, and flash access depending of the "Life Cycle" of the device, in combination with other security features. In other words, some features are only available during development or during certain stages of manufacturing. For instance, under normal conditions, programming and erasing of the flash with an external debug tool is only available while the device is in the "Customer Delivery" or "OEM Production" stages. However, if censorship is enabled, once the device is moved to the "In Field" state, a debug tool cannot be connected to the device to program or erase the flash, except under specific conditions³. Illegal states of the Life Cycle will hold the device in reset. The Life Cycle of a device can only be advanced in the order shown in the following figure.



Figure 4. Life cycle sequence

Table 10. Device life cycles

Life Cycle name	SSCM Life Cycle Value
Freescall production	0b110
Customer delivery	0b011
OEM production	0b010
In field	0b111
Failure analysis	0b000

9.7 SENT Receiver (SRX)

The Single Edge Nibble Transmission (SENT) Receiver (SRX) is a multi-channel receiver for receiving serial data frames transmitted by a sensor implementing the SENT encoding scheme and presenting them to the CPU for further processing.

9.8 LinFLEX

The MPC574xR replaces the enhanced Serial Communications Interface (eSCI) module implemented on the MPC5644A with a LinFLEX module. The LinFLEX module provides the same communication functionality as the eSCI with an enhanced feature set.

3. The device implements a censorship mechanism that allows a tool to access the device if censorship is both enabled and the tool provides the proper password to allow access.

Migration tip

Existing eSCI software will not execute on the LinFLEX as they are different modules with different register sets. For standard UART communications, the developer can investigate the UART mode that is supported in the LinFLEX module. The same methods for managing serial communications (DMA and interrupt paced) are provided, and will be very similar to UART or LIN driver code from the eSCI. See the MPC5746R Reference Manual for complete details.

9.9 Input Glitch Filter (IGF)

A new input glitch filter (IGF) module is included to filter the input signals for timer channels (eTPU and eMIOS). This module contains 32 channels. The inputs to the IGF are connected to the pins through an input multiplexer, and the filtered outputs from the input glitch filter module are routed to inputs on the timer channels. The glitch filter can be used with all timer inputs. See the MPC5746R Reference Manual for complete details.

NOTE

When enabling the IGF, in addition to the IGF configuration registers, there are two sets of registers in the SIUL Input Multiplexed Signal Configuration Registers for the eMIOS and eTPU signals. One that selects the pad to IGF and one to select the IGF to eTPU/eMIOS input. SIUL MSCR# 652 to 683 registers are used for IGF.

After system reset, the IGF modules are in Configuration mode by default. This mode disables IGF output and prevents the signals from the pins from reaching the eMIOS and eTPU modules.

To route an affected signal directly to an eMIOS or eTPU instance, program the applicable channel of the applicable IGF instance:

1. Ensure that module is enabled in the Module Configuration Register (IGF_MCRn[MDIS] = 0) and that the IGF channel is disabled: IGF_MCRn[FGEN] is 0.
2. Configure the rising and falling edges for bypass:
 - Rising edge: Write 00b to IGF_MCRn[RFM].
 - Falling edge: Write 00b to IGF_MCRn[FFM].
3. Enable the IGF channel: Write 1 to IGF_MCRn[FGEN].

At the end of this sequence, the programmed filter enters Normal mode and passes the affected signal through to the eTPU or eMIOS module. Alternately, IGF can be configured to filter out glitches. See the MPC5746R Reference Manual for additional details.

9.10 Decorated Storage Memory Controller (DSMC)

The DSMC supports atomic read-modify-write bus transactions for 5 types of store operations and 3 types of load operations.

- Bit field inserts
- Compare-and-store
- Bitwise AND, OR, and XOR operators
- Simple memory load
- Swap
- Load-and-set-1 (bit)

9.11 Crossbar Integrity Checker

Functional safety

The Crossbar Integrity Checker (XBIC) verifies the integrity of the crossbar transfers. It verifies the attribute information and verifies feedback of information for each data phase during crossbar transfers. It also provides for error injection for test of software handlers.

9.12 Reset Generation Module

Unlike the MPC5644A reset controller, MPC5746R goes through a set of operations and checks from power up to the release of reset. The total reset sequence is monitored and controlled by the Reset Generation Module (RGM).

The RGM differentiates between multiple reset sources and manages the reset sequence of the chip. The reset sequence is a state machine (PHASE0, PHASE1, PHASE2, PHASE3 and IDLE) which controls the different phases of the reset sequence and the reset signals generated in the system. See the MPC5746R Reference Manual for additional information on difference reset sources and RGM Registers.

9.13 System Status and Configuration Module

During the MPC5644A boot sequence, the Boot Assist Module (BAM) is responsible for finding the reset configuration halfword (RCHW). If a valid RCHW is not found, the BAM searches for the serial boot sequence.

During the MPC5746R boot sequence, the System Status Configuration Module (SSCM) reads the DCF records for a valid boot code. If the flash memory data from the DCF records is not valid, the boot sequence will automatically switch to the Boot Assist Flash (BAF) and search for serial boot modes. See the MPC5746R Reference Manual for additional information on how to configure the SSCM.

10 Functional safety

The MPC574xR is part of a new generation of devices with enhanced safety features, intended for ISO26262 ASIL-D compliance. These features will require new code, and should be included in the software architecture planning. The functional safety features implemented in the MPC574xR are described in detail in the Functional Safety chapter of the MPC5746R Reference Manual. A brief overview of new safety modules is provided below.

10.1 Cyclic Redundancy Check (CRC)

The CRC is a module dedicated to CRC computations. This allows offloading of the work from the CPU. It provides programmable polynomial selection for each context. The CRC module for MPC5746R supports a single context data stream for calculating the checksum.

The module supports four hard wired polynomials (two CRC-8, CRC-32 Ethernet, and CRC-16-CCITT). See the MPC5746R Reference Manual for additional information.

10.2 Memory Error Management Unit (MEMU)

The MEMU is responsible for collecting and reporting error events associated with Error Correction Code(ECC) logic used on SRAM, peripheral system RAM, and flash memory. When any of the error events occur, the MEMU receives an error signal that causes an event to be recorded and corresponding error flags to be set and reported to the Fault Control and Collection Unit(FCCU).

10.3 Fault Collection and Control Unit (FCCU)

The FCCU offers a hardware channel to collect errors and to place the device into a safety state when a failure in the device is detected. No CPU intervention is required for collection and control operation.

The FCCU provides up to two bidirectional signals (EOUT[1:0]) as a failure indication to the external world. After power on, the EOUT signals have high impedance. They show operational state only on software request. In case of a failure event and on software request for EOUT pins indication, the pin(s) are set to faulty state. The overall fault detection, processing and indication time is less than 10 ms. See the MPC5746R Reference Manual for additional information on FCCU and how to configure FCCU.

10.4 Self-Test Control Unit (STCU2)

The STCU is a comprehensive programmable hardware module which supports safety execution directives (self-test, LBIST/MBIST).

It is possible for hardware to manage the device's Logic Built-In Self Test (LBIST) and the SRAM/ROM Built-In Self Test (MBIST) blocks. However, this is optional. The STCU2 module has been designed to provide very flexible configurations, allowing the user to program the parallel/serial execution of the MBIST or the LBIST depending on the power/timing/coverage constraints. Please see the MPC5746R Reference Manual and Safety Manual for additional information on how to configure the STCU2.

10.5 Register Protection (REG_PROT)

A register protection mechanism is built into some modules on the device. This feature allows software control of register write permissions. This provides protect defined memory mapped address locations in a module under protection from being written. The address locations that can be protected are module specific. For the list of modules with protected registers, please see the MPC5746R Reference Manual.

10.6 SRAM initialization

The MPC5644A require SRAM initialization after power on due to ECC. The MPC574xR implements ECC on all of the peripheral memories as well which requires initialization.

Table 11. Peripheral SRAM requiring initialization

Module	Base address	Size	SRAM start address	SRAM end address	
eTPU Parameter RAM	0xFFCC_8000	16K	0xFFCC_8000	0xFFCC_BFFF	Supports byte initialization
eTPU Code RAM	0xFFCD_0000	16K	0xFFCD_0000	0xFFCD_3FFF	Supports byte initialization
FlexCAN1	0xFBEB_4000	16K	0xFBEB_4000	0xFBEB_7FFF	
FlexCAN3	0xFBEB_C000	16K	0xFBEB_C000	0xFBEB_FFFF	
FlexCAN0	0xFFEB_0000	16K	0xFFEB_0000	0xFFEB_3FFF	
FlexCAN2	0xFFEB_8000	16K	0xFFEB_8000	0xFFEB_BFFF	

Table continues on the next page...

Table 11. Peripheral SRAM requiring initialization (continued)

Module	Base address	Size	SRAM start address	SRAM end address	
Internal Overlay RAM	0x0D00_0000	16K	0x0D00_0000	0x0D00_3FFF	
Emulation Device Overlay memory	0x0C00_0000	1M	0x0C00_0000	0x0C0F_FFFF	

10.7 End-to-end ECC

The MPC574xR implements End-to-end Error Correction Codes (e2eECC) through the crossbar (XBAR) switches (system bus). The e2eECC checks that information from a peripheral is not corrupted while in-transit to the CPU core. For the most part, this is transparent to the system, however, it requires updated error handlers be implemented to take advantage of the new feature.

11 Removed peripherals

This section highlights any module that was available on the MPC5644A and is no longer available on the MPC574xR.

11.1 FlexRay

FlexRay operation is removed for the MPC574xR.

11.2 Enhanced Serial Communications Interface (eSCI)

The eSCI module is replaced by the LinFlex module on the MPC574xR. See [LinFLEX](#).

11.3 Variable Gain Amp

Variable gain amp operation is removed for the MPC574xR.

Appendix A References

Additional information can be found in the documentation listed below.

Table A-1. References

Document	Title	Availability
AN4591	Lauterbach MPC57xx Nexus Trace Tools	nxp.com

Table continues on the next page...

Table A-1. References (continued)

Document	Title	Availability
AN4670	MPC5746R Software Initialization	nxp.com
AN5134	Introduction to Zipwire	nxp.com
AN5181	Introduction to the MPC5746R Trace Adapter	nxp.com
AN5234	MPC5743R Single Core Initialization	nxp.com

Appendix B Revision history

Revision number	Date	Changes
1	01/18/2013	Initial release (Confidential)
1.1	01/25/2013	<ul style="list-style-type: none"> • Introduction : Added description of 90 nm process. • Overview : Specified complete core number; added row for Book E. • Editorial changes and improvements throughout.
2.0	August 2017	<ul style="list-style-type: none"> • MPC5746R block diagram : Added peripheral allocation for MPC5746R. • MPC574xR flash memory boot overview : Added BAF to the list in the boot process. • Overlay SRAM : Added overlay SRAM section to the Memory. • Reset Generation Module : Added RGM module to the new modules and peripherals. • System Status and Configuration Module : Added SSCM module to the new modules and peripherals. • Zipwire (Interprocessor Bus) : Updated Zipwire module. • Self-Test Control Unit (STCU2) : Added STCU2 module to the functional safety section. • Updated REG_PROT, FCCU and CRC modules. • Editorial changes and improvements throughout.

How to Reach Us:**Home Page:**nxp.com**Web Support:**nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP 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. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2013–2017 NXP B.V.

Document Number AN4658
Revision 2.0, August 2017

