Migration from IBM 750CX/CXE to MPC7447A

by *Douglas Hamilton*
European Applications Engineering
Networking and Computing Systems Group
Freescale Semiconductor, Inc.

1 Scope and Defintions

The purpose of this application note is to inform the reader of the key differences between IBM 750CX/CXE and MPC7447A PowerPC processors in order to ease migration from IBM 750CX/CXE-based systems to MPC7447A. It will look at the architectural differences, investigating which features have changed, and why they have changed, before discussing the impact of these changes on a migration in terms of hardware and software.

Throughout this document you will see references to:

- IBM 750CX/CXE, which are applicable to MPC750/740, MPC755/745, and IBM 750CX/CXE devices. Any IBM 750CX/CXE-specific features will be explicitly stated.
- MPC7447A, which are applicable to the MPC7450 family of products (MPC7450, MPC7451 and MPC7441, MPC7455 and MPC7445, and MPC7457 and MPC7447/MPC7447A, except where otherwise stated). Since the purpose of this document is to aid the migration from the IBM 750CX/CXE, which does not support L3 cache, the L3 cache features of the MPC745x devices are not mentioned.

Contents

freescale™ semiconductor

2 Feature Overview

There are many differences between the IBM 750CX/CXE and MPC7447A devices beyond the clear difference of the core complexes. This chapter covers the differences between the cores, as well as other areas of interest, including the cache configuration and system interfaces.

2.1 Cores

The key processing elements of the G3 core complex used in the IBM 750CX/CXE are shown below in Figure 1. The G4 complex used in the MPC7447A is shown in Figure 2.

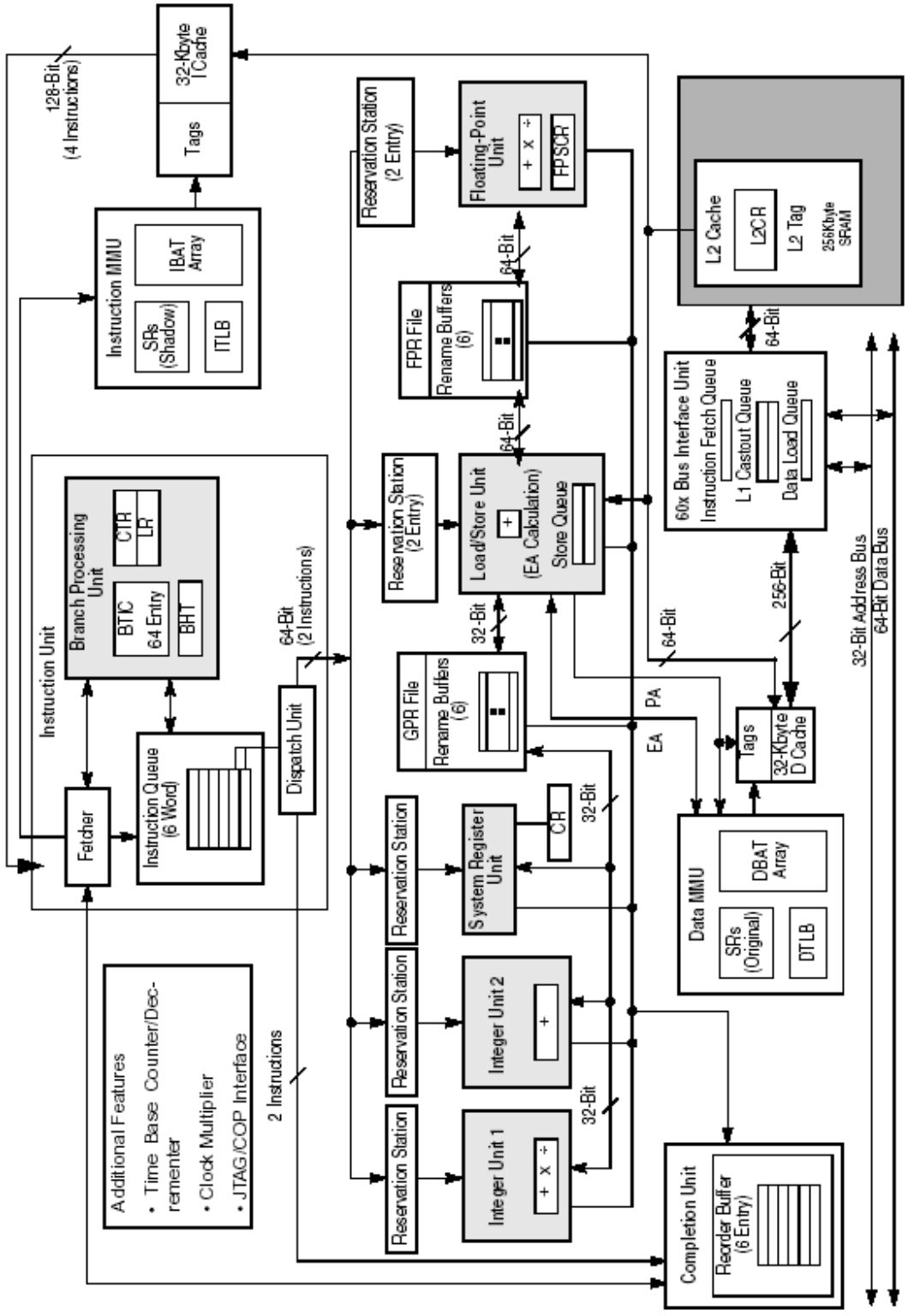

Figure 1. IBM 750CX/CXE Core

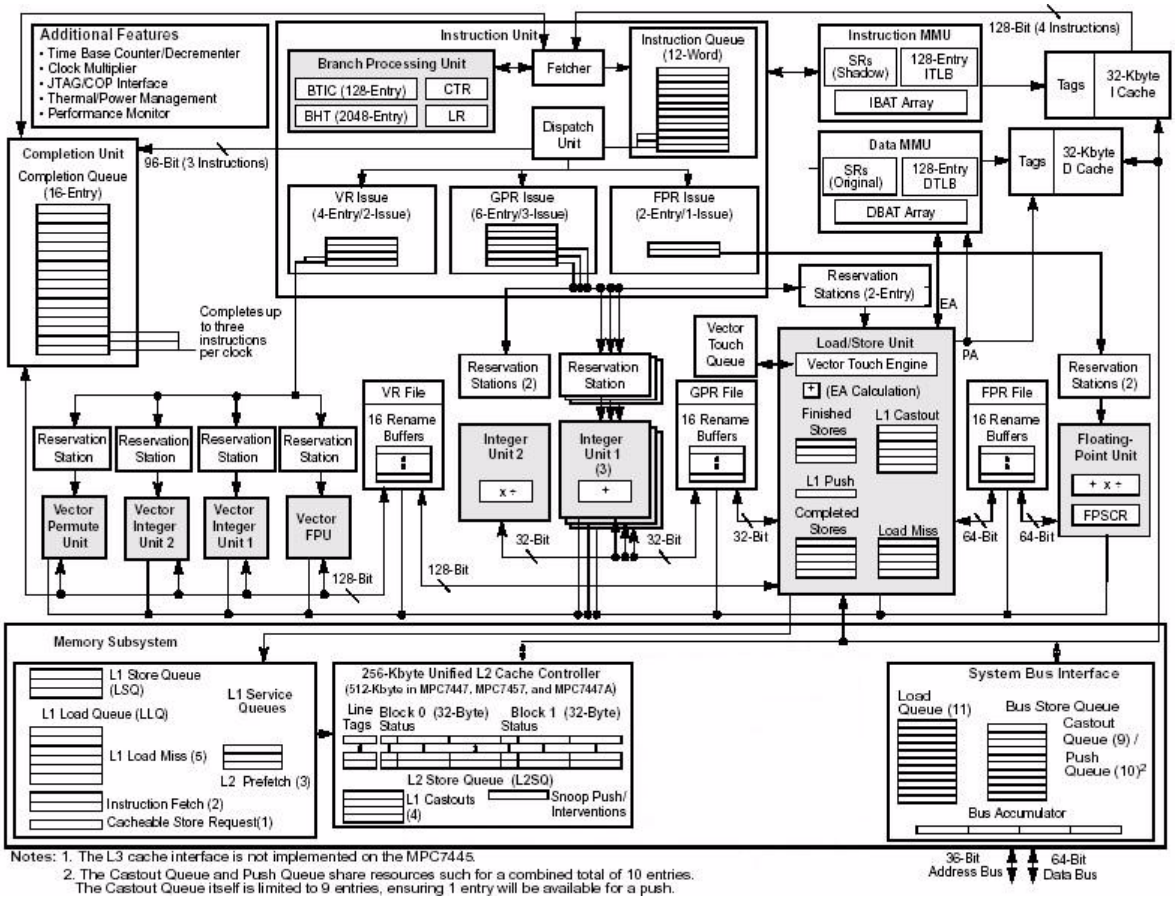

Figure 2. MPC7447A Core

2.1.1 Integer Units

IU1 and IU2 denote Fixed Unit 1 and 2, which are the complex and simple integer units, respectively. The multiply and divide instructions of IU1 are multi-cycle, while all other operations are completed in a single cycle. Both the integers operate on thirty-two 32-bit registers. Table 1 shows the operations that each Fixed Unit can perform. Each unit consists of three units: an adder/comparator, shift/rotate unit, and logical unit. In addition to these standard units, FXU1 also has a multiply/divide unit.

Table 1. FXU Operations

	IU1	IU2
Add, shift, logical functions	Yes	Yes
Multiply/divide	Yes	No

Similar to the IBM 750CX/CXE, the MPC7447A has one complex integer unit with the same functionality as IU1. Unlike the IBM 750CX/CXE, the MPC7447A has three simple integer units similar to IU2, instead of one. A good compiler can take advantage of this when presented with a combination of instructions with multi-cycle latencies, which would tie up two of the integer units, allowing the remaining units to start executing, preventing stalling. Also, the MPC7447A has sixteen GPR rename buffers to support the sixteen entry completion queue, as opposed to the six entry completion queue for the IBM 750CX/CXE.

The floating-point can also source rename buffers as a source operand without waiting for the value to be committed and retrieved from GPR.

2.1.2 Floating Point Units

The IBM 750CX/CXE floating-point unit has 32 64-bit registers for single-precision and double-precision IEEE-754 standards. Different operations have various latencies associated with them due to the three stage pipeline with multiply, add, and normalize stages. The latency/throughput varies from 3/1 clock cycles for single multiply-add, increasing to 4/1 clocks for double multiply and double multiply-add, since two cycles are required in the multiply unit.

The MPC7447A floating-point unit meets the same standards for IEEE-754 precision and, in addition, has an increased pipeline depth of five stages to allow even double-precision calculations to have a one cycle throughput. Although the latency is increased, the overall throughput is better for the majority of double-precision calculations. The floating-point can also source rename buffers as a source operand without waiting for the value to be committed and retrieved from FPR.

2.1.3 Instruction Queues

The instruction queue in the IBM 750CX/CXE can hold up to six instructions. While the instruction queue depth allows, the instruction fetcher retrieves up to the four instructions maximum per clock. Two instructions can be dispatched simultaneously to fixed or floating-point units, the branch processing unit, and load/store unit to execute in a four stage pipeline containing fetch, dispatch, execute, and complete stages.

The MPC7447A offers a twelve slot instruction queue with a maximum of four fetches per cycle and can dispatch up to three instructions per cycle to any of the eleven instruction units, branch processing unit, four integer units, floating-point unit, four 128-bit (Altivec) vector units, or the load/store unit.

2.1.4 Branch Processing Unit

The branch processing unit found in the IBM 750CX/CXE can process one branch while resolving two speculative branches per cycle. It uses a 512 deep BHT for dynamic branch prediction to produce four possible outcomes (not taken, strongly not taken, taken, strongly taken) and incorporates a 64 entry BTIC to reduce branch delay slots by supplying the next instruction(s) from this cache for a particular branch target address, rather than the instruction cache saving a one clock cycle penalty.

In contrast, the MPC7447A processes one branch per cycle, like the IBM 750CX/CXE, but can resolve three speculative branches per cycle. The increased BHT, with 2048 entries, offers the same four predication states, but with the advantage of the larger size. In addition to this, the BHT can be cleared to weakly not taken using HID0[BHTCLR]. The BTIC is twice the size of the IBM 750CX/CXE to give 128 entries arranged as 32 sets using a 4-way set associative arrangement.

2.1.5 Completion Unit

The completion unit works in the IBM 750CX/CXE with the dispatch unit so that it can track dispatched instructions and retire them in order to the completion queue. In following with the dispatch unit, two

instructions can be retired per clock cycle, providing there are slots available in the completion queue. When the instruction is removed from the queue, the rename buffers must have been freed and results any written processor registers, such as GPRs, FPRs, LR, and CTR.

For the MPC7447A, due to deeper pipelines, we can have up to sixteen instructions at some stage of pipeline processing and retire a maximum of three instructions per clock to one of the sixteen completion queue slots.

2.2 Pipeline Comparison

The difference in pipeline depths between the IBM 750CX/CXE and MPC7447A is quite significant. With the IBM 750CX/CXE, minimum depth has been kept to a rather short 4 stages of instruction (fetch, dispatch/decode, execute, and complete). Write-back is included in the complete stage. The pipeline diagram for the IBM 750CX/CXE is shown in Figure 3.

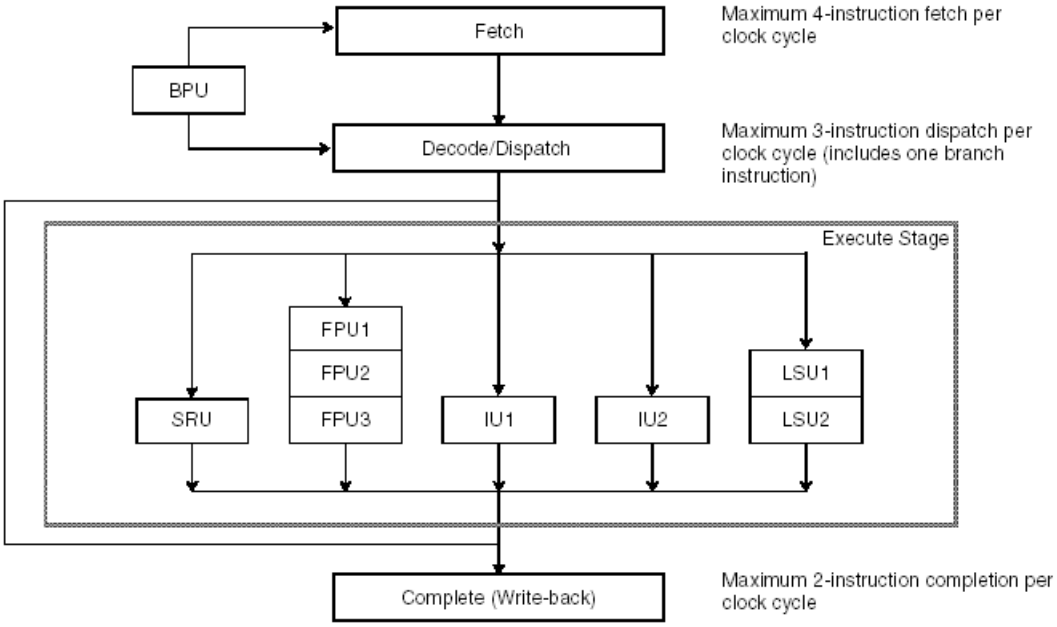

Figure 3. IBM 750CX/CXE Pipeline Diagram

It shows a maximum depth of 6 stages using the floating-point unit. If branch prediction did not work well for a particular application, then having such a short pipeline would be advantageous due a fairly small pipeline flushing penalty. However, branch predication and modern compilers can more often than not prevent frequent pipeline flushes, and so the completion rate of 2 instruction retirements per clock becomes more of a performance bottleneck. It is also worth noting that the IBM 750CX/CXE will not be able to sustain clock rates much greater than 1.1GHz without increasing the depth of the pipeline.

With a minimum depth of 7 stages, the MPC7447A pipeline, shown in Figure 4, boasts efficient use of its additional hardware resources by dispatching 3 instructions per cycle to its execution units and the ability to retire 3 instructions per cycle. Due to the higher maximum frequency of the MPC7447A (up to 1.5GHz), the extra pipeline depth is required to make efficient use of faster running pipeline stage's hardware, reducing the latency of many instructions, such as many floating-point and complex integer instructions.

Compilers can take advantage of the extended pipeline to ensure the target maximum of 16 instructions in flight at any one time is as close as possible.

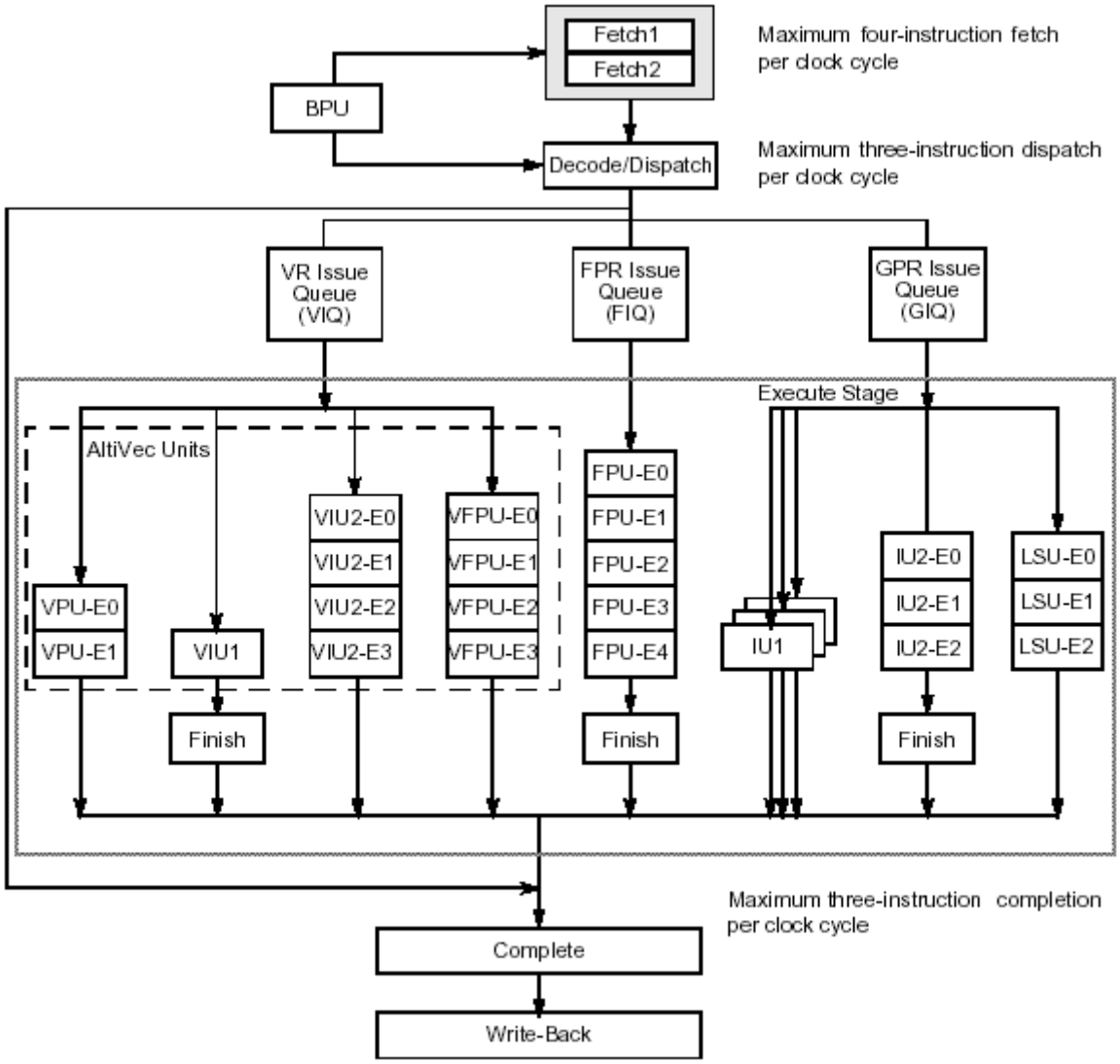

Figure 4. MPC7447A Pipeline Diagram

2.3 L1 and L2 Cache

Table 2 summarizes the differences in L1 and L2 cache configuration.

Table 2. L1, L2 Cache Configurations

	IBM 750CX/CXE	MPC7447A
L1		
Size, configuration	32KB Instruction, 32KB Data 8-way set associative	32KB Instruction, 32KB Data 8-way set associative
Memory Coherency	MEI (Data only)	MESI (Data only)
Locking	Completely	By way
Replacement policy	Pseudo-least-recently used (PLRU)	Pseudo-least-recently used (PLRU)
Per page/block write configuration	Write-back or write-through (Data)	Write-back or write-through (Data)
L2		
Size, configuration	256KB, 2-way set associative Two 32 byte blocks/line	512KB, 8-way set associative (7447A) 1MB, 8-way set associative (7448) Two 32 byte blocks/line
Memory Coherency	MEI	MESI
Locking	No	Completely
Replacement policy	Cast out replacement only	3 bit counter or pseudo random
Parity	8 bits/64 bytes on tags	8 bits/64 bytes on tags and data

2.4 MMU

Table 3 shows the standard PowerPC MMU translation method. The presence of TLBs and page table search logic is optional, although both implementations incorporate them.

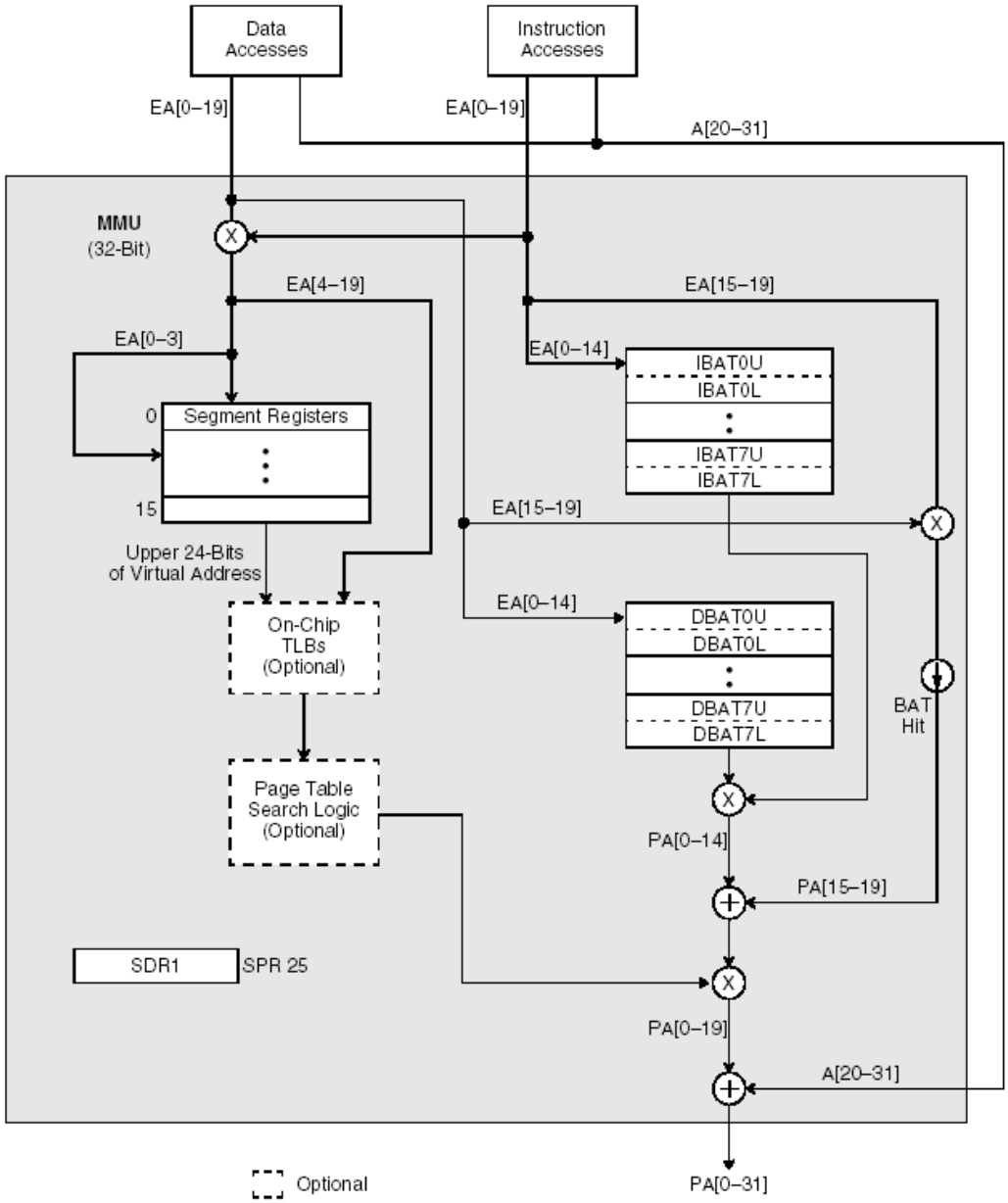

Figure 5. Effective to Physical Mapping

Both the IBM 750CX/CXE and MPC7447A offer the same common features as seen below:

- 128 entry, 2-way associative instruction TLB and data TLB
- Translation for 4KB page size and 256MB segment size
- Block sizes from 128KB to 256MB (4GB for MPC7447A)

The IBM 750CX/CXE only has 4 data and 4 instruction BAT pairs, whereas the MPC7447A has 8 pairs each of data and instruction BATs, allowing a larger range of addresses to be mapped in a simple system using BATs only.

The other significant difference is that the MPC7447A can support 36 bit physical addressing by enabling HID0[XAEN], thus allowing the increased 64GB memory space. The extended block size of greater than 256MB is enabled by asserting HID0[XBSEN], HID0[HIGH_BAT_EN], and using the extra XBL field in the upper BAT registers to select larger blocks up to 4GB. The increased area of memory that can be mapped per BAT means that the programmer does not have to use multiple BATs to map multiple sequential 256MB blocks on the MPC7447A.

An added feature on the MPC7447A is software support for page table searching to offer a custom page table entry and searching operation, if required.

2.5 System Interface

Both the IBM 750CX/CXE and MPC7447A support the 60x bus protocol. The MPC7447A also supports the MPX bus protocol, which is a more efficient protocol based on the 60x implementation. Table 3 highlights the differences in the IBM 750CX/CXE and MPC7447A 60x support.

Table 3. 60x Bus Features

IBM 750CX/CXE 60x Features	MPC7447A 60x Features
32 bit addressing with 4 bits odd parity	36 bit addressing with 5 bits odd parity
64 bit data bus with 8 bits odd parity, 32 bit data bus support	64 bit data bus with 8 bits odd parity
Three state MEI cache coherency protocol	Four state MESI cache coherency protocol
L1 and L2 snooping support for cache coherency	L1 and L2 snooping support for cache coherency
Address-only broadcast instruction support	Address-only broadcast instruction support
Address pipelining	Address pipelining
Support for up to 2 outstanding transactions	Support for up to 16 outstanding transactions
133Mhz maximum bus speed (100Mhz on 750CX, 133Mhz on 750CXE)	167Mhz maximum bus speed

In addition, the MPC7447A supports MPX bus mode, offering up to 16 out-of-order transactions, data streaming, and data intervention for MP systems. These features make the system bus operation much more efficient, thus increasing the effective bandwidth available in the system. The advantages of the MPX bus can be found in Section 3.3, “MPX Mode.”

2.6 Thermal Assist Unit

The Thermal Assist Unit (TAU) used in the IBM 750CX/CXE provides a means of monitoring the junction temperature, offering an advantage over case or cabinet temperature readings since the die temperature would be very different.

It can operate on a one- or two-threshold system, whereby the threshold values are programmed into one or two of the TAU's four special purpose registers. When the temperature reaches one of these thresholds, an interrupt is generated, allowing software to take appropriate action to reduce the temperature accordingly.

Instead of the TAU, the MPC7447A incorporates a temperature diode that connects to an external temperature monitor device. These devices are widely available from vendors such as Analog Devices, Maxim, and National Semiconductor. Using the negative temperature coefficient of the diode at a constant current, the monitor device can determine the junction temperature. Figure 6 shows how the monitoring device can be connected directly to the anode and cathode of temperature diode on the MPC7447A. The monitor chip is also connected via the 60x or MPX bus to a bridge chip/system controller, which then communicates with the monitor chip itself using I^2C. This second connection allows thresholding values to be defined so that the monitor chip can generate interrupts via the Bridgechip in a similar manner to the TAU in the IBM 750CX/CXE.

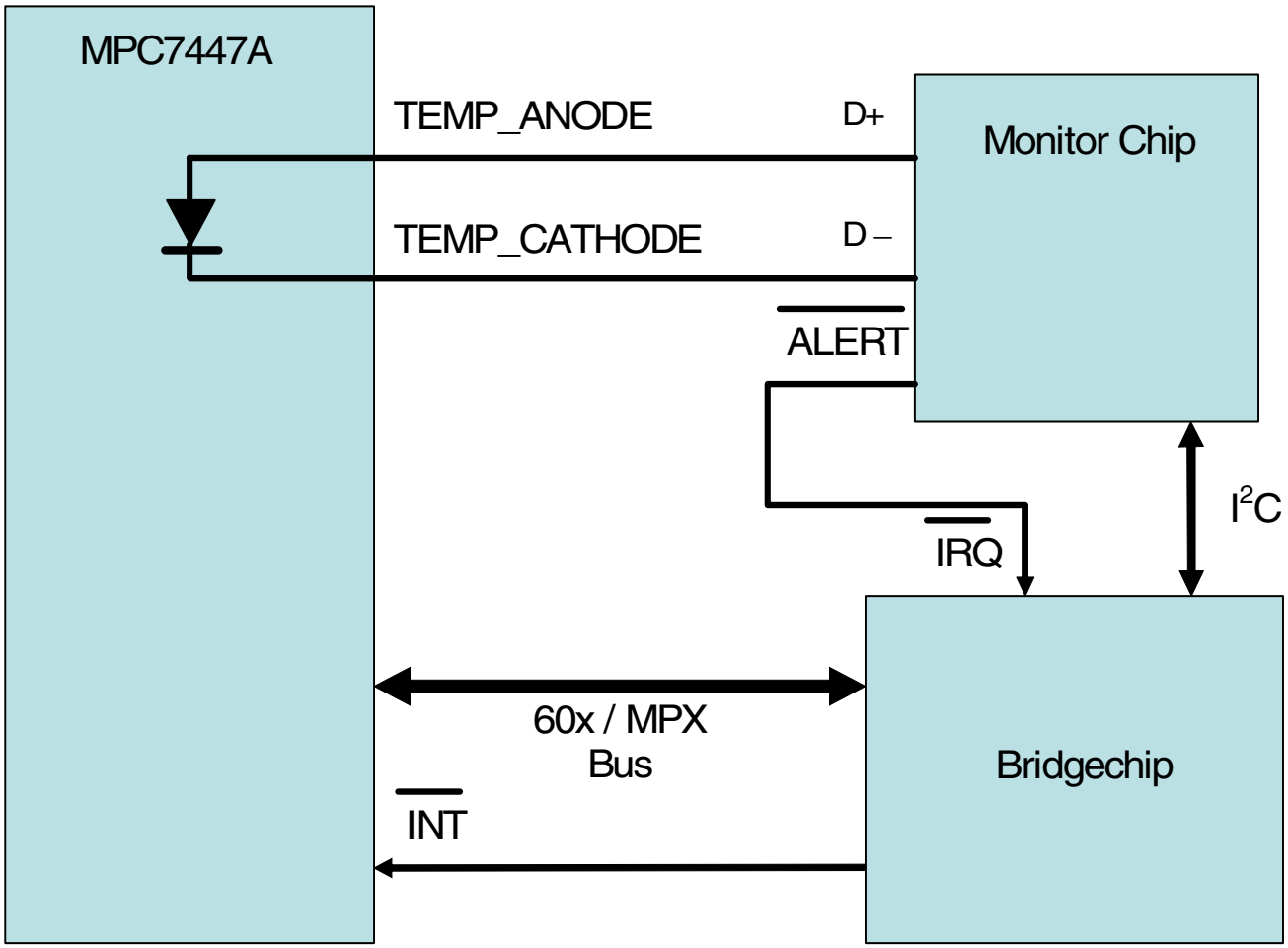

Figure 6. Temperature Monitoring Device Connection

3 MPC7447A Specific Features

This section briefly introduces some major features of MPC7447A devices that are not available on the IBM 750CX/CXE and explains how they can offer significant performance improvements.

3.1 Altivec

Perhaps the most notable difference between the IBM 750CX/CXE and MPC7447A is that of Altivec. It is a short vector parallel extension of the PowerPC architecture in terms of both instructions and hardware. It is available on all MPC7447A(+) implementations and it can offer up to 11 times improvement in performance significance versus the scalar implementations of some applications.

The key features of Altivec are:

- 162 new powerful arithmetic and conditional instructions for intra- and inter-element (parallelism support, for example)
- 4 operands per instruction, 3 sources and 1 destination
- Pipelined execution units to give:
  - 1 cycle latency for simple and permute operations
  - 3-4 cycle latency for compound/complex operations
- No penalty for issuing Altivec/Integer instruction mix

The new instructions allow vector/SIMD operations on 128-bit wide Vector Registers (VR) through any of the four Altivec execution units—permute, simple, complex, and float—each of which have 2, 1, 4, and 4 stage pipes, respectively. These 128-bit VRs can be used as single 128 bit quantity, but also to provide varying levels of parallelism to give a maximum of 16 operations per instruction on 8 bit quantities or, to put into a more comparable format, four 32-bit integer-based operations per instruction. These different levels of parallelism—16x8 bit, 8x16 bit, or 4x32 bit—can be seen in Figure 7.

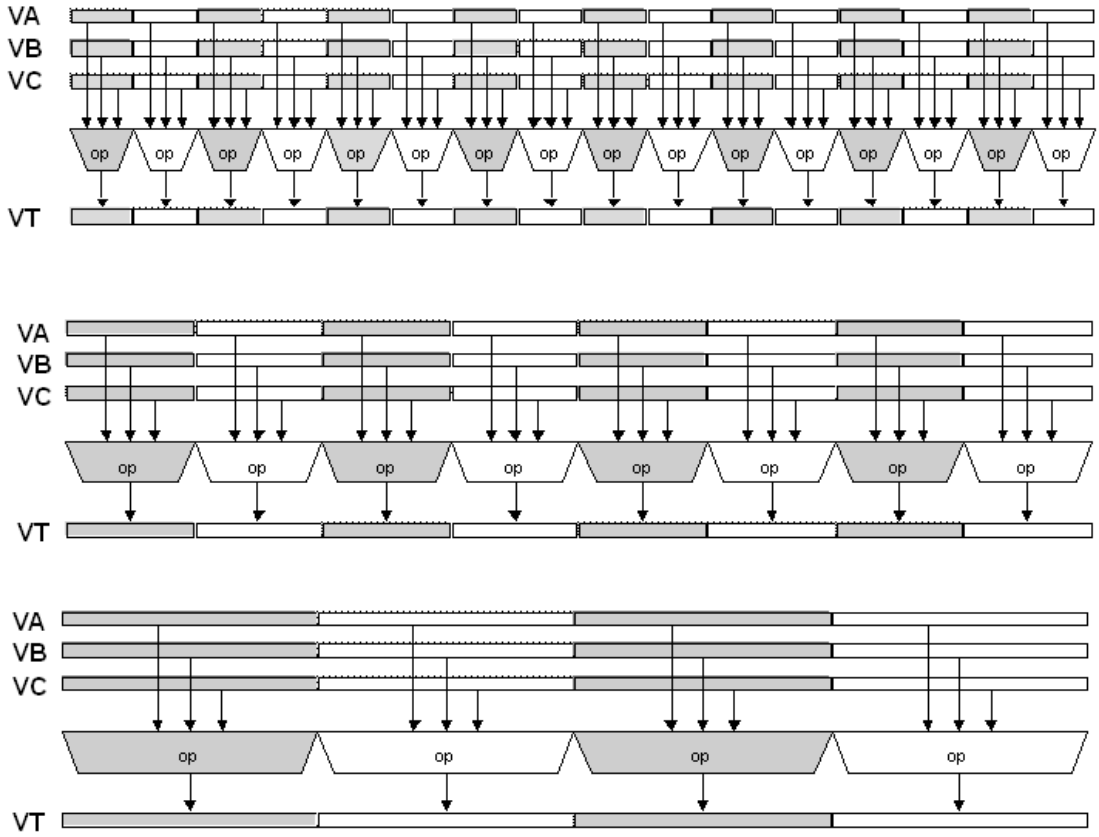

Figure 7. Altivec Degrees of Parallelism

3.2 MESI vs. MEI Comparison

Another important difference is that of the MEI cache coherency features on the IBM 750CX/CXE and the enhanced MESI capability of the MPC7447A. These protocols are used as a coherency mechanism in SMP (Symmetric Multi-Processing) configurations to indicate the relationship between 32-byte blocks stored in cache and their corresponding blocks in main memory. In an SMP system, some or all of the main memory is shared, therefore it is important to find the most efficient method of maintaining coherency across the caches and memory of the CPUs.

MEI refers to the cache coherency states available in the IBM 750CX/CXE:

- M – Modified – This block is modified with respect to main memory
- E – Exclusive – This block is valid and only present in this CPU's cache
- I – Invalid – This block is invalid with respect to main memory

It is best to illustrate the MEI protocol operation by way of an example. In a dual processor SMP system using IBM 750CX/CXE processors, we can refer to the processors, as well as CPU1 and CPU2, operating on a shared area of memory. If CPU1 loads a cache line from this area of main memory, it is marked as Exclusive with the assumption that we have flushed cache on both CPUs. If, however, CPU2 snooped the read request from CPU1 and already had it modified in its cache, then it would have changed its MEI status to Invalid and pushed the block into main memory, causing CPU1 to wait for and then read the latest version of the data. Then, if CPU2 tries to read the data again, it must read it from main memory and, to

make the situation worse, CPU1 may have since Modified the data in its cache. If CPU1 did modify the data, then CPU2 would have to wait for CPU1 to write its data back to memory for the CPU2 to access.

The extra bandwidth used and time wasted in waiting for each CPU to write its cache block back to memory for the other CPU to access is a very inefficient use of the bus. To help combat this problem, the MPC7447A supports the MPX bus, which extends the 60x functionality with some efficiency improvements, as discussed in the next section. The main method used to improve performance on MPC7447A was to incorporate the MESI protocol, which includes a new state:

- S – Shared – This block exists in multiple caches and is consistent with main memory (it is read only)

The addition of this state reduces the wasted time and bandwidth associated with MEI coherency and requires an additional 60x/MPX signal called ~SHD. If we look at the previous example, it is easy to see the benefits of the MESI over MEI. If CPU1 tried to read a block of main memory into its cache, CPU2 would snoop the transaction as before, but this time assert the ~SHD signal to tell CPU1 that it also has a cached copy of this block too. CPU1 would load the block into its cache with Shared status and CPU2 would change its cache entry to Shared from Exclusive, allowing both CPUs to access the data quickly from cache, providing they are only reading it.

3.3 MPX Mode

The MPX bus protocol is based on the 60x bus protocol. It also includes several additional features that allow it to provide higher memory bandwidth than the 60x bus and more efficient utilization of the system bus in a multiprocessing environment.

Memory accesses that use the MPX bus protocol are divided into address and data tenures. Each tenure has three phases—bus arbitration, transfer, and termination. The MPX bus protocol also supports address-only transactions. Note that address and data tenures can overlap. One of the key differences to the 60x bus is that the MPX does not require an idle cycle between tenures. To illustrate the importance of this difference, consider the following example:

- 100Mhz 60x bus:
  - Transfer rate = (32 bytes / 5 clock cycles) * 100MHz = 640MB/s
- 100Mhz MPX bus:
  - Transfer rate = (32 bytes / 4 clock cycles) * 100MHz = 800MB/s

Also, if you consider the higher bus speeds of 167MHz available on the MPC7447A, this figure is scaled accordingly to give significant increase to 1336MB/s—over 50% more than the IBM 750CXE with its 851.2MB/s maximum with a 133MHz 60x bus.

The address and data tenures in the MPX bus protocol are distinct from one another and each tenure consists of three phases—arbitration, transfer, and termination. The separation of the address and data tenures allows advanced bus techniques—such as split-bus transactions, enveloped transactions, and pipelining—to be implemented at the system level in multiprocessor systems.

The MPX bus mode’s support for data intervention and full data streaming for burst reads and writes is realized through the addition of two new signals—HIT and DRDY.

The HIT signal is a point-to-point signal output from the processor or local bus slave to the system arbiter. This signal indicates a valid snoop response in the address retry (ARTRY) window (the cycle after an address acknowledge (AACK) that indicates that the MPC7447A will supply intervention data). Intervention occurs when the MPC7447A has the data that has been requested by another master's bus transaction in its L1 or L2. Instead of asserting ARTRY and flushing the data to memory, the MPC7447A may assert HIT to indicate that it can supply the data directly to the other master. This external intervention functionality is disabled by MSSCR0[EIDIS]. The DRDY signal is also used by the MPX bus protocol to implement data intervention in the case of a cache hit. The SHD1 signal operates in conjunction with the SHD0 signal to indicate that a cached item is shared.

MPX mode offers one final improvement to the 60x by supporting out of order transactions. As mentioned previously, the MPC7447A supports up to 16 outstanding transactions, compared to the 2 supported by the IBM 750CX/CXE. This means that the MPC7447A has increased efficiency with its deeper pipeline of transactions. A further improvement specific only to MPX mode is that these transactions can be out of order, allowing lower latency devices to return data as soon as they are ready, without waiting for higher latency devices to return data first just because their transaction was first.

4 Programming Model

Both the IBM 750CX/CXE and MPC7447A have to support the PowerPC standard architecture in order to retain compatibility in user mode. Recompilation is not necessary for the IBM 750CX/CXE user code to execute properly on the MPC7447A. However, in supervisor mode there are many differences between device dependent registers. Even though some of the names are the same, the fields are often changed in name and/or bit position. There are also additional registers in different PowerPC implementations to support additional features. This section maps the supervisor level registers between IBM 750CX/CXE and MPC7447A, and points out any additional or device specific features.

The diagrams in Figure 8 and Figure 9 show the IBM 750CX/CXE and MPC7447A programming models respectively.

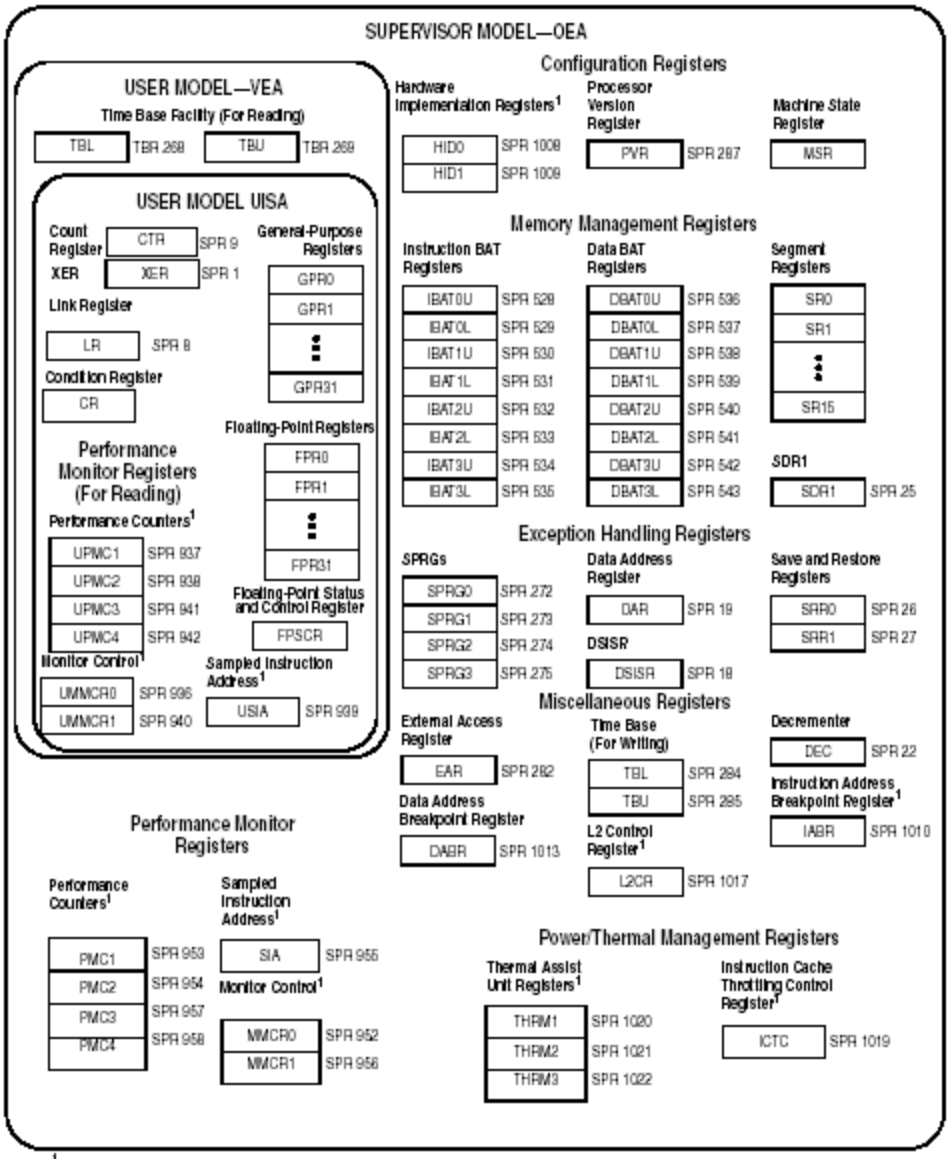

Figure 8. IBM 750CX/CXE Registers

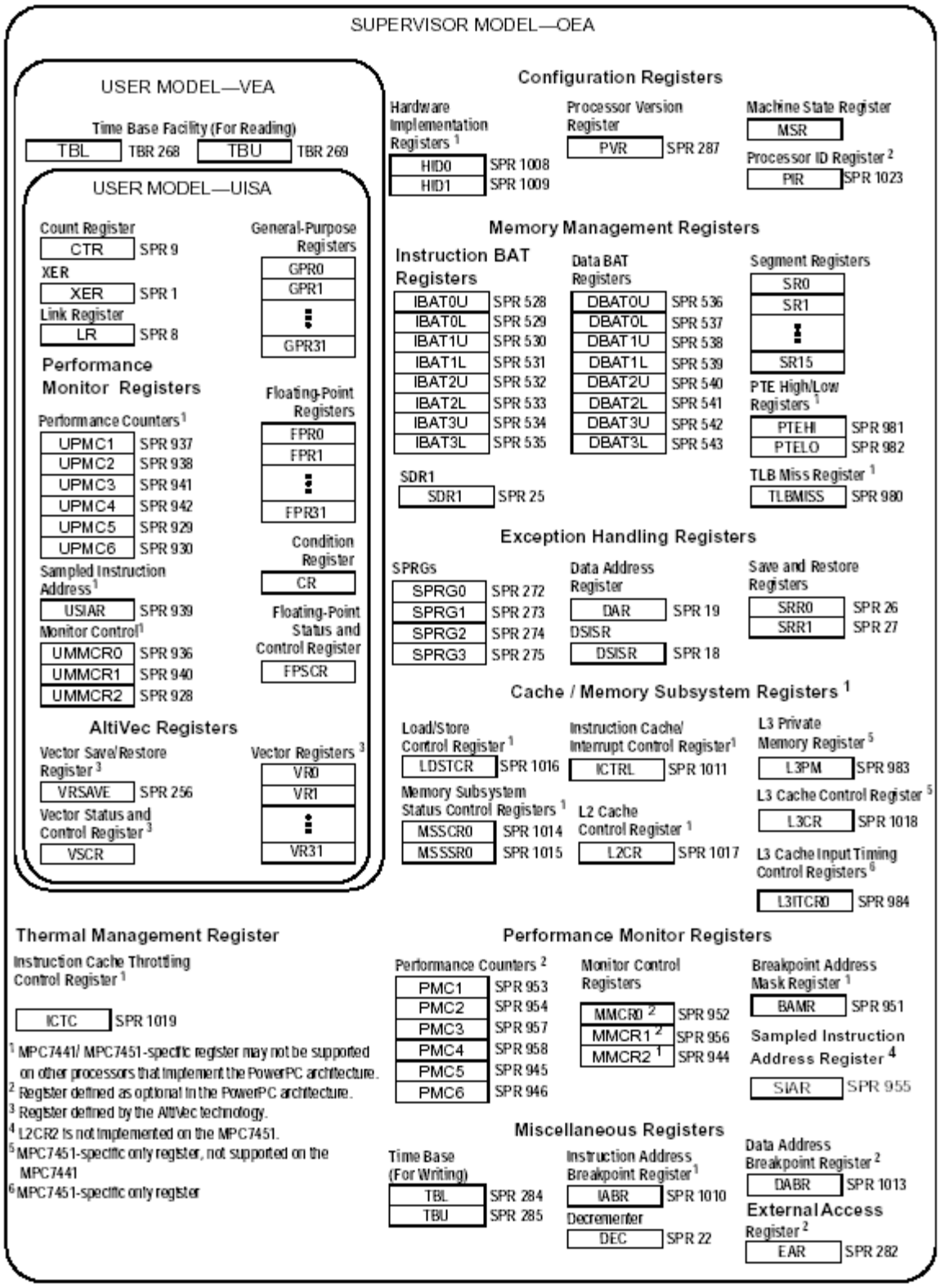

Figure 9. MPC7447A Registers

4.1 Differences in HID0 and HID1

Although both the IBM 750CX/CXE and MPC7447A have both of these registers defined in their implementation, they are optional to the standard and, therefore, differences in bit settings between devices do exist. The following table summarizes these differences and shows the mapping of fields between devices.

Table 4. IBM 750CX/CXE HID0 to MPC7447A Mapping

Function	IBM 750CX/CXE	MPC7447A
Enable MCP	HID0[EMCP]	HID1[EMCP]
Disable 60x bus address and data parity generation	HID0[DBP]	N/A [1]
Enable 60x bus address parity checking	HID0[EBA]	HID1[EBA]
Enable 60x bus data parity checking	HID0[EBD]	HID1[EBA]
Disable precharge of ~ARTRY	HID0[PAR]	HID1[PAR]
Doze mode enable	HID0[DOZE]	N/A [2]
Nap mode enable enable	HID0[NAP]	HID0[NAP]
Sleep mode enable enable	HID0[SLEEP]	HID0[SLEEP]
Dynamic power management enable	HID0[DPM]	HID0[DPM]
Read instruction segment register	HID0[RISEG]	N/A [3]
Miss-under-miss enable	N/A [4]	N/A [4]
Not a hard reset	HID0[NHR]	HID0[NHR]
Instruction cache enable	HID0[ICE]	HID0[ICE]
Data cache enable	HID0[DCE]	HID0[DCE]
Instruction cache lock	HID0[ILOCK]	HID0[ILOCK]
Data cache lock	HID0[DLOCK]	HID0[DLOCK]
Instruction cache flush invalidate	HID0[ICFI]	HID0[ICFI]
Data cache flush invalidate	HID0[DCFI]	HID0[DCFI]
Speculative data and instruction cache disable	HID0[SPD]	HID0[SPD]
Enable M bit on bus for instruction fetches (M from WIM states)	HID0[IFEM]	HID0[23] [5]
Store gathering enable	HID0[SGE]	HID0[SGE]
Data cache flush assist	HID0[DCFA]	HID0[25] [6]
BTIC enable	HID0[BTIC]	HID0[BTIC]
Address broadcast enable	HID0[ABE]	HID1[ABE] [7]

Table 4. IBM 750CX/CXE HID0 to MPC7447A Mapping (continued)

Function	IBM 750CX/CXE	MPC7447A
Branch History Table enable	HID0[BHT]	HID0[BHT]
No-op the data cache touch instructions	HID0[NOOPTI]	HID0[NOOPTI]

1 Not available in MPC7447A implementation.
2 Not required on MPC7447A due to processor-system handshake protocol system explained in Power Management.
3 Not available in MPC7447A implementation.
4 Always enabled in MPC7447A implementation. Not supported on the IBM 750CX/CXE.
5 Reserved. Used for IFEM in earlier processors but is also used for Extended BAT Block Size Enable.
6 Reserved. Defined as DCFA on earlier processors.
7 Must be enabled in multiprocessing systems. HID1[SYNCBE] enables address broadcast for sync and eieio instructions.

4.2 Power Management

Although they are very similar, there are a couple of differences between the IBM 750CX/CXE and MPC7447A power management functionality. This section only mentions the differences. Features like Instruction Cache Throttling, to slow the instruction dispatch rate, is the same in both implementations. Both implementations support the four states: Full Power, Doze, Nap, and Sleep.

From Table 4 above, you should note that the there is no HID0[DOZE] bit for the MPC7447A because the MPC7447A enters Doze mode when requested by the processor-system protocol. The processor can transition to Doze mode from:

1. Full Power, if HID0[NAP] or HID0[SLEEP] is asserted and the core is idle.
2. Nap, if the system negates ~QACK to signal a snoop operation is outstanding.

It can transition from Doze mode to:

1. Full Power, following one of many possible interrupts: external, ~SMI interrupt, ~SRESET,~HRESET, machine check or decrementer interrupt.
2. Nap, if the system asserts ~QACK with HID0[NAP] set.

or

3. Sleep, if system asserts ~QACK with HID0[SLEEP] set.

Additionally, the MPC7447A has a Deep Sleep mode, which can offer further power savings from Sleep mode by setting PLL_CFG to 0xF, which turns off the PLL, hence allowing the SYSCLK source to be disabled.

For further explanation on standard power management features between both implementations, please refer to the *MPC7447A Reference Manual*.

4.2.1 PLL Configuraton

HID1 primarily holds PLL configuration and other control bits in both the IBM 750CX/CXE and MPC7447A although there are a couple of differences, as shown below, due to the PLL in the IBM 750CX/CXE versus the Dynamic Frequency Selection (DFS) found in the MPC7447A, but not other current MPC7450 family device. For this reason, there is not a direct mapping between the two. The concept behind DFS is to save power through reducing the core clock rate when full rate is not required.

4.2.1.1 PLL Configuration

Table 5 below shows the fields in HID1 required to read the PLL configuration.

Table 5. IBM 750CX/CXE HID1/Dual PLL Settings

Function	IBM 750CX/CXE
PLL configuration, PLL_CFG[0-3] (Read only)	HID1[0-3]

4.2.1.2 DFS Configuration

The configuration of DFS is very simple and allows the core clock frequency to be halved.

To illustrate the simplicity of the DFS features:

1. The frequency is switched completely “on the fly”.
2. This change occurs in only one clock cycle.
3. It requires no idle time or operations before or during the transition.

Considering the following equation:

$P = (C \bullet V2 \bullet f) + P_{DS}$

Where: P = core power consumption

C = effective capacitance (approximately as a constant)

V = core voltage, V_{DD}

f = core frequency, f_{CORE}

P_{DS}= deep sleep mode power consumption

Excluding deep sleep mode power consumption, which is a minimum fixed power cost for an inactive core, the dynamic power consumption of the device is halved when DFS is applied. Note that static (leakage) power is not affected by DFS, so the power consumption with DFS enabled is not exactly 50% of the full-power consumption. For more information, see Section 4.2.1.2, “DFS Configuration.” This provides a significant advantage in supporting dynamic processing requirements in power sensitive applications.

Table 6 shows the bits corresponding with DFS mode.

Table 6. MPC7447A HID1/DFS Settings

Feature	MPC7447A
DFS divide by two enable	HID1[DFS]
PLL configuration, PLL_CFG[0-4] (Read only)	HID1[PC0-PC4]

Please note that only integer clock multipliers are permitted (for example, 9x halved to 4.5x). The PLL does not support quarter multipliers (8.5x halved to 4.25x, for example). Also, note that the halved frequency must meet the minimum clocking frequency specified for the part.

4.3 Cache / Memory Subsystem Configuration

The MPC7447A implements two registers named Memory Subsystem and Status Control Registers, MSSSR and MSSCR, that do not exist in the IBM 750CX/CXE. Some of the functions in these extra registers are held in other IBM 750CX/CXE registers. Table 7 summarizes this relationship.

Table 7. IBM 750CX/CXE Mapping to MPC7447A MSSSR, MSSCR Registers

Function	IBM 750CX/CXE	MPC7447A
Address bus parity error	SRR1[AP]	MSSSR[APE]
Data bus parity error	SRR1[DP]	MSSSR[DPE]
Bus transfer error acknowledge	SRR1[TEA]	MSSSR[TEA]

In addition, MSSCR stores more configuration data. This configuration relates to features not available in the IBM 750CX/CXE, including L3 cache parameters for MPC745x devices, and also defines the number of outstanding bus transactions, MSSCR[DTQ], and intervention for MPX mode, MSSCR[EIDIS].

4.4 Differences in L1 and L2 Cache Configuration

Due to the differences in each programming model, the L1 and L2 cache configuration and status bits are located in different registers in the MPC7447A and the IBM 750CX/CXE.

The MPC7447A has some features for cache parity detection and reporting which are not present on the IBM 750CX/CXE. As you can see in Table 8, these functions are spread across SSR1, which the IBM 750CX/CXE has, but the bits in question are reserved, as well as the MSSSR, Instruction Cache, and Interrupt Control Register, ICTRL, which the IBM 750CX/CXE does not have.

Table 8. MPC7447A Cache Parity Settings

Function	IBM 750CX/CXE	MPC7447A
L1 instruction-cache/instruction-tag parity error status/mask	N/A [1]	SRR1[1]
L1 data-cache/data-tag parity error status/mask	N/A[1]	SRR1[2]
L2 tag parity error status/mask	N/A[1]	MSSSR[L2TAG] – Tag error (MSSSR[L2DAT]) – Data error
Enable L1 instruction-cache/ instruction-tag parity checking	N/A[1]	ICTRL[EICE] (ICTRL[EICP]) [2]
Enable L1 data-cache/data-tag parity checking	N/A[1]	ICTRL[EDEC]
Enable L2 tag parity checking	N/A[1]	L2CR[L2PE] [3]

1 Not available in IBM 750CX/CXE implementation.

2 When the EICP bit is set, the parity of any instructions fetched from the L1 instruction cache are checked. Any errors found are reported as instruction cache parity errors in SRR1. If EICE is also set, these instruction cache errors cause a machine check or checkstop. If either EICP or EICE is cleared, instruction cache parity is ignored.
Note that when parity checking and error reporting are both enabled, errors are reported even on speculative fetches that are never actually executed. Correct instruction cache parity is always loaded into the L1 instruction cache, regardless of whether checking is enabled or not.

3 Enables tag AND data parity.

Table 9 shows the mapping of the IBM 750CX/CXE's L2CR to the MPC7447A.

Table 9. IBM 750CX/CXE L2CR to MPC7447A Mapping

Function	IBM 750CX/CXE	MPC7447A
L2 cache enable	L2CR[L2E]	L2CR[L2E]
L2 double bit checkstop enable	L2CR[CE]	N/A [1]
L2 data only	L2CR[DO]	L2CR[DO]
L2 global invalidate	L2CR[L2II]	L2CR[L2I]
L2 write through	L2CR[L2WT]	N/A [1]
L2 test support	L2CR[L2TS]	N/A [1]
L2 cache way locking	N/A [2]	L2CR[D0] and L2CR[IO]
L2 instruction only	N/A [2]	L2CR[IO]
L2 global invalidate progress bit	L2CR[L2IP]	N/A [1]

1 Not available in MPC7447A implementation.

2 Not available in IBM 750CX/CXE implementation

4.4.1 MPC7450 Extended Capabilities

The MPC7447A also offers the choice of the first or second replacement algorithm, L2CR[L2REP], and an L2 hardware flush feature, L2CR[L2HWF], which the IBM 750CX/CXE does not.

An L2 feature supported on the MPC7447A family, but not the IBM 750CX/CXE, is L2 prefetching. This can offer an improvement in performance by loading the second block of a cache line after a cache miss on the line. The idea being that the second block may be required in the near future, even if it is not required right now. The MPC7447A family takes advantage of this concept, known as spatial locality, using up to three hardware prefetch engines.

The L2 prefetching feature can be enabled by setting the L2 prefetch enable bit in memory configuration subsystem register, MSSCR0[PFE], providing the L2 cache is enabled and not configured as data or instruction only.

4.4.2 L1 and L2 Cache Locking

The MPC7447A contains a Load/Store Control Register, which configures L1 data cache locking by way. The LDSTCR is not present in the IBM 750CX/CXE because it is not supported. It can be configured on the MPC7447A using the 8 bits in LDSTCR[DCWL], indicating which way(s) to lock.

Similarly, ICTRL is also not present on the IBM 750CX/CXE, since its ICTRL[ICWL] is used to lock the L1 instruction cache by way, which is not supported in the IBM 750CX/CXE.

The IBM 750CX/CXE cannot lock L2 cache by way or completely. The MPC7447A does not support locking by way, but the whole cache can be locked by setting both L2CR[DO] and L2CR[IO].

4.5 Memory Management Registers

Since the IBM 750CX/CXE does not have the ability to resolve page table entries in software, it has no need for PTEHI, PTELO, and TLBMISS registers known as SPR 981, 982, and 980, respectively.

The TLBMISS register is automatically loaded when software searching is enabled (HID0[STEN] = 1) and a TLB miss exception occurs. Its contents are used by the TLB miss exception handlers (the software table search routines) to start the search process.

The PTEHI and PTELO registers are used by the **tlbld** and **tlbli** instructions to create a TLB entry. When software table searching is enabled and a TLB miss exception occurs, the bits of the page table entry (PTE) for this access are located by software and saved in the PTE registers.

A full explanation of software page table searching can be found in the *MPC7447A Reference Manual*.

4.6 Performance Monitor

Although it is optional, both implementations support the performance monitor features. It gives the user software the ability to monitor and count specific events, including processor clocks, L1 and L2 cache misses, types of instructions dispatched, and branch prediction statistics, among others. The count of these events can be used to trigger an exception.

The performance monitor has three key objectives:

- To increase system performance with efficient software, especially in a multiprocessing system—Memory hierarchy behavior can be monitored and studied in order to develop algorithms that schedule tasks (and perhaps partition them) and structure and distribute data optimally.
- To characterize processors—Some environments may not be easily characterized by a benchmark or trace.
- To help system developers bring up and debug their systems.

The MPC7447A contains two additional performance counters, PMC5 and PMC6, a breakpoint address mask register, BAMR, and an extra monitor control register, MMCR2. This section looks at any differences in the common registers and the purpose of the extra MPC7447A registers. The MPC7447A offers the extra registers to monitor more events, including Altivec-based events, which the IBM 750CX/CXE obviously does not have to support. Full listings of PMC events available in each implementation can be found in *IBM PowerPC 750CX/CXE RISC Microprocessor User Manual* and *MPC7450 RISC Microprocessor Family User's Manual*.

Each implementation provides read registers in user mode for PMC and MMCR registers with the prefix U (for example, UPMC1, UMMCR1, and so on).

4.6.1 Monitor Mode Control Registers

The mapping between the MMCR0 and MMCR1 is very similar, but not identical. Table 10 and Table 11 show this mapping for the IBM 750CX/CXE MMCR0 and MMCR1, respectively.

Table 10. IBM 750CX/CXE MMCR0 to MPC7447A

Function	IBM 750CX/CXE	MPC7447A
Disable counting unconditionally	MMCR0[DIS]	MMCR0[FC]
Disable counting while in supervisor mode	MMCR0[DP]	MMCR0[FCS]
Disable counting while in user mode	MMCR0[DU]	MMCR0[FCP]
Disable counting while MSR[PM] is set	MMCR0[DMS]	MMCR0[FCM1] [1]
Disable counting while MSR[PM] is zero	MMCR0[DMR]	MMCR0[FCM1] [1]
Enable performance monitor interrupt signaling	MMCR0[ENINT]	MMCR0[PMXE]
Disable counting of PMC*n* when a performance monitor interrupt is signalled	MMCR0[DISCOUNT]	MMCR0[FCECE] [2]
64 bit time base transition selector	MMCR0[RTCSELECT]	MMCR0[TBSEL]
Enable interrupt when RTCSELECT defined bit transitions off/on	MMCR0[INTONBITTRANS]	MMCR0[TBEE]

Table 10. IBM 750CX/CXE MMCR0 to MPC7447A (continued)

Function	IBM 750CX/CXE	MPC7447A
Threshold value, 0-63, which can be varied to get to characterize the events occurring above the threshold	MMCR0[THRESHOLD]	MMCR0[THRESHOLD]
Enable interrupt due to do PMC1 overflow	MMCR0[PMC1INTCONTROL]	MMCR0[PMC1CE]
Enable interrupts due to PMCn overflow	MMCR0[PMCINTCONTROL]	MMCR0[PMC*n*CE] [3]
Trigger counting of PMC2-4 after PMC1 overflows or after a interrupt is signalled	MMCR0[PMCTRIGGER]	MMCR0[TRIGGER] [4]
PMC1 event selector, 128 events	MMCR0[PMC1SELECT]	MMCR0[PMC1SEL]
PMC2 event selector, 64 events	MMCR0[PMC2SELECT]	MMCR0[PMC2SEL]

1 MSR[PM] on the IBM 750CX/CXE corresponds to MSR[PMM] on the MPC7447A.
2 For all PMCs, not just PMCn.
3 Enable overflow interrupts on PMC[1-4] for IBM 750CX/CXE and PMC[1-6] for MPC7447A.
4 Trigger counting of PMC[2-6] for MPC7447A.

Table 11. IBM 750CX/CXE MMCR1 to MPC7447A

Function	IBM 750CX/CXE	MPC7447A
PMC3 event selector, 32 events	MMCR1[PMC3SELECT]	MMCR1[PMC3SEL]
PMC4 event selector, 32 events	MMCR1[PMC4SELECT]	MMCR1[PMC4SEL]
PMC5 event selector, 32 events	N/A [1]	MMCR1[PMC5SEL]
PMC6 event selector, 64 events	N/A [1]	MMCR1[PMC6SEL]

1 PMC5 and PMC6 not present in IBM 750CX/CXE.

As mentioned previously, the MPC7447A also has a MMCR2 register with a one bit field, MMCR2[THRESHMULT]. This can be used to extend the range of the MMCR0[THRESHOLD] field by multiplying by 2 if set at 0, or by 32 if set at 1.

The MPC7447A also has a breakpoint address mask register, BAMR, that is used as a mask for debug purposes to compare to IABR[0:29] when PMC1 is set to monitor event 42. This event monitors for IABR hits specifically by checking they match BAMR.

For example:

```
Match = ((IABR[0–29] & BAMR[0–29]) ==
(completion_address[0–29] & BAMR[0–29]))
```

5 Hardware Considerations

5.1 Pin-out Comparison

Since there is no footprint/pin-out compatibility, the easiest way to compare the IBM 750CX/CXE and MPC7447A pins is to look at the different pins on the IBM 750CX/CXE that do not exist on the MPC7447A, and then to look at the pins present on the MPC7447A but not on the IBM 750CX/CXE.

5.1.1 IBM 750CX/CXE Uncommon Pins

Table 12 shows the signal name, pin number, and a description of the signal.

Table 12. IBM 750CX/CXE Additional Signals

Signal Name	Pin Number	Active	I/O	Description
~DBWO/L2_TSTCLK	E3	Low	I	Data bus write only
~BR	N1	Low	O	Address arbitration
OVdd	24 total, see 750CX/CXE Datasheet	High		Supply for Receiver/Drivers NOTE – This is the same as VDD_SENSE in MPC7447A

5.1.2 MPC7447A Uncommon Pins

Table 13 shows the signal name, pin number, and a description of the signal.

Table 13. MPC7447A Additional Signals

Signal Name	Pin Number	Active	I/O	Description
~BMODE0	G9	Low	I	Bus mode select 0
~BMODE1	F8	Low	I	Bus mode select 1
~DRDY	R3	Low	O	Data ready output signal to system arbiter
DTI[0:3]	G1, K1, P1, N1	High	I	Data transfer index (for outstanding bus transactions)
EXT_QUAL	A11	High	I	Extension qualifier
GND_SENSE	G12, N13			Internally connected to GND allowing an external device to know core ground level.
~HIT	B2	Low	O	MPX support for cache to cache transfers and local bus slaves.
Ovdd	E18, G18			Supply voltage connection for system interface
~PMON_IN	D9	Low	I	Transitions counted by PMC1, event 7
~PMON_OUT	A9	Low	O	Asserted when any performance monitor threshold or condition occurs regardless of whether exceptions are enabled or not

Table 13. MPC7447A Additional Signals (continued)

Signal Name	Pin Number	Active	I/O	Description
~SHD[0:1]	E4, H5	Low	I/O	Assertion indicates processor contains data from the snooped address. Second SHD signal required for MPX bus mode.
TEMP_ANODE	N18			Anode from internal temperature diode
TEMP_CATHODE	N19			Cathode from internal temperature diode
TEST[0:3]	A12, B6, B10, E10		I	For internal factory test. Should be pulled up to OVdd for normal operation.
TEST[4]	D10		I	For internal factory test. Should be pulled down to GND.
VDD_SENSE	G13, N12			Internally connected to OVdd allowing an external device to know I/O voltage level. (Were OVdd in earlier MPC74xx implementations) NOTE – This is the same as OVdd in IBM 750CX/CXE

5.2 60x Signal Differences

One of the changes in terms of hardware between the IBM 750CX/CXE and MPC7447A is that the MPC7447A does not support 3.3V I/O. It only supports 1.8V and 2.5V, as shown in Table 14.

Table 14. Supported I/O Voltages

Voltage Level	IBM 750CX/CXE	MPC7447A
1.8V	BVSEL=0, L1TSTCLK=1	BVSEL=0
2.5V	BVSEL=1, L1TSTCLK=1	BVSEL=1
3.3V	BVSEL=1, L1TSTCLK=0	N/A

Table 15 shows some of the differences in 60x signals between the IBM 750CX/CXE and MPC7447A. The IBM 750CX/CXE contains some optional 60x signals that are not implemented in the MPC7447A; all other 60x signals are the same.

Table 15. 60x Signal Differences

Signal Description	IBM 750CX/CXE	MPC7447A
60x bus mode select	Default	~BMODE0=VDD ~BMODE1=VDD
Address bus	A[0:31]	A[0:35] [1]
Address parity	AP[0:3]	AP[0:4] [2]
Address parity error	~APE	N/A
Transaction burst	~TBST (input/output)	~TBST (output)
Cache inhibited	~CI (output)	~CI (output)
Write through	~WT (output)	~WT (output)
Data bus write only	~DBWO	N/A
Data parity error	~DPE	N/A

1 Use A[4-35] for 32 bit addressing, with A[0-3] pulled down if not in use.

2 In 32-bit mode, AP[0] should be pulled up. In 36-bit mode, use AP[0:4] as follows:
- AP[0] contains odd parity for A[0:3].
- AP[1] contains odd parity for A[4:11].
- AP[2] contains odd parity for A[12:19].
- AP[3] contains odd parity for A[20:27].
- AP[4] contains odd parity for A[28:35].

In the MPC7447A, ~BMODE1 is sampled after ~HRESET is negated to set the processor ID in MSSCR0[ID]. The value of the processor ID is important in a multiprocessor system where one would want to define one processor with the value 0 by negating ~BMODE1 and make that processor responsible for booting and configuring other processors and system logic. Other processors would have ~BMODE1 tied high to differentiate. In this case, the processor 0 could also configure the other processors Processor ID Register, PIR, with unique values within the system.

Another important point to make is the fact the MPC7447A supports up to 16 pipelined transactions configured by MSSCR[DTQ]. Since it does not support out of order transactions, hence no ~DBWO, the Data Transaction Index, DTI[0:3], should be pulled low.

6 Revision History

Table 16 provides a revision history for this application note. Note that this revision history table reflects the changes to this application note template, but can also be used for the application note revision history.

Table 16. Document Revision History

Revision Number	Date	Substantive Change(s)
1	06/22/2005	Minor editing
0	12/29/2004	Initial release

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

email:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com

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

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: AN2809
Rev. 1
06/2005

freescale™
semiconductor