AN2424/D
Rev. 0, 12/2002

MPC7410 and MPC7450: Comparison and Compatibility

Raiyan T. Zaman
CPD Applications

Freescale Semiconductor, Inc.

The primary objective of this application note is to describe the differences in functional timing between the MPC7410 and MPC7450 microprocessors. This document also compares the fetch, dispatch, execute, finish, and complete stages in both microprocessors, including a detailed description of the different execution units and their interaction with other functional blocks of each processor. This application note covers the following topics:

1 Overview

The MPC7410 microprocessor is an implementation of a family of reduced instruction set computer (RISC) microprocessors that implement the 32-bit PowerPC architecture and AltiVec™ technology. One goal of the MPC7410 is to achieve compelling multimedia performance. The scalar pipelines of the MPC7410 are similar to those of the MPC750, with a five-stage pipeline model—fetch, decode/dispatch, execute, complete, and write-back. In the MPC7410, double- and single-precision versions of floating-point instructions incur equal latency in multiplication.

The MPC7450 also implements the 32-bit PowerPC architecture. It has a seven-stage superscalar pipeline model—fetch1, fetch2, decode/dispatch, issue, execute, complete, and write-back. The MPC7450 has an extra fetch stage and an issue stage between dispatch and execute compared to the MPC7410.

To improve throughput, both microprocessors implement pipelining, superscalar instruction dispatch, branch folding, removal of fall-through branches, and multiple execution units that operate independently and in parallel. However, the MPC7410 handles two levels of speculative branching, whereas the MPC7450 handles three.

The main differences between the MPC7410 and MPC7450 microprocessors are different resource sizes, the use of issue queues, and the addition of a finish stage between completion and execution stages. For more details on the particular execution stages, see Chapter 6, “Instruction Timing,” in the *MPC7410 RISC Microprocessor User’s Manual* and the *MPC7450 RISC Microprocessor Family User’s Manual*.

© Freescale Semiconductor, Inc., 2004. All rights reserved.

For More Information On This
Go to: www.freescale

Figure 1 and Figure 2 show the flow of instructions through the superscalar pipeline of the MPC7410 and the MPC7450 microprocessors, respectively. Figure 1 and Figure 2 also list the high-level instruction timing features of the MPC7410 and MPC7450 microprocessors.

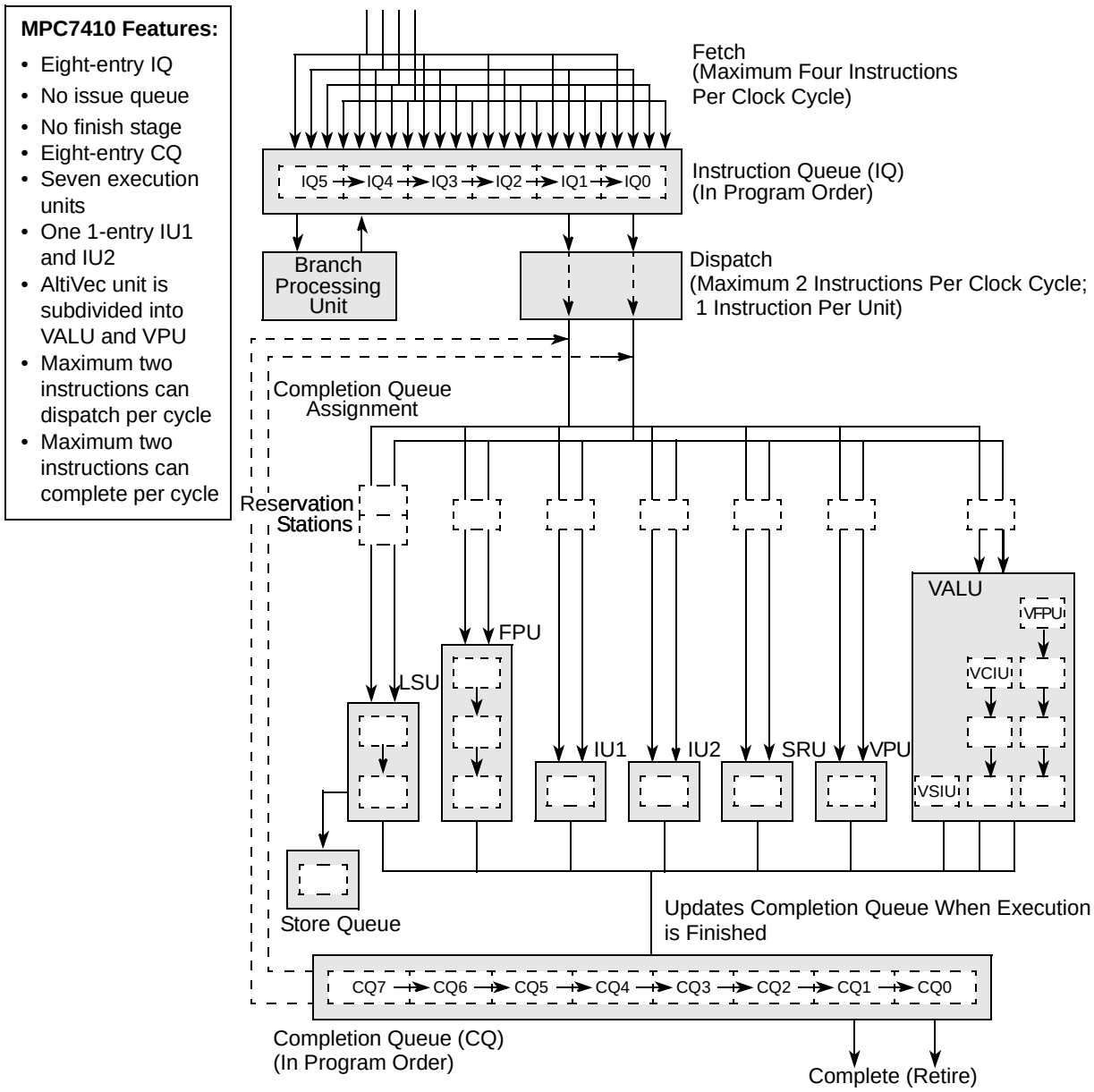

Figure 1. MPC7410 Instruction Flow Diagram

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Unique Features of MPC7450:

- 12-entry IQ
- Three issue queues
- Three finish stages
- 16-entry CQ
- Ten execution units
- Three separate 1-entry IU1s and one 3-stage IU2
- AltiVec unit is subdivided into four independent execution units
- Maximum three instructions can dispatch per cycle
- Maximum three instructions can complete per cycle

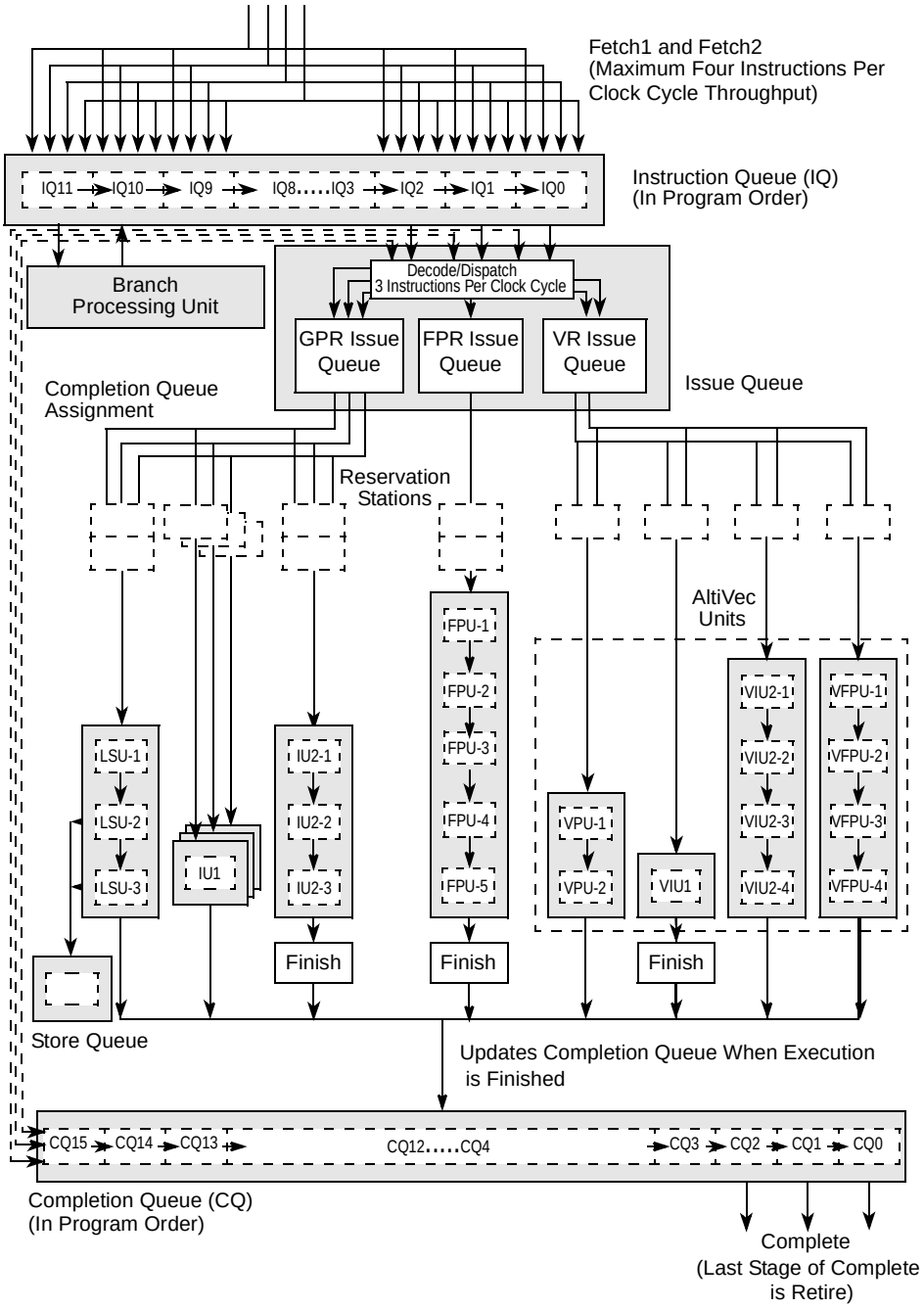

Figure 2. MPC7450 Instruction Flow Diagram

As an instruction passes from execution stage to execution stage within an execution unit, a subsequent instruction can follow through the stages as the former instruction vacates them, allowing several instructions to be processed simultaneously. Although it may take several cycles for an instruction to pass through all of the required stages when the pipeline is full, one instruction can be completed on every clock cycle. In other words, one instruction per cycle throughput is achieved in a given execution unit.

Figure 3 and Figure 4 show relationships between stages and events in the MPC7410 and MPC7450, respectively. The stages are labeled above the pipeline model and the starting points of the events are marked with arrows.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

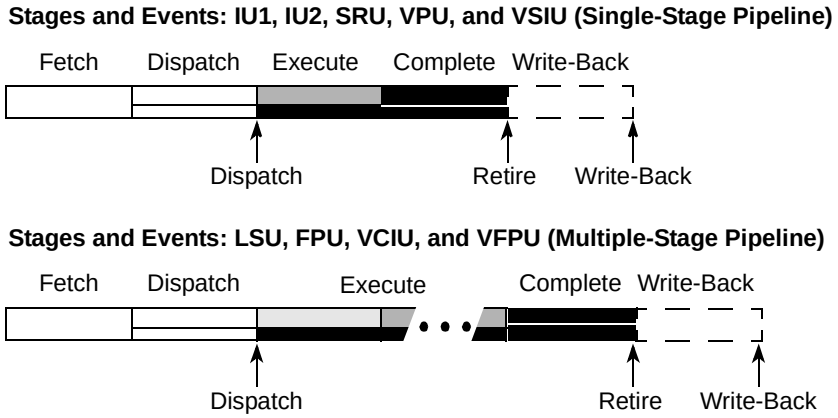

Figure 3. MPC7410 Microprocessor Pipeline Stages and Events

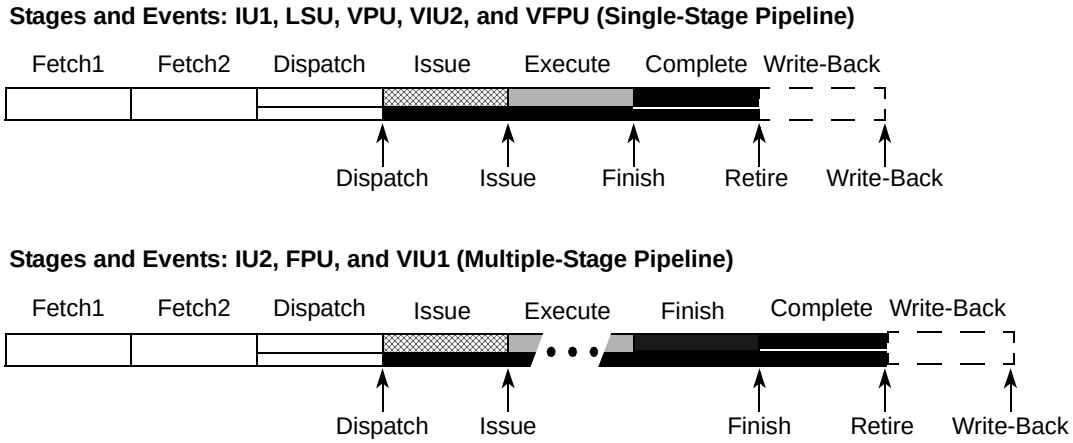

Figure 4. MPC7450 Microprocessor Pipeline Stages and Events

A pipeline can be described as either a physical entity or a sequence of events. In the latter case, a stage is a period of time during which certain actions are performed, such as decoding the instruction, performing an arithmetic operation, or writing back the results. Typically, the latency of a stage is one microprocessor clock cycle.

Some events, such as dispatch, retire, and write-back, happen instantaneously and may be described as occurring at the end of a stage. For example, the retire must occur at the end of the complete stage; an instruction is retired after it has updated architecture-defined registers with its results and is removed from the completion queue.

See Figure 6-1, in Chapter 6, “Instruction Timing,” of the *MPC7410 RISC Microprocessor User’s Manual* and *MPC7450 RISC Microprocessor Family User’s Manual* for more detail.

An instruction can spend multiple cycles in one stage. An integer multiply, for example, requires multiple cycles in the execute stage. When this occurs, subsequent instructions in that pipeline may stall. An instruction can also occupy more than one stage simultaneously, especially in the sense that a stage describes a physical resource—for example, when instructions are dispatched they are assigned an entry in the completion queue (CQ) at the same time they are passed to the issue queues.

Freescale Semiconductor, Inc.

2 High-Level Comparison

The instruction timing comparisons between the MPC7410 and MPC7450 microprocessors are divided into the following four categories:

- Instruction fetch timing
- Branch prediction
- Instruction dispatch, issue, and reservation station considerations
- Execution, rename, finish, completion, and retire considerations

2.1 Instruction Fetch Timing

Instruction fetching is the process of bringing instructions from memory (such as a cache or system memory) and placing them in the entries of the IQ. The latency associated with accessing an instruction depends on whether the instruction is in the branch target instruction cache (BTIC), on-chip caches, off-chip L3 cache (MPC7450 only), or system memory (in which case latency is further affected by bus traffic, bus clock speed, and address translation issues).

A comparison of MPC7410 and MPC7450 instruction fetch timing is listed in Table 1. Table 1 is divided into four sub-categories—general information, cache arbitration, cache hit, and cache miss.

Table 1.Comparison of Instruction Fetch Timing

Features	MPC7410	MPC7450
General Information		
Number of instructions that microprocessor can hold in IQ and CQ	12 instructions	24 instructions
Maximum number of instructions in the IQ	Six instructions	12 instructions
Pipeline control	Five-stage pipeline control—fetch, decode/dispatch, execute, complete, and write-back.	Seven-stage pipeline control—fetch1, fetch2, decode/dispatch, execute, issue, complete, and write-back.
Fetch stage is primarily involved in retrieving instructions	One fetch stage—fetch	Two fetch stages—fetch1 and fetch2
Maximum number of instructions that can be fetched from the instruction cache in one cycle	Four instructions	
Fetch latency	Depends on whether the fetch hits the BTIC, instruction MMU, L1 instruction cache, or L2 cache.	Depends on whether the fetch hits the BTIC, instruction MMU, L1 instruction cache, L2 cache, or off-chip L3 cache.
Required latency for cache block boundary	Needs one cycle, as cache block boundary does not break the four instructions into multiple accesses.	Needs at least two cycles, as cache block boundary breaks the four instructions into two accesses.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 1.Comparison of Instruction Fetch Timing (continued)

Features	MPC7410	MPC7450
Cache Arbitration		
Reasons for refetch	Instructions cannot be fetched if cache is busy due to a cache-line-reload operation	Instructions cannot be fetched if cache or MMU is busy due to a higher priority operation, including reload.
Cache Hit		
Word length	Critical-double-word is written simultaneously to cache and forwarded to the IQ to minimize stall	Critical-quad-word is written simultaneously to cache and forwarded to the IQ to minimize stall
Cache Miss		
Effect of cache miss	If no cache hit occurs, a memory transaction is required, affecting fetch latency due to bus traffic, bus clock speed, and memory translation.	If no cache hit occurs, a memory transaction is required, affecting fetch latency due to bus traffic, and bus clock speed.

2.2 Branch Prediction Comparison

Branch prediction is the process of guessing the direction, or target, of a conditional branch. Although, branch direction prediction (performed by the branch prediction unit (BPU)) involves guessing whether a branch will be taken, target prediction involves guessing the target address of a **bclr** branch. The PowerPC architecture also defines a means for static branch prediction as part of the instruction encoding.

If a branch is predicted as taken, all instructions are flushed from the instruction queue (IQ) in the next clock cycle. If the branch is unconditional, or if the specific conditions are already known, the branch can be resolved immediately. If the branch direction and target address are not known, they must be predicted. To reduce branch mispredictions, instructions that set CR bits should be separated from the branch instructions. Both microprocessors offer resources to resolve branch instructions and improve the accuracy of branch predictions. Table 2 describes differences and similarities in handling branch instructions between the two microprocessors.

Table 2.Comparison of Branch Prediction Unit Features

Features	MPC7410	MPC7450
General Information		
BTIC	64-entry	128-entry
Branch Target Instruction Cache		
Resources	BTIC, dynamic branch prediction, and static branch prediction.	BTIC, dynamic branch prediction, static branch prediction, and link register stack.
Levels of speculation	Two-level speculative branch handling	Three-level speculative branch handling
Instructions in BTIC	First two instructions in the target stream can be fetched into the IQ on the next clock cycle	First four instructions in the target stream can be fetched into the IQ on the next clock cycle
Number of cycle required for branch instructions to dispatch	First one or two instructions can dispatch on the next clock cycle	First four instructions can dispatch on the third clock cycle
BTIC invalidation	Disabled and invalidated through bits in HID0	

Freescale Semiconductor, Inc.

Table 2.Comparison of Branch Prediction Unit Features (continued)

Features	MPC7410	MPC7450
Number of idle cycles if instructions are not in BTIC	One idle cycle is required if the target instructions are in instruction cache instead of BTIC	Two idle cycles are required if the target instructions are in instruction cache instead of BTIC
Dynamic Branch Prediction		
Size of branch history table (BHT)	512-entry	2048-entry [1]
Branch direction and data dependency	Conditional branch direction cannot be resolved if this is a CR data dependency	Conditional branch direction cannot be resolved if this is a LR, CR, or CTR data dependency
Enabling dynamic branch prediction	Enabled through HID0[BHT]	
Link Stack Registers (LSR)		
LSR and its importance	No LSR	Stalls are avoided by implementing an eight-entry LSR
An **mtspr** (LR) followed by a **bclr**	Fetching stops and the branch waits for the **mtspr** to execute	The link stack is used to predict the instruction target address
Related Latency		
Latency in branch operation **bclr**	One-cycle	One- or two-cycle
Latency in condition register logical execution	One-cycle	Two-cycle

[1] Helps prevent aliasing in the BHT, which reduces prediction accuracy.

2.3 Dispatch, Issue, and Reservation Station Considerations

Several factors affect a microprocessor's ability to dispatch instructions—the mix of instructions, the availability of the execution unit, destination rename registers, and availability of CQ entries, as well as the handling of completion-serialized instructions. Note that the availability of issue queue entries apply only to the MPC7450.

The dispatch stage decodes instructions in the bottom entries of the IQ, renames any source/target operands, determines which issue queue (MPC7450 only) each non-branch instruction is dispatched, and determines whether the required space is available in both the issue (MPC7450 only) and completion queue. The decode/dispatch stage fully decodes each instruction; most instructions are dispatched to the execution unit reservation stations of the MPC7410 or to the issue queues of the MPC7450.

The issue stage (MPC7450 only) is responsible for reading source operands from rename registers and register files. This stage ends when instructions are assigned and routed to the proper execution unit. The issue stage also determines when instructions and their operands should be latched into the reservation station. For more information on the issue queues, see Section 6.2, "Instruction Timing Overview," in the *MPC7450 RISC Microprocessor Family User's Manual*.

Reservation stations are buffers that allow instructions to be dispatched (MPC7410 only) or issued (MPC7450 only) even though the results of other instructions on which the dispatched or issued instruction may depend, are not available. If a data dependency keeps an instruction from starting execution, that

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

instruction is issued to a reservation station and the rename registers are assigned, eliminating the dispatch unit stalls (MPC7410 only) or issue queue stalls (MPC7450 only) of the subsequent instructions.

Table 3 describes the different considerations of the dispatch and issue stages of the two microprocessors.

Table 3.Comparison of Dispatch, Issue, and Reservation Stations

Features	MPC7410	MPC7450
Dispatch Unit		
Maximum dispatch throughput per cycle	Two instructions	Three instructions
Instructions dispatch from IQ	Lowest two entries—IQ0 and IQ1	Lowest three entries—IQ0, IQ1, and IQ2
Number of resource requirements for dispatcher to avoid stall	Three resource requirements: • Rename check • Execution unit available check • CQ available check (CQ is not full) and in-order completion	Four resource requirements: • Issue queue available check • Rename check • Branch ready check • CQ available check (CQ is not full) and in-order completion
Importance of dispatcher	N/A	Dispatcher performs the following functions: • Locates source operands • Allocates rename registers for destination operands
To reduce dispatch unit stalls due to instruction data dependencies	• Single-entry reservation station for FPU, SRU, VPU, VALU, and each IU • Two-entry reservation station for LSU	• Issue queues • Single-entry reservation station for AltiVec units and each IU1s • Two-entry reservation stations for LSU, IU2, and FPU.
Instruction serialization limits dispatch	One instruction per cycle even though the microprocessor can dispatch two instructions per cycle	One instruction per cycle even though the microprocessor can dispatch three instructions per cycle
Types of instruction serialization	Five basic types—execution, store, sync, completion, and refetch.	Three basic types—execution, store, and refetch.
Issue Queue		
Number of issue queues	N/A	Three issue queues • Six-entry GPR issue queue (GIQ) • Four-entry vector issue queue (VIQ) • Two-entry floating-point issue queue (FIQ)
Number of instructions that can be accepted in the issue queues per cycle	N/A	Three instructions to various issue queues: • GIQ can accept three instructions (IU1, IU2, and all LSU, including floating-point and AltiVec loads and stores). • VIQ can accept two instructions (all AltiVec instructions other than load, store, and vector touch). • FIQ can accept one instruction (FPU).
Number of resource requirements for issue queues	N/A	One resource requirement: • An execution unit available check

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 3.Comparison of Dispatch, Issue, and Reservation Stations (continued)

Features	MPC7410	MPC7450
Advantage of issue queue	N/A	• Issue queue can hold decoded instructions even if execution units are busy • Issuing instructions out-of-order from GIQ can eliminate stalls
Reservation Stations		
Definition of reservation station	Reservation stations are buffers between the dispatch and execute stages	Reservation stations are buffers between the issue and execute stages
Number of instructions dispatched to the reservation stations per cycle	As many as two instructions can be dispatched to the reservation station associated with FPU, SRU, VPU, VALU, IU, and LSU execution units.	As many as three instructions can be issued in any order to the LSU, IU2, and IU1 reservation stations from the bottom three GIQ entries.

2.4 Execution, Rename, Finish, Complete, and Retire Considerations

The execution stage accepts instructions when the required reservation stations are not busy. Execution begins when all operands are available, the instruction is in the bottom reservation station, and any execution serialization requirements are met. This stage consists of the time between the acceptance of instruction by an execution unit (or reservation station) and the point at which the instruction vacates the execution unit.

The execution unit executes instructions (possibly over multiple cycles), writes results on its result bus, and notifies the CQ when the instruction finishes. The execute stage does not update architected registers. Both microprocessors can execute instructions out-of-order, but in-order completion by the completion unit ensures a precise execution mechanism. For more information on in-order completion, see Section 6.3.3.2, “Instruction Serialization,” in the *MPC7450 RISC Microprocessor Family User’s Manual*. The execution unit also reports any exceptions to the completion stage. Instruction-generated exceptions are not taken until the excepting instruction is next to retire.

When an instruction finishes execution, results are made available to subsequent instructions in the appropriate GPR, FPR, or VR rename registers, which are temporary buffers for holding results of instructions that have finished execution but have not completed. After results are available in the rename registers, they are stored in the correct GPR, FPR, or VR during the write-back stage. If a subsequent instruction needs the result as a source operand, the result is simultaneously made available to the appropriate execution unit, which allows a data-dependent instruction to be decoded and dispatched without waiting to read the data from the register file.

If no rename registers are available, instructions cannot decode or dispatch, and remain in the IQ until the required rename registers become available. Under most conditions, rename registers have sufficient bandwidth to allow dispatch of two or three instructions per clock cycle, depending on the microprocessor. For more information on rename registers see Section 6.3.3.1, “Rename Register Operation,” in the *MPC7450 RISC Microprocessor Family User’s Manual*.

The finish stage is only defined in the MPC7450 microprocessor. Note that not all execution units have a finish stage, although all of them have finish events. In this stage, an executed instruction finishes by notifying the CQ that execution is complete and results have been made available to subsequent instructions. Architecture-defined registers are not updated until the instruction is retired. For most execution units,

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

finishing occurs at the end of the last cycle of execution; however, for FPU, IU2, and VIU2, finishing occurs at the end of a separate, one-cycle stage after the final execution stage.

The completion stage follows the execution stage in the MPC7410 and the finish stage in the MPC7450 microprocessors. The completion unit maintains the correct program order after instructions are dispatched from the IQ with in-order completion and a precise exception model. In-order completion ensures the correct architectural machine state and commits the results to architecture-defined registers in proper order to allow proper recovery from a mispredicted branch or exception.

Completion logic detects program-related exceptions when the instruction causing the exception reaches the last position in the CQ. Prior instructions are allowed to complete before the exception is taken, and all following instructions are canceled. Also, their execution results in rename buffers are discarded, and the correct instruction stream is fetched. When all completion requirements are met and no dependencies exist, two or three instructions (depending on the microprocessor) can retire from the CQ in program order, and the results are written back to the architecture-defined registers in the clock cycle after retirement. Both microprocessors allow an instruction to finish and complete in the same cycle, if it resides in CQ0.

Table 4 describes the differences and similarities of the execution, finish, complete, and retire stages of the two microprocessors.

Table 4.Comparison of Execution, Rename, Finish, Complete, and Retire Stages

Features	MPC7410	MPC7450
Maximum number of instructions executed per cycle	Eight instructions in 8-entry CQ	Sixteen instructions in 16-entry CQ
Multiple-stage execution units	FPU, LSU, VCIU, VFPU	FPU, LSU, IU2, VIU2, VFPU, VPU
	LSU has two-stage pipeline	LSU has three-stage pipeline
	FPU has three-stage pipeline	FPU has five-stage pipeline
	IU2 has single-stage pipeline	IU2 has three-stage pipeline
Integer units	One single-cycle IU1—executes all integer instructions	Three single-cycle IU1s—executes all integer instructions except multiply, divide, and move to/from special-purpose register instructions.
	One single-cycle IU2—executes all integer instructions except multiply and divide	Multiple-cycle IU2—executes CR logical operations, integer multiply and divide, and move to/from special-purpose register instructions.
	Most integer instructions have a single-cycle execution latency	
Integer store gathering	Performs store gathering only for write-through operations to nonguarded space	Performs store gathering for write-through and cache-inhibited operations to nonguarded space
	Store gathering is not done for the following: • Byte-reverse store operations • A store that occurs during a table search operation • Little-endian and floating-point store operations	Store gathering is not done for the following: • Double-word to quad-word gathering if in 60x bus mode • Cache-inhibited or write-through stores if the result would violate MPX bus alignment

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Table 4.Comparison of Execution, Rename, Finish, Complete, and Retire Stages (continued)

Features	MPC7410	MPC7450
Floating-point unit	Most FPU instructions have three-cycle execution latency and one-cycle throughput	Most FPU instructions have five-cycle execution latency and one-cycle throughput
	fdivs, **fdiv**, and **fres** have 17–31 cycles latency.	**fdivs**, **fdiv**, and **fres** have 14–35 cycles latency.
	For best floating-point performance: • All exceptions should be disabled in the FPSCR and MSR • If an exception is enabled, the FPU takes an additional cycle to complete instructions.	For best floating-point performance: • IEEE floating-point exceptions should be disabled in the FPSCR and MSR • If an exception is enabled, the instruction traps to the program interrupt handler.
Load/store unit	Two-cycle latency and one-cycle throughput	GPR and vector load instructions have a three-cycle latency and one-cycle throughput. FPR load instructions have a four-cycle latency and one-cycle throughput.
AltiVec unit	Two AltiVec units—Permute unit (VPU) and arithmetic logical unit (VALU) consist of three subunits (VFPU, VSIU, VSIU).	Four independent units for vector computations—permute (VPU), simple integer (VIU1), complex integer (VIU2), and floating-point (VFPU).
	VFPU has a four-stage pipeline that is a part of VALU	VFPU has a four-stage pipeline
	All VFPU instructions have a four-cycle latency in non-Java mode and a five-cycle latency in Java mode	Most VFPU instructions have a four-cycle latency regardless of Java or non-Java mode
	VSIU has a single-stage pipeline that is a part of VALU	VIU1 has a single-stage pipeline
	VCIU has a three-stage pipeline that is a part of VALU	VIU2 has a four-stage pipeline
	VPU (separated from VALU) has a single-stage pipeline	VPU has a two-stage pipeline
Performance effects of memory operand placement	No effects on AltiVec and there is no constraint on using string instructions	Alignment exception occurs to non-cached memory because of AltiVec memory operations and usage of string instructions is strongly discouraged
Maximum number of instructions dispatched to AltiVec unit	One instruction to the VALU and one to the VPU	Two instructions to four independent AltiVec units
Rename registers	GPR, FPR, and VR each have six rename registers.	GPR, FPR, and VR each have 16 rename registers.
	CR, LR, and CTR have one rename register.	CR, LR, and CTR have one rename register.
Number of instructions renamed from dispatcher	Six instructions to each of GPRs, VRs, and FPRs.	Four, three, and two instructions to GPRs, VRs, and FPRs, respectively.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 4.Comparison of Execution, Rename, Finish, Complete, and Retire Stages (continued)

Features	MPC7410	MPC7450
Finish stage	N/A	Finish queue is associated with the following execution units—IU2, FPU, and VIU1.
Maximum number of instructions that can retire from the CQ per clock cycle	Two instructions	Three instructions
Requirements for instructions to retire simultaneously but in order	N/A	Instructions must be non-speculative to complete
		As many as three rename registers can be updated in a given register file
Write-back stage (in the context of instruction handling)	Write-back occurs when a result is written into the architectural registers (typically the GPRs, FPRs, and VRs).	The write-back stage occurs in the clock cycle after the instruction is retired

3 Conclusion

The MPC7410 and MPC7450 microprocessors have similar functional features relevant for instruction execution. However, the main differences between the execution of instructions in the MPC7410 and MPC7450 are different resource sizes, use of issue queues, and splitting of the completion and execution stages. With a longer basic pipeline compared to the MPC7410 microprocessor, the MPC7450 provides higher frequency. Also, the MPC7450 is optimized for minimizing the number of idle functional units and pipeline stalls. The longer pipeline of the MPC7450 is more sensitive to branch mispredictions, which causes some code sequences to run slower. The MPC7450 has more integer computing capacity available for scheduling because of the additional integer units.

4 Revision History

Table 5 provides a revision history for this application note.

Table 5.Revision History

Rev. No.	Substantive Change(s)
0	Initial release.

Freescale Semiconductor, Inc.

THIS PAGE INTENTIONALLY LEFT BLANK

Freescale Semiconductor, Inc.

THIS PAGE INTENTIONALLY LEFT BLANK

Freescale Semiconductor, Inc.

THIS PAGE INTENTIONALLY LEFT BLANK

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
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 or +81 3 5437 9125
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
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor

AN2424/D