**Freescale Semiconductor**
Application Note

AN2981
Rev. 0, 05/2005

# Migrating from the MCF5206e to the ColdFire® MCF5208

by: Melissa Hunter
TSPG Applications

The MCF5207 and MCF5208 processors are high-performance upgrades for the MCF5206e device. Leveraging the widespread industry success of the MCF5206e, the MCF5208 family is designed specifically to provide higher performance in embedded applications at a lower cost.

This application note describes what designers and engineers consider when migrating from the MCF5206e to the MCF5208 family of processors. For additional information about the MCF5208 family, refer to the *MCF5208 Reference Manual* (MCF5208RM).

Throughout each of the following sections, there will be frequent references to more detailed information, as this application note is not intended to be an all-encompassing reference for either of these two devices or families. A list of all suggested references is given at the end of this document.

To locate any published errata or updates for this document, refer to the web site at http://www.freescale.com/coldfire.

**Table of Contents**

1 Comparison Overview ..... 2
2 Device Differences ..... 4
2.1 Electrical Characteristics ..... 4
2.2 Performance ..... 10
2.3 Mechanical Characteristics ..... 11
2.4 External Signal Comparison ..... 12
2.5 ColdFire V2 Enhancements ..... 16
2.6 Chip Architecture Enhancements ..... 19
2.7 On-chip Module Differences ..... 20
2.8 Chip Configuration ..... 28
2.9 Interrupt Processing ..... 30
2.10 Software Porting ..... 31
3 Summary ..... 32
4 References ..... 32
5 Document Revision History ..... 32

**NOTE**

This document provides an overview of both the MCF5207 and MCF5208 microprocessors. It was written from the perspective of the MCF5208 device, but except where noted, the information can be applied to both processors.

This document contains information on a new product under development. Freescale reserves the right to change or discontinue this product without notice.

# 1 Comparison Overview

Many current users of the MCF5206e will find several advantages in migrating to the MCF5208. Migration from the MCF5206e to the MCF5208 includes the following increases in device performance and functionality:

- Maximum CPU operating speed increases from 54 MHz to 166 MHz.
- Maximum system bus frequency increases from 54 MHz to 83 MHz.
- Double the amount of on-chip SRAM (16 Kbytes) and cache (8 Kbytes).
- 8-Kbyte cache is configurable as instruction, data, or split cache, unlike the MCF5206e's instruction-only cache.
- Enhanced multiply accumulate unit (EMAC) over the original multiply accumulate unit (MAC).
- Addition of three extra PC hardware breakpoint registers.
- Asynchronous DRAMC is replaced with a DDR/SDR SDRAMC.
- Increase in the number of universal asynchronous receiver/transmitter (UART) modules from two to three.
- Two 16-bit timers are replaced with four 32-bit timers.
- Increase in the number of DMA channels from 2 to 16.
- Significant increase in the number of available GPIOs.
- Addition of several modules:
  - 10/100 fast Ethernet (media access) controller (FEC) for the MCF5208 device
  - Queued serial peripheral interface (QSPI)
  - Four periodic interrupt timers (PITs)

Table 1 provides a brief look at the design considerations and benefits of migrating from the MCF5206e to the MCF5208/7. It also provides page references to detailed descriptions of specific migration issues.

**Table 1. MCF5206e to MCF5208/7 Migration Issue Summary**

| Device Feature | Device Implementation | | | Impact | | Migration Difficulty | Page Number |
|---|---|---|---|---|---|---|---|
| | MCF5206e | MCF5207 | MCF5208 | Hardware | Code | | |
| CPU Core | V2 ColdFire | | | — | √ | Low | 16 |
| Instruction Set | ISA_A | ISA_A+ | | — | √ | Low | 16 |
| BDM Module | Rev. A | Rev. B+ | | — | √ | Low | 17 |
| Multiply Accumulate | MAC | EMAC | | — | √ | Medium | 18 |
| CPU Frequency | 54 MHz | 166.67 MHz | | √ | √ | Low | 10 |
| Performance (Dhrystone 2.1) | 50 MIPS | 159 MIPS | | — | √ | Low | 10 |
| System/Bus Frequency | 54 MHz | 83.33 MHz | | √ | √ | Low | 10 |
| Voltage | 3.3V | 3.3V I/O, 3.3/2.5/1.8V Mem, 1.5V Core | | √ | — | Medium | 4 |
| Power Requirement | 402.6 mW (typical run power at 54MHz) 287 mW (STOP power at 54MHz) | TBD | TBD | √ | — | Medium | 8 |
| Packaging | 160 QFP | 144 LQFP 144 MAPBGA | 160 QFP 196 MAPBGA | √ | — | Medium | 11 |
| SRAM Size | 8 KBytes | 16 KBytes | | — | √ | Low | 10 |
| Instruction/Data Cache Size | 4KBytes I-cache | 8 KBytes data, instruction, or split cache | | — | √ | Low | 10 |
| DRAMC | Asynchronous DRAMC | SDR/DDR Synchronous DRAMC | | √ | √ | High | 21 |
| Chip Selects | Up to 8 | Up to 6 | | √ | √ | Low | 23 |
| Ethernet | — | — | Yes | √ | √ | Low | 27 |
| UART | 2 | 3 | | √ | √ | Low | 27 |
| $I^2C$ | M-bus | $I^2C$ | | √ | √ | Low | 27 |
| QSPI | — | Yes | | √ | √ | Low | 27 |
| Timers | Two 16-bit Timers | Four 32-bit Timers | | — | √ | Low | 25 |
| WDT | Yes | Yes | | — | √ | Low | 26 |
| PITs | — | Two 16-bit PITs | | — | √ | Low | 28 |

**Table 1. MCF5206e to MCF5208/7 Migration Issue Summary (continued)**

| Device Feature | Device Implementation | | | Impact | | Migration Difficulty | Page Number |
|---|---|---|---|---|---|---|---|
| | MCF5206e | MCF5207 | MCF5208 | Hardware | Code | | |
| External Interrupts | Up to 3 discrete interrupts requests or up to 7 encoded interrupt priority levels | Up to 3 discrete interrupts | | √ | √ | Low | 30 |
| DMA | 2 channel | 16 channel | | √ | √ | Low | 21 |
| GPIOs | Up to 8 | Up to 30 | Up to 50 | √ | √ | Medium | 25 |
| JTAG | JTAG boundary scan | | | √ | √ | Low | 26 |

# 2 Device Differences

This section addresses differences between the MCF5206e and the MCF5208 and highlights what the user needs to consider when making this migration. The categories discussed in this section include electrical characteristics, performance, mechanical characteristics, chip configuration, external signals, core and architectural improvements, on-chip modules, and software considerations.

## 2.1 Electrical Characteristics

The following sections discuss differences in electrical characteristics between the two devices, including voltage conversion, driver strength/capacitive loading, and power consumption.

### 2.1.1 Voltage Conversion

One important consideration in migrating from the MCF5206e to the MCF5208/7 is the difference in voltage. The MCF5206e is a 3.3V only device, while the MCF5208 has split voltage rails. The core supply is always 1.5V, while the I/O is 3.3V. The memory bus can be 3.3V when interfacing to SDR SDRAM, 2.5V when interfacing to DDR SDRAM, or 1.8V when interfacing to mobile (low-power) DDR SDRAM. Table 2 shows some of the suggested combinations of supply voltages for the MCF5208.

**Table 2. MCF5208 Voltage Supply Summary**

| I/O $V_{DD}$ | SD_$V_{DD}$ | Core $V_{DD}$ | Description |
|---|---|---|---|
| 3.3V | 3.3V | 1.5V | This dual-supply configuration is recommended for use with SDR SDRAMs. |
| 3.3V | 2.5V | 1.5V | These are the recommended supply rails when using standard DDR SDRAM.<br>**Note:** The MCF5208 does not require a $V_{REF}$ input when using DDR SDRAM, but the memory will require a 1.25V $V_{REF}$ supply voltage. |
| 3.3V | 1.8V | 1.5V | This is the recommended voltage supply configuration for using low-power DDR SDRAM devices. |

The memory supply determines the operating voltage of the system bus for any access. The value used for SD_VDD in a system is based on the type of SDRAM used, but the decision will have an impact on any device that is connected to the system bus. The next sections discuss some of the design considerations for mixing devices with different operating voltages on the system bus.

#### 2.1.1.1 Peripheral Voltage Migration Concerns

Since peripheral devices are more likely to be available with only one option for the operating voltage, they present the largest concern from a migration standpoint. The MCF5208 I/Os are not 5V tolerant, so bus-interfaced external peripheral devices used in an existing MCF5206e system might not be compatible with the MCF5208 bus without proper system design. If any external peripherals that are only available as 5V devices are needed, then level shifting buffers (such as the SN74AVC8T245) will be required to isolate the peripheral's 5V I/Os from the MCF5208 to ensure that the processor and peripheral both see the correct logic levels.

Buffers are also required for 3.3V peripherals when 2.5V or 1.8V is used for the system's SD_VDD. In addition to protecting the pins from a voltage that exceeds their supply, level shifting buffers also help to ensure that the $V_{IH}$ and $V_{IL}$ requirements for both the ColdFire processor and the peripheral device are met. For example, if 1.8V is used for the SD_VDD supply, then a high output from the processor will not reach a voltage high enough to meet the minimum $V_{IH}$ requirements for most 3.3V devices. Level shifters can be used to correct the voltage level mismatch and prevent the device from erroneously detecting low voltage levels on all interface pins, while also protecting the bus interface pins on the MCF5208.

#### 2.1.1.2 Memory Voltage Migration Concerns

Many bulk memories can be found in 3.3V, 2.5V, and 1.8V varieties, so in most cases replacements for memory devices should not present a problem. However, in some cases the use of 3.3V memories might be a more cost effective solution even if SD_VDD is powered at 2.5V or 1.8V. Depending on the types, sizes, and number of memory devices used in the system, there could be cost benefits to using 3.3V memories and level shifting buffers. In particular, if level shifting buffers are already being used in order to accommodate the use of 3.3V peripheral devices, then the use of 3.3V devices for Flash, SRAM, and other non-SDRAM memories might help to decrease overall system costs.

#### 2.1.1.3 System Bus Design Block Diagram

Figure 1 shows the example configuration for a system using 3.3V SDR SDRAM.

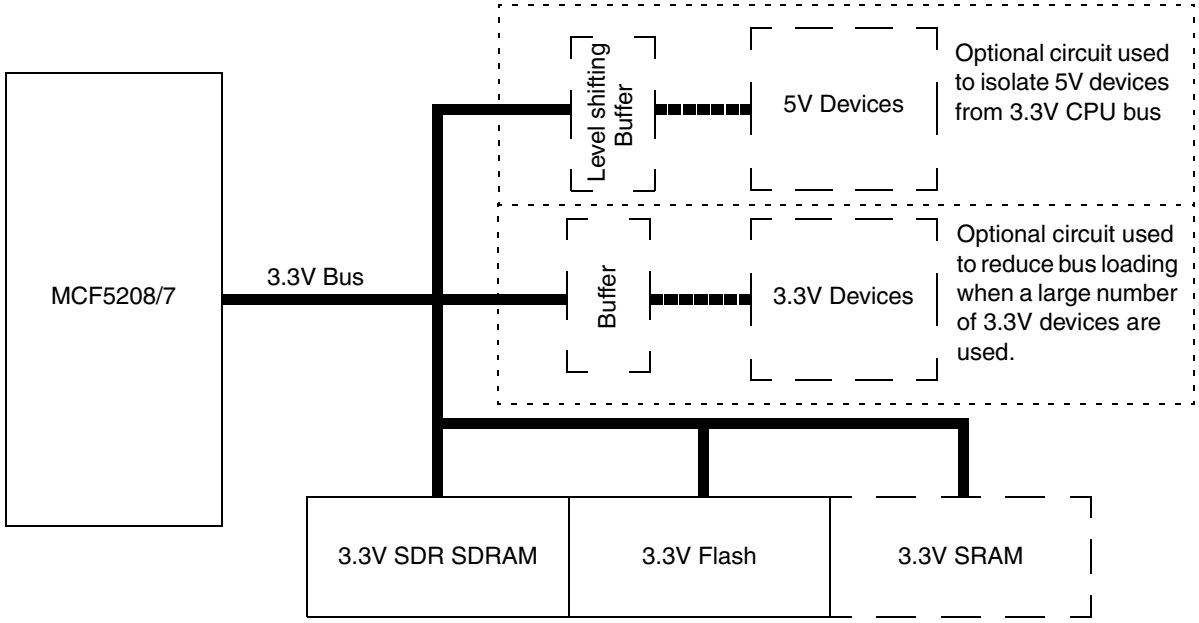


**Figure 1. Example System Bus Design for an SDR SDRAM System**

Since this is a 3.3V system, the Flash and SRAM are connected directly to the MCF5208's bus. However, care should be taken to avoid exceeding the specified 50pF capacitive load for the bus. If the number of devices connected to the 3.3V bus creates a load greater than 50pF, then an optional buffer can be added, as shown in the figure. This prevents additional devices from overloading the processor's bus. If any 5V devices are used in the system, then a level shifting buffer must be used to isolate the 5V device(s) from the MCF5208.

The same basic system bus architecture can be used for applications with 2.5V DDR SDRAM or 1.8V mobile DDR SDRAM. Figure 2 shows an example block diagram for a system using 2.5V DDR SDRAM.

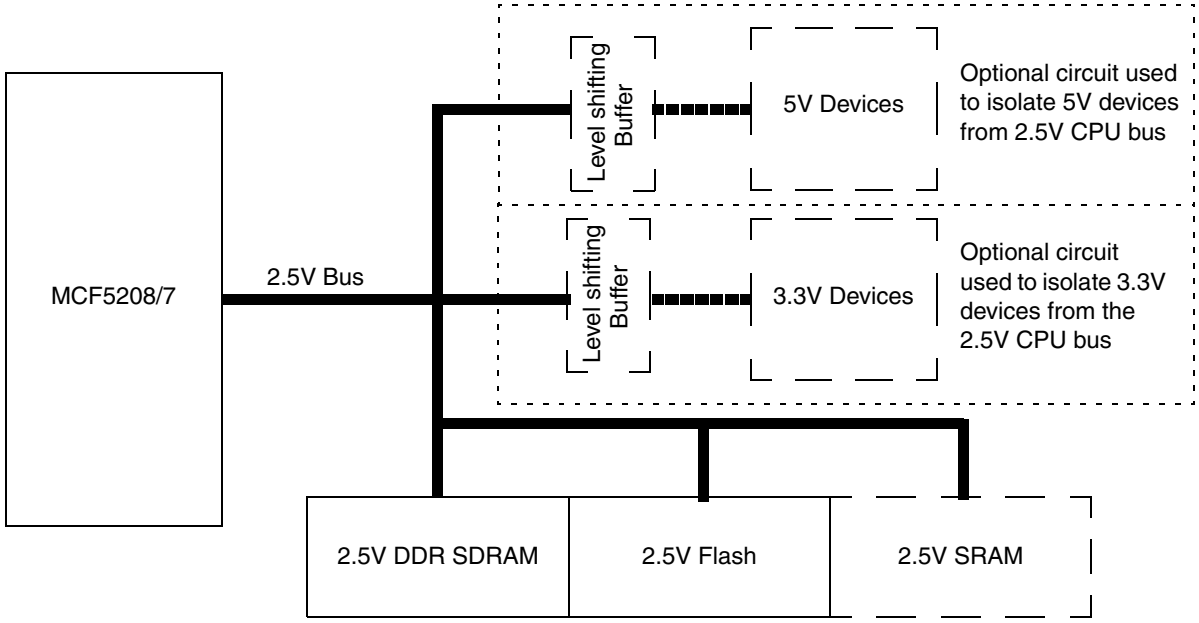


**Figure 2. Example System Bus Design for a DDR SDRAM System**

The block diagram is similar to what would be used for a 3.3V system bus, except 2.5V devices are used for the Flash and optional SRAM and all 3.3V devices are behind a level shifting buffer.

Figure 3 shows a second block diagram that could be used for 2.5V or 1.8V systems.

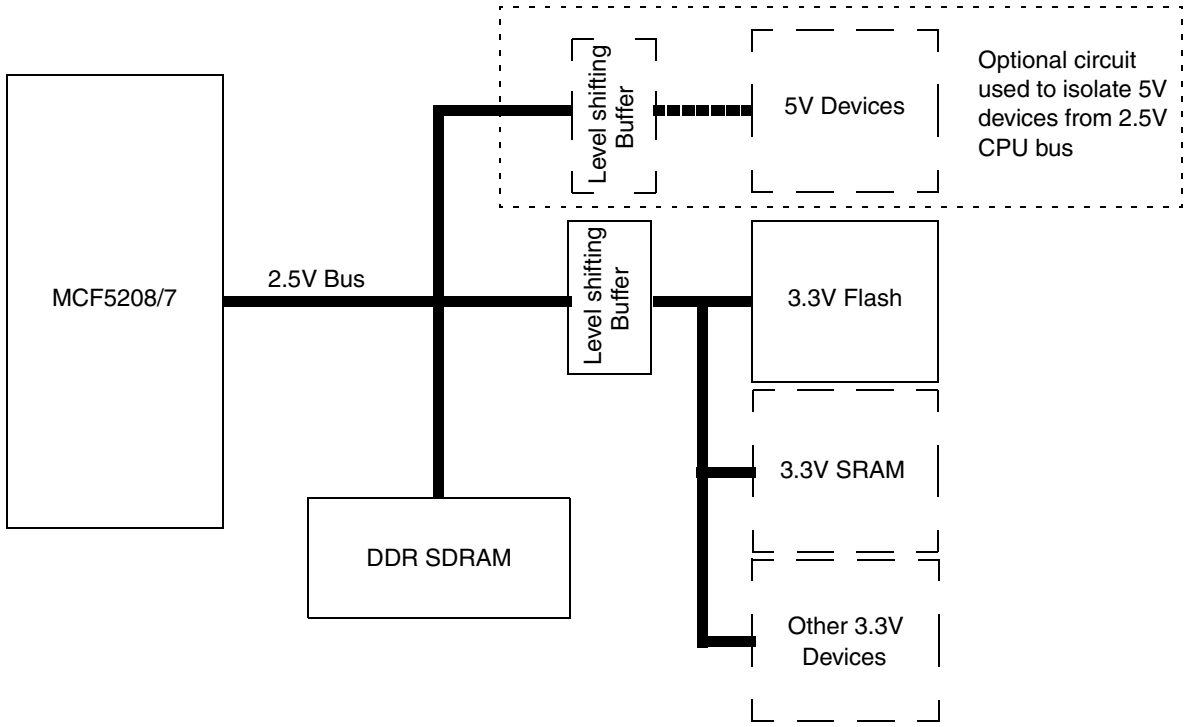


**Figure 3. Alternate Example System Bus Design for a DDR SDRAM System**

In this version of the system, the only device directly connected to the CPU bus is the DDR SDRAM. Since the boot Flash and other memories are behind a level shifting buffer, 3.3V devices can be used. One of the drawbacks to this approach is that additional wait states might be needed for accesses to the memories or peripherals on the other side of the level shifter to account for delays added by the buffer itself. The trade-offs between performance and overall costs should be evaluated to determine the ideal solution for a particular application.

## 2.1.2 Drive Strength

On the MCF5206e, the pads have one drive strength (50pF), but on the MCF5208 there are four programmable options for the drive strength—10pF, 20pF, 30pF, and 50pF. The processor can boot in either 10pF drive or 50pF drive; then the drive strength can be reprogrammed to one of the other modes during system initialization. The MCF5208 also allows for different drive strengths on functional groups of pins. For example, the bus signals which will typically be connected to multiple devices can be setup for 50pF drive, but the FEC pins could be programmed for 10pF drive.

## 2.1.3 Power Consumption

The MCF5208 incorporates several features designed to allow for greater control over the device's power consumption either when the processor is running or when it is stopped. The following sections describe some of these features in more detail.

### 2.1.3.1 Low Power Modes

The MCF5206e uses a STOP instruction to enter a low-power mode. Executing the STOP instruction would cause the ColdFire core to stop executing instructions until an interrupt of an appropriate level was received to wake-up the CPU. This reduces the power consumption of the device since the core is not executing; however, all of the clocks to the chip are still active and enabled therefore the STOP mode current is still high (almost 300mA when running at 54 MHz).

The MCF5208 improves upon the MCF5206e's STOP by adding a selection of low-power modes—WAIT, DOZE, and STOP. All three modes are still entered by executing a STOP instruction. The state of the LPCR[LPMD] field determines the mode entered by the STOP instruction. These modes allow for greater control over the chip's power consumption.

WAIT and DOZE mode on the MCF5208 provide similar operation to executing a STOP instruction on the MCF5206e. The CPU is idled, and the CPU and memory clocks are also shut off. This causes a significant decrease in power consumption with little or no delay required for the CPU to begin operation again once a wake-up is received.

On the MCF5208 STOP mode will idle the CPU and disable the CPU, memory, and system clocks. The LPCR[STPMD] field allows for programmable disabling of additional clocks in STOP mode to decrease power consumption even more. The bus clock, PLL, and oscillator can also be disabled. When all of the possible clocks are turned off, this yields the lowest possible power consumption (current draw in the microamps range). The trade-off for the lower power consumption is a longer wake-up time. In particular, if the PLL is disabled, when an interrupt is detected to wake-up the CPU, the PLL has to regain lock before the system can begin the wake-up process. Also, since the system clocks are disabled in STOP mode, only external interrupts can be used to wake-up the CPU without a reset (the modules are not clocked and therefore cannot issue interrupt requests).

### 2.1.3.2 Limp Mode

The MCF5208 also includes a limp mode, which allows the CPU to be clocked directly by a divided input clock and bypass the PLL completely. Although limp mode can be used for regular operation, some of the modules (SDRAMC and the FEC) cannot operate at the low frequencies used in limp mode; therefore, limp mode is primarily intended to be used as a low-power mode. Since the power consumption of the device is directly proportional to the clock speed, using a low speed clock when possible will reduce power consumption. The limp mode can also be used in combination with the low power modes to further reduce power consumption. For example, if the device is put into limp mode to reduce the system clock frequency and then placed into WAIT or DOZE mode, the power consumption will be less than if the full system clock frequency were used. Since the part is only in WAIT/DOZE mode, all of the peripherals (with the exception of the SDRAMC and FEC) can be active and are capable of issuing an interrupt to wake-up the CPU.

### 2.1.3.3 Module Clock Disables

The MCF5208 incorporates peripheral power management registers that allow for a simple means of enabling and disabling the clock for on-chip peripherals on a module-by-module basis. Disabling the clocks for any modules that are not used in the system can help to decrease current draw in both RUN mode and low-power modes. The module clocks can also be enabled/disabled on-the-fly, so that infrequently

used modules that have a large current draw can be disabled when they are not being used. For instance, if Ethernet is used periodically to download software/firmware updates directly to the device but is not used as part of the main function of the device, then the overall current consumption could be decreased by disabling the clock to the FEC module until the device needs to check for updates.

## 2.2 Performance

The following sections discuss some of the MCF5208 features that can directly help to increase the overall system performance.

### 2.2.1 Frequency

One of the primary advantages of migrating to the MCF5208 is the significant increase in performance and frequency. Whereas the MCF5206e has a Dhrystone 2.1 MIPS rating of approximately 50 MIPS when running at 54 MHz, the MCF5208 offers approximately 103 MIPS of performance with the same system frequency (108 MHz core frequency). That is, the MCF5208 offers over a 2x performance boost without changing the bus and system frequency. Moreover, the MCF5208 can provide 159 MIPS performance at its maximum frequency of 166 MHz.

### 2.2.2 System PLL

The MCF5208 uses an on-chip PLL to generate the system clocks for the chip. The PLL is designed to operate from a 16 MHz input from either a crystal using the on-chip oscillator circuit or an external oscillator.

The PLL feedback divider register (PFDR) is used to control the multiplication factor used by the PLL. The MCF5208's PLL supports multipliers from 7.33 to 11.25 times the input reference frequency for the core frequency. The system frequency used for modules and the bus is always half the core frequency.

### 2.2.3 On-chip SRAM

The MCF5208 has 8 KBytes of on-chip SRAM, whereas the MCF5206e only has 4 KBytes. The on-chip SRAM provides single cycle access memory to the CPU that is clocked at the CPU frequency. Use of the on-chip SRAM to store critical code, data structures, and/or stack space can yield a significant performance increase over storing code in external memory.

The SRAM on the MCF5208 is dual-ported. On the MCF5206e, the on-chip SRAM has one port, so only the CPU can access the memory. The dual-port on the MCF5208 allows for on-chip masters other than the CPU to read and write the SRAM. This means that the source and/or destination addresses for an eDMA transfer can point into SRAM space and the descriptors and/or buffers for the FEC can be stored in the SRAM.

### 2.2.4 Cache

In addition to providing more on-chip SRAM memory than the MCF5206e, the MCF5208 also has twice as much on-board cache memory. The cache also has more configuration options. The MCF5206e's cache

is an instruction cache only; however, the MCF5208 cache can be used as an instruction or data cache. There is also an option to configure the cache as a split data and instruction cache where one half the cache is dedicated to instructions and the other half to data.

## 2.3 Mechanical Characteristics

With the trend towards miniaturization, smaller, lighter, and higher performance products have paved the way for smaller component packages and higher pin counts. For this reason, the mold array process-ball grid array (MAPBGA) is used as the production package for the MCF5208 and MCF5207. The MCF5207 and MCF5208 are also available in quad flat pack (QFP) packages, but QFP versions of the device are slightly higher in cost. In addition, the MCF5208 has fewer pins on the QFP version of the device, so some functionality is lost when using the QFP. This creates a definite advantage for customers to migrate from the use of QFPs to BGAs.

The BGA is a surface mount package that uses solder balls arranged in a grid array instead of the lead pins normally used in quad flat pack (QFP) and other packages. The MCF5208's 196-pin and MCF5207's 144-pin MAPBGA packages give many advantages over a QFP. The most evident advantage of the MAPBGA versus the QFP is the savings in board real estate. Most BGAs are typically 20-25% smaller than their QFP counterparts. For the MAPBGA package, the entire surface of the package, rather than just the edges, can be used for interconnection. When the total board area required to place and route the package is taken into account, the MAPBGA can reduce size by as much as 50%. Figure 4 illustrates the differences between the QFP and the MAPBGA.

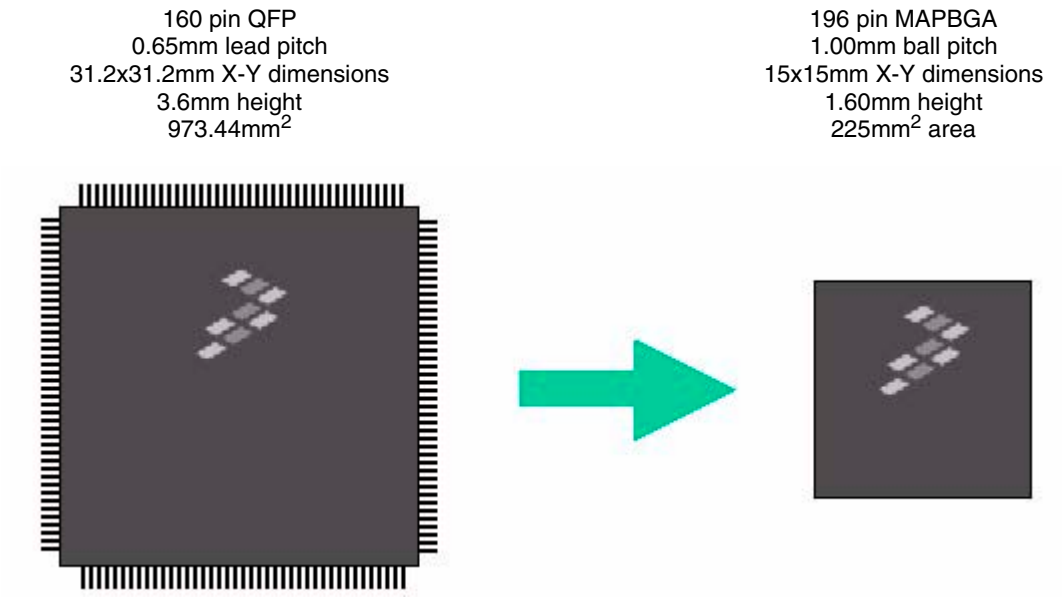


**Figure 4. Customer Board Area Reduction Example**

Lead pitch is a major consideration when working with high pin count QFPs. For these packages, the lead pitch can be as fine as 0.65 mm. The BGA package with the same number of pins is easier to mount, since the lead pitch is wider than a QFP package. BGAs are also less fragile and easier to handle both before and during assembly. The placement operation for this package is usually far easier and more reliable than for fine-pitch QFPs.

BGAs also have higher assembly yields. For example, BGAs typically have less than 5 parts per million joints (ppmj) compared to the 50–100 ppmj of QFP devices. BGAs have higher assembly yields than QFPs for the following reasons:

- No bent leads or coplanarity problems.
- Self-aligning on solder pads.
- Solder balls are always solderable (unlike plated leads).
- Easy solder paste printing.
- Large pitch: 1.0 mm, 1.5 mm, 1.27 mm.
- Large ball diameter sizes: MAPBGA 19.68 mils.

In general, BGAs have better electrical and thermal properties than their QFP counterparts. The long fingers of the QFP lead frame make it more inductive than shorter ones. Additional enhancements can be made to the BGA package by adding metal layers for power and ground.

## 2.4 External Signal Comparison

Because of the devices' different packages, it is not useful to compare their pinouts. It is useful, however, to discuss the commonality of signals brought out of each device. Table 3 is a summary of each device's externally-accessible signals.

**NOTE**

Some signal functions on the MCF5208 and MCF5207 are multiplexed on the same pin. Therefore, all signal functions are not available at the same time. Refer to the *MCF5208 Reference Manual* for more information on pin muxing.

**Table 3. MCF5206e and MCF5208/7 External Signal Comparison**

| Module | MCF5206e Signal | MCF5208/7 Signal | Comment |
|---|---|---|---|
| Reset | $\overline{\text{RSTI}}$ | RESET | |
| | $\overline{\text{RSTO}}$ | $\overline{\text{RSTOUT}}$ | |
| Clocks | CLK | EXTAL | Used as clock input for the MCF5208 when an external oscillator is used. |
| | — | XTAL | |
| | — | FB_CLK | FB_CLK is the system clock output from the MCF5208's PLL. |

**Table 3. MCF5206e and MCF5208/7 External Signal Comparison (continued)**

| Module | MCF5206e Signal | MCF5208/7 Signal | Comment |
|---|---|---|---|
| Mode Selects | — | $\overline{\text{RCON}}$ | Reset configuration can be overridden via D[9,7:1] |
| | | DRAMSEL | This signal is used to select between SDR and DDR operation.<br>DRAMSEL high = SDR mode<br>DRAMSEL low = DDR mode |
| | MTMOD | JTAG_EN | These signals are used to select between JTAG and BDM functionality; however, the pin states are reversed.<br>BDM = MTMOD high = JTAG_EN low.<br>JTAG = MTMOD low = JTAG_EN high. |
| External Memory Interface | A[27:0] | A[23:0] | A[27:24] are muxed with $\overline{\text{CS}}$[7:4] and $\overline{\text{WE}}$[3:0] on the MCF5206e<br>A[23:22] are muxed with $\overline{\text{FB\_CS}}$[5:4] on the MCF5208. |
| | D[31:16] | D[31:16]/SD_D[31:16] | When DDR is used with the MCF5208 the upper half of the data bus becomes a dedicated 16-bit wide port for the DDR. |
| | D[15:0] | D[15:0]/FB_D[31:16] | When DDR is used with the MCF5208 the lower half of the data bus becomes a dedicated 16-bit wide port for the FlexBus. |
| | $\overline{\text{WE}}$[3:0] | $\overline{\text{BE}}/\overline{\text{BWE}}$[3:0]/SD_DM[3:0] | |
| | R/$\overline{\text{W}}$ | R/$\overline{\text{W}}$ | |
| | | $\overline{\text{OE}}$ | |
| | SIZ[1:0] | — | |
| | TT[1:0] | — | |
| | ATM | — | |
| | $\overline{\text{TS}}$ | $\overline{\text{TS}}$ | |
| | $\overline{\text{TA}}$ | $\overline{\text{TA}}$ | |
| | $\overline{\text{ATA}}$ | — | |
| | $\overline{\text{TEA}}$ | — | On the MCF5208 an external $\overline{\text{IRQ}}$ signal along with $\overline{\text{TA}}$ could be used to duplicate functionality similar to $\overline{\text{TEA}}$. |
| Chip Selects | $\overline{\text{CS}}$[7:0] | $\overline{\text{FB\_CS}}$[5:0] | |
| Bus Arbitration | $\overline{\text{BR}}$ | — | The MCF5208 does not support external bus masters, so the $\overline{\text{BR}}$, $\overline{\text{BG}}$, and $\overline{\text{BD}}$ signals are not implemented. |
| | $\overline{\text{BG}}$ | — | |
| | $\overline{\text{BD}}$ | — | |

**Table 3. MCF5206e and MCF5208/7 External Signal Comparison (continued)**

| Module | MCF5206e Signal | MCF5208/7 Signal | Comment |
|---|---|---|---|
| SDRAM Controller | RAS[1:0] | SD_RAS | |
| | CAS[3:0] | SD_CAS | |
| | $\overline{\text{DRAMW}}$ | $\overline{\text{SD\_WE}}$ | |
| | — | SD_CS[1:0] | |
| | — | SD_CLK | |
| | — | $\overline{\text{SD\_CLK}}$ | |
| | — | SD_CKE | |
| | — | SD_A10 | |
| | — | SD_DQS[3:0] | |
| | — | SD_SDR_DQS | |
| External Interrupts | IRQ[7,4,1]/IPL[2:0] | IRQ[7,4,1] | The interrupt request pins on the MCF5206e can be used as either predefined interrupt request pins or as encoded interrupt priority level signals. |
| Fast Ethernet Controller | — | FEC_MDIO | The MCF5206e and MCF5207 do not include FEC functionality. Only the MCF5208 includes these signals. |
| | — | FEC_MDC | |
| | — | FEC_COL | |
| | — | FEC_CRS | |
| | — | FEC_RXCLK | |
| | — | FEC_RXDV | |
| | — | FEC_RXD[3:0] | |
| | — | FEC_RXER | |
| | — | FEC_TXCLK | |
| | — | FEC_TXEN | |
| | — | FEC_TXER | |
| | — | FEC_TXD[3:0] | |
| $I^2C$ | SDA | I2C_SDA | |
| | SCL | I2C_SCL | |
| QSPI | — | QSPI_CS[2:0] | The MCF5206e does not include QSPI functionality. |
| | — | QSPI_CLK | |
| | — | QSPI_DIN | |
| | — | QSPI_DOUT | |

**Table 3. MCF5206e and MCF5208/7 External Signal Comparison (continued)**

| Module | MCF5206e Signal | MCF5208/7 Signal | Comment |
|---|---|---|---|
| UARTs | RXD[2:1] | URXD[2:0] | |
| | TXD[2:1] | UTXD[2:0] | |
| | $\overline{\text{CTS}}$[2:1] | $\overline{\text{UCTS}}$[2:0] | |
| | $\overline{\text{RTS}}$[2:1] | $\overline{\text{URTS}}$[2:0] | |
| Timers | TIN[2:1] | DTIN[3:0] | |
| | TOUT[2:1] | DTOUT[3:0] | |
| Debug Port and JTAG | DSCLK/$\overline{\text{TRST}}$ | DSCLK/$\overline{\text{TRST}}$ | |
| | TCK | PSTCLK/TCK | The MCF5206e uses the clock input as PSTCLK instead of having a dedicated signal. |
| | $\overline{\text{BKPT}}$/TMS | $\overline{\text{BKPT}}$/TMS | |
| | DSI/TDI | DSI/TDI | |
| | DSO/TDO | DSO/TDO | |
| | DDATA[3:0] | DDATA[3:0] | |
| | PST[3:0] | PST[3:0] | |
| | — | ALLPST | ALLPST is the logical AND of the four PST signals. This signal is used on QFP devices to determine if the CPU is halted or running without having access to full real-time trace information (PST[3:0] and DDATA[3:0]) |
| DMA | DREQ[1:0] | $\overline{\text{DREQ0}}$ | |
| | — | $\overline{\text{DACK0}}$ | |
| Test | — | TEST | TEST should be connected to VSS for normal operation. |
| | — | PLL_TEST | This signal is used for factory testing and should be treated as a no connect (N.C.) |
| | $\overline{\text{HIZ}}$ | — | |
| Power Supplies | VDD | EVDD | |
| | VSS | VSS | |
| | — | IVDD | 1.5V core supply |
| | — | SD_VDD | 3.3V/2.5V/1.8V SDRAM supply |
| | | PLL_VDD | Filtered PLL supply |
| | — | PLL_VSS | Ground for PLL filter circuit |

## 2.5 ColdFire V2 Enhancements

The MCF5206e and MCF5208 are both based on the ColdFire Version 2 core; however, a number of enhancements have been made to the core, BDM, and MAC module. Table 4 gives a summary of the enhancements that have been made to the ColdFire V2 since the MCF5206e. These differences are described in detail in the following sections.

**Table 4. V2 Enhancements Summary**

| Feature | | MCF5206e | MCF5208 |
|---|---|---|---|
| Core | Stack pointer | Single stack pointer | User and supervisor stack pointer |
| | Instruction Set | ISA_A | ISA_A+ |
| BDM | Hardware breakpoints | One PC breakpoint register | Four PC breakpoint registers |
| Multiply Accumulate Unit | — | MAC | Enhanced MAC (EMAC) |

### 2.5.1 V2 Core Enhancements

This section discusses the enhancements to the ColdFire V2 core present in the MCF5208 but not found in the MCF5206e.

#### 2.5.1.1 User and Supervisor Stack Pointers

The MCF5206e supports a single stack pointer that is used for both user and supervisor mode operations. However, the V2 core used for the MCF5208 has been enhanced to support two independent stack pointer (A7) registers—the supervisor stack pointer (SSP) and the user stack pointer (USP).

The dual stack pointer functionality is enabled by setting the enable user stack pointer bit in the cache control register, CACR[EUSP]. If this bit is cleared, then the MCF5208 will function like the MCF5206e with just one stack pointer (A7). CACR[EUSP] is cleared at reset, so by default the stack pointer functionality is the same for the MCF5206e and MCF5208.

#### 2.5.1.2 Additions to the Instruction Set Architecture

The original ColdFire instruction set architecture (ISA) was derived from the M68000-family opcodes based on extensive analysis of embedded application code. After the initial ColdFire compilers were created, developers identified ISA additions that would enhance both code density and overall performance. Additionally, as users implemented ColdFire-based designs into a wide range of embedded systems, they identified frequently used instruction sequences that could be improved by the creation of new instructions. This observation was especially prevalent in development environments that made use of substantial amounts of assembly language code.

The MCF5208 supports ISA_A+ which includes all of the functionality of the original ISA_A implemented on the MCF5206e with the addition of several new instructions. Since no functionality is removed, the instruction set architecture enhancements should not pose an issue for migration.

Table 5 summarizes the new instructions added to Revision A+ ISA. For more details, refer the *ColdFire Programmer's Reference Manual*.

**Table 5. ISA Revision A+ New Instructions**

| Instruction | Description |
|---|---|
| BITREV | The contents of the destination data register are bit-reversed; that is, new Dx[31] = old Dx[0], new Dx[30] = old Dx[1],..., new Dx[0] = old Dx[31]. |
| BYTEREV | The contents of the destination data register are byte-reversed; that is, new Dx[31:24] = old Dx[7:0],..., new Dx[7:0] = old Dx[31:24]. |
| FF1 | The data register, Dx, is scanned, beginning from the most-significant bit (Dx[31]) and ending with the least-significant bit (Dx[0]), searching for the first set bit. The data register is then loaded with the offset count from bit 31 where the first set bit appears. |
| STLDSR | Pushes the contents of the status register onto the stack and then reloads the status register with the immediate data value. |

## 2.5.2 BDM Revision B+ Enhancements

As with the core, enhancements have been made to the debug module based on feedback from customers and third-party developers. The MCF5206e implemented the original version of the ColdFire debug module (revision A); however, the MCF5208 implements the new revision B+. Table 6 summarizes the differences between the revision A debug module and revision B+. Since revision B+ encompasses all of the original functionality of revision A, it is backwards compatible and should pose little or no concern when migrating an existing design to the MCF5208. In fact, the new functionality can make debugging easier.

**Table 6. Debug Module Revision B+ Enhancements**

| Feature | MCF5206e | MCF5208 | Comments |
|---|---|---|---|
| Debug module | Rev. A | Rev. B+ | |
| Breakpoint functionality | BDM and hardware breakpoints use common resources, so some BDM commands will override hardware breakpoint settings. | Separate registers used for BDM commands and hardware breakpoints. No restrictions concerning the interaction between BDM commands and hardware breakpoints. | |

**Table 6. Debug Module Revision B+ Enhancements (continued)**

| Feature | MCF5206e | MCF5208 | Comments |
|---|---|---|---|
| PC breakpoints | One PC breakpoint register with a mask | • One PC breakpoint register with a mask<br>• Three additional PC breakpoint registers with no masking | |
| $\overline{\text{BKPT}}$ | Halts the processor | Halts the processor or optionally generates debug interrupt | |
| BDM header | 26-pin BDM header | 26-pin BDM header | |
| Real-time trace | PST[3:0]/DDATA[3:0] | PST[3:0]/DDATA[3:0] on BGA packages<br>ALLPST on QFP devices | ALLPST is generating by ANDing the PST[3:0] lines. |

The new debug module implementation has added hardware registers so that there are no restrictions concerning the interaction between BDM commands and the use of the hardware breakpoint logic. In some cases, the additional hardware is not program-visible. In other cases, there have been extensions to the debug module programming model.

The hardware register containing the BDM memory address is not a program-visible resource. Rather, it is a hardware register loaded automatically during the execution of a BDM command. In the Rev. B design, the execution of a BDM command does not affect the hardware breakpoint logic unless those registers are specifically accessed.

Other registers added to the debug module programming model are the BDM address attribute register (BAAR), and three new PC breakpoint registers (PCBR1–3). The BAAR is mapped to a DRc[3:0] address of 0x05. This 8-bit register is equivalent in the format of the low-order byte of the AATR register (See the *MCF5208 Reference Manual* for more information). This register specifies the memory space attributes associated with all BDM memory-referencing commands. The three new PC breakpoint registers are mapped to DRc[3:0] addresses 0x18, 0x1A, and 0x1B. These new registers provide additional PC breakpoint locations. However, there are no masking registers associated with them.

In order to reduce the pin count and allow for QFP packages, a new signal (ALLPST) has been added. Typically BDM cables that do not support real-time trace will AND the PST[3:0] signals together to determine if the CPU is currently running (ANDing PST[3:0] = 0) or halted (ANDing PST[3:0] = 1). The ALLPST signal implements the ANDing logic for the PST[3:0] signals on chip. Therefore a debugger can determine the running or halted state of the CPU with one signal instead of four. The ALLPST signal should be connected to the four pins on the BDM header that would normally be connected to PST[3:0].

## 2.5.3 MAC vs. EMAC

The ColdFire family supports two MAC implementations with different performance levels and capabilities. The original MAC uses a three-stage execution pipeline optimized for 16-bit operands and featuring a 16x16 multiply array with a single 32-bit accumulator. This is the MAC implemented on the MCF5206e. The EMAC used on the MCF5208 features a four-stage pipeline optimized for 32-bit operands, with a fully pipelined 32x32 multiply array and four 48-bit accumulators.

**Table 7. MAC vs. EMAC Comparison**

| Feature | MCF5206e MAC | MCF5208 EMAC | Comments |
|---|---|---|---|
| Pipeline | Three stage execution pipeline | Four stage execution pipeline | |
| Operand sizes | Optimized for 16-bit operands | Optimized for 32-bit operands | |
| Multiply array | 16x16 multiply array | 32x32 multiply array | |
| Accumulators | Single 32-bit accumulator | Four 48-bit accumulators | |
| Maximum result size | 32-bit products | 40-bit products | |
| Data types supported | Signed and unsigned integers | Signed integers, Unsigned integers, and Signed fixed-point fractions | |
| Instruction Set | MAC ISA | EMAC ISA | The EMAC has an extended ISA to support accesses to the added registers. |

# 2.6 Chip Architecture Enhancements

## 2.6.1 MCF5208 Split Bus Architecture

One of the major system changes on the MCF5208 is the use of a split bus architecture. The MCF5208 is capable of supporting both single data rate (SDR) and double data rate (DDR) SDRAMs; however, this flexibility in memory selection is not without challenges. The timing and signal routing requirements for a design incorporating DDR make the use of a shared data bus used for both SDRAM accesses and standard external access (Flash, SRAM, peripherals, etc.) problematic.

In order to solve this problem, the MCF5208 implements a programmable split bus architecture. This new bus architecture simplifies system design using DDR memory while keeping pin count and cost low. The data bus can be configured as a shared 32-bit SDRAM and FlexBus bus (similar to the way the MCF5206e operates) or as two separate 16-bit buses where the upper 16-bits of the bus are used exclusively for SDRAM controller cycles and the lower 16-bits of the data bus are dedicated to the FlexBus.

The state of the DRAMSEL pins is sampled at reset to select between the two modes. The 32-bit bus mode should be used for systems using SDR SDRAM or no SDRAM. If DDR is used, then the split 16-bit bus option should be selected. The operating mode selected at reset will determine the available port sizes for FlexBus accesses. If the 32-bit bus mode is selected, then the FlexBus will support 32-, 16-, and 8-bit port sizes. If the split 16-bit bus mode is used, then the FlexBus supports 16- and 8-bit port sizes. Programming a chip select control register (CSCR*n*) for a 32-bit port size while in split bus mode will result in undefined behavior. Using the split 16-bit bus mode allows for the upper half of the data bus to be dedicated to SDRAMC accesses. The routing and timing on the data signals is critical for DDR system design. Since the data lines aren't shared between DDR and memory or peripherals on the FlexBus, the data lines for the DDR can be routed point-to-point. This makes it much easier to keep trace lengths short, match the trace lengths, and minimize bus loading.

### 2.6.2 Architecture Changes

In addition to the split bus, the MCF5208 incorporates some architectural changes that help to improve overall performance.

First, the MCF5208 includes a cross-bar switch to handle bus arbitration between on-chip bus masters. The cross-bar switch connects the bus masters and bus slaves allowing bus masters to simultaneously access different bus slaves with no interference. The cross-bar switch is also responsible for providing arbitration among the bus masters when they access the same slave. A variety of bus arbitrations methods and attributes may be programmed on a slave by slave basis.

The MCF5208 has also made some changes to the memory map. On the MCF5206e memory locations for the internal memories, module register space, and external memories are all fully programmable. The MCF5208 still allows for programmable base addresses for internal and external memories, but the allowable addresses for different types of memory are restricted to certain address ranges. Table 8 shows the allowable address ranges for different memory types on the MCF5208. The restricted address ranges allow the cross-bar switch to make efficient decisions about bus cycle routing to the slave ports.

**Table 8. MCF5208 System Memory Map**

| Address Range | Destination Slave | Memory Region Size |
|---|---|---|
| 0x0000_0000–0x3FFF_FFFF | FlexBus | 1024 MBytes |
| 0x4000_0000–0x7FFF_FFFF | SDRAMC | 1024 MBytes |
| 0x8000_0000–0x8FFF_FFFF | On-chip SRAM | 256 MBytes |
| 0x9000_0000–0xBFFF_FFFF | Reserved | 768 MBytes |
| 0xC000_0000–0xDFFF_FFFF | FlexBus[1] | 512 MBytes |
| 0xE000_0000–0xEFFF_FFFF | Reserved | 256 MBytes |
| 0xF000_0000–0xFFFF_FFFF | Internal Peripheral System (IPS) space | 256 MBytes |

NOTES:
1 The second FlexBus region from 0xC000_0000–0xDFFF_FFFF can be used for non-cacheable, non-memory devices. The region from 0x0000_0000–0x3FFF_FFFF should be used for external memory that might be cached.

The internal peripheral system (IPS) space is where the on-chip module registers reside. This is similar to the MBAR space used on the MCF5206e. However, on the MCF5208 the base address for the module register space is fixed at 0xFC000_0000, whereas on the MCF5206e it is programmable to any 1 KByte boundary.

## 2.7 On-chip Module Differences

This section examines the MCF5208 on-chip modules and highlights differences between these modules and those found on the MCF5206e. The discussion starts with modules that provide basic functionality similar a module found on the MCF5206e, but where the programming model and features for the new module are significantly different. Next, modules with similar functionality and programming models are compared. Then modules that are the same between both devices are described. Finally, a brief description is given of the functionality of new modules found on the MCF5208 but not on the MCF5206e.

## 2.7.1 Different Modules

There are a number of modules on the MCF5208 that provide similar functionality to a module found on the MCF5206e; however, the implementation of the modules is significantly different.

### 2.7.1.1 DDR SDRAMC vs. Asynchronous DRAMC

One of the main advantages in migrating an MCF5206e design to the MCF5208 is the synchronous DRAM controller (SDRAMC). At the time of the MCF5206e's introduction, asynchronous DRAM (ADRAM) was the most commonly used bulk memory for embedded systems and in common usage for the PC market. However, the PC market no longer uses ADRAMs; as a result, the prices for ADRAMs have increased while availability has decreased. Migration to a system utilizing SDRAM in place of ADRAM allows for a reduction in memory costs for most systems.

The MCF5208 incorporates an SDRAMC that allows for flexibility in system memory solutions. The SDRAMC supports both single data rate (SDR) SDRAM and double data rate (DDR) SDRAM. Currently, SDR SDRAM is commonly used for embedded devices, but the PC market has shifted to DDR. By interfacing to both types of memory, the SDRAMC allows for the design of a low-cost system using SDR SDRAM with the possibility of migrating to DDR as it becomes a more cost-effective solution. In addition, SDRAM manufacturers are now offering mobile (low-power) DDR. The basic operation of mobile DDR is very similar to standard DDR, except the signalling is different. The MCF5208 design took the signalling and voltage levels for all three types of SDRAM (SDR, DDR, and mobile DDR) into account, and was specifically designed to support all three memory types.

The flexibility of the SDRAMC to support a wide range of SDRAM devices allows for a system designed with SDR (which is more commonly used in embedded system design now) the ability to migrate to DDR or even mobile DDR at a later time as the SDRAM market shifts.

### 2.7.1.2 eDMA vs. DMA

The MCF5208 uses a new, enhanced DMA module that offers a significant increase in functionality over the DMA module used on the MCF5206e. Table 9 gives a comparison of features supported by the MCF5206e DMA and the eDMA.

**Table 9. eDMA vs. DMA Comparison**

| Feature | MCF5206e DMA | MCF5208 eDMA | Comments |
|---|---|---|---|
| Channels | 2 channels | 16 channels | |
| Transfer sizes | 8, 16, 32, or 128-bit | 8, 16, 32, or 128-bit | |
| Transfer types | Single and dual address | Dual Address | |
| External requests | Up to 2 external request pins | Up to 1 external request and 1 external acknowledge pins | |
| Internal requests | — | UARTs and DMA timers can request DMA transfers | |
| Cycle steal mode | √ | Yes, can be supported using minor and major counters. | See paragraphs below for more information. |

**Table 9. eDMA vs. DMA Comparison (continued)**

| Feature | MCF5206e DMA | MCF5208 eDMA | Comments |
|---|---|---|---|
| Auto-align | √ | — | The eDMA does not support auto-alignment. The source address, destination address, and minor byte count must be evenly divisible by the source and/or destination size. |
| Channel arbitration | Fixed priority, where bandwidth control setting can be used to override the normal priority scheme. | Programmable fixed or round robin priority. In fixed mode the priority for each channel is programmable. | |
| Byte Count | 16-bit byte count | • 32-bit minor byte count<br>• 14-bit major loop count | |
| Bandwidth Control | √ | √ | For the MCF5206e the DMA can be programmed to negate its bus request for one cycle on specific byte count boundaries. The eDMA can be programmed to stall for 4 or 8 cycles after each r/w. |
| Source and Destination Address incrementing | Programmable as on or off. If increment is enabled then the source and/or destination is always incremented by the transfer size. | Programmable signed 32-bit increment or decrement. Value must be a multiple of the transfer size. | In addition to allowing for programmable increment values, the eDMA also has two 32-bit last address adjustment registers used as a final increment/decrement values for the source and destination at the completion of the major loop. This value can be used to restore the address to the initial value or adjust the address to point to the next data structure. |
| Address modulus | — | Supports source and destination address modulo to freeze upper address bits. | This feature is useful for applications that use circular queues for the source and/or destination. |
| Scatter/gather | — | √ | In scatter/gather mode, one of the entries in the eDMA's transfer control descriptor (TCD) is used to point to the next TCD for that channel. When the current transfer completes, the next TCD will automatically be loaded into the channel's registers. |
| Channel linking | — | • Link to channel after minor loop complete<br>• Link to channel after major loop complete | This feature allows the completion of one DMA channel (minor or major loop) to set the DMA request for a second channel. |
| Interrupts | Programmable interrupt at completion of transfer | Programmable interrupt at:<br>• completion of entire transfer (major count decrements to zero)<br>• and/or at the halfway point (when the major count decrements to half it's original value) | |

One of the major changes between the eDMA and the DMA has to do with the programming model. The eDMA uses 32 byte transfer control descriptors (TCDs) to program the attributes of each DMA transfer including the source and destination addresses, byte count, and configuration. The TCDs can be written directly to the control registers for a given channel (similar to programming the channel registers on the MCF5206e) or the TCDs can be stored in memory as a linked list. When the TCDs are stored in memory, the scatter/gather function can be used to program the DMA to automatically load a new TCD from memory when the current operation completes. Alternatively, a second DMA channel could also be used to load TCDs for one or more DMA channels.

In addition, the eDMA uses two nested loops for transfers. The minor byte count (defined by the NBYTES register) is the byte count used for the inner loop, and the major count (defined by BITER and CITER) determines how many times the inner loop executes. A DMA request is required to start each execution of the inner loop. This feature can be used to duplicate the cycle steal functionality of the MCF5206e's DMA. For example, if the cycle steal function was used to move 64 bytes where the source and destination transfer sizes were both longword, the same functionality could be duplicated on the eDMA. The NBYTES field is programmed to 4 so that one longword is transferred for each DMA request. Then the BITER and CITER fields are set to 16 in order to transfer a total of 64 bytes.

Some of the major differences between the DMA and eDMA have been discussed, but a full description of the functionality and new features of the eDMA is beyond the scope of this document. Please refer to the *MCF5208 Reference Manual* for more details on eDMA operation.

## 2.7.2 Similar Modules

The MCF5208 and MCF5206e share several modules that provide similar functionality. These modules are discussed in the following sections.

### 2.7.2.1 FlexBus vs. External Bus Interface

The MCF5208 uses the FlexBus module for controlling chip selects, timing, and attributes for non-DRAM external bus accesses. The operation of the FlexBus is very similar to that of the bus interface of the MCF5206e. However, additional functionality has been added and some timing characteristics are different. Table 10 shows a summary of the major features of both interfaces with comments giving a comparison or contrast.

**Table 10. Bus Interface Unit vs. FlexBus Comparison**

| Feature | MCF5206e | MCF5208 |
|---|---|---|
| **Split Bus Architecture** | | |
| Port Sizes/Data bus width | 8, 16, and 32-bit port sizes supported | • 8, 16, and 32-bit supported when using SDR or no DRAM<br>• 8 and 16-bit port supported when DDR is used |
| **Pin Count Reduction** | | |
| Address bus | • Up to 28 bits of address<br>• Allows for blocks up to 512 MBytes in size | • Up to 24 bits of address<br>• Allows for blocks up to 32 MBytes in size |

**Table 10. Bus Interface Unit vs. FlexBus Comparison (continued)**

| Feature | MCF5206e | MCF5208 |
|---|---|---|
| Chip Selects | Up to 8 programmable chip selects | Up to 6 programmable chip selects |
| **Access Controls** | | |
| Write Protection | No write protection controlled by the chip select | Programmable write protection on a per chip select basis |
| Access mask | Programmable access masks for CPU space/IACK, supervisor/user, and data/code cycles | No programmable access masks |
| **Programmable Bus Timing Parameters** | | |
| Wait States | Up to 15 wait states | • Up to 63 wait states<br>• Secondary wait state counter |
| Address Setup | 0 or 1 clocks of address setup | 0–3 clocks of address setup |
| Read Address Hold | 0 or 1 clocks of read address hold | 1–4 clocks of read address hold |
| Write Address Hold | 0 or 1 clocks of write address hold | 1–4 clocks of write address hold |
| Auto-acknowledge | Internal transfer acknowledge can be asserted to terminate bus cycles after the expiration of the wait state counter | Internal transfer acknowledge can be asserted to terminate bus cycles after the expiration of the wait state counter |
| Burst control | Programmable burst enable on a per chip select basis | Programmable burst enable on a per chip select and direction basis |

Since all MCF5206e and MCF5208 designs will require the use of the external bus, the differences between the two implementations are a migration concern for all designs. The changes between the bus interfaces are described in more detail in the following sections.

#### 2.7.2.1.1 Split Bus Architecture's Impact on the FlexBus

The MCF5208 implements a split bus architecture. The data port for the FlexBus can be either 32-bits wide (as on the MCF5206e) or 16-bits wide depending on the SDRAM mode selected. See Section 2.6.1, "MCF5208 Split Bus Architecture" for more information.

#### 2.7.2.1.2 Pin Count Reduction

The amount of on-chip integration for the MCF5208 required the removal of some bus signals in order to keep pin count and package costs down. The number of externally available address lines has been reduced as well as the number of chip select signals. For systems requiring access to a block of memory larger than 32 MBytes, two or more chip selects and a PLD or some discrete logic could be used to increase the max block size. If a system needs more chip selects for accessing peripherals or smaller blocks of memory, a PLD or demux circuit could be used to generate additional chip selects based on the upper address bits. In addition to the address and chip selects, some of the bus control signals have been eliminated.

**NOTE**

The 32-MByte per chip select memory size only applies to memory connected to the FlexBus. The SDRAM controller can support much larger blocks of memory (up to 256 MBytes per chip select).

#### 2.7.2.1.3 Access Controls

The MCF5208 does not implement the access control masks used by the MCF5206e to allow for discrete masking of supervisor/user, data/code, and CPU/interrupt acknowledge accesses on a per chip select basis. However, the MCF5208 does include a new access control feature allowing for programmable write protection of address spaces on a per chip select basis. When the write protect access mask bit is set (CSCR[WP]), a write access to a write-protected chip select region will generate an access error.

#### 2.7.2.1.4 Programmable Bus Timing Parameters

The function of the primary wait state counter is the same, but the MCF5208 allows for a wider range of wait states as well as more control over wait states for burst accesses. First of all, the MCF5208 allows for up to 63 wait states, whereas the MCF5206e only supports up to 15 wait states. In addition, a secondary wait state counter has been added. The secondary wait state counter allows for a different wait state counter to be used for burst accesses. If secondary wait states are enabled, the primary wait state counter terminates the first beat of a burst and the secondary wait state counter is used to terminate the following beats. This allows for much greater flexibility and efficiency for burst operations.

The MCF5208 also allows for a greater range of options for address setup and hold times. On the MCF5206e, the address setup, write address hold, and read address hold are binary features that are either enabled or disabled. The MCF5208 implements these features as multi-bit fields in the chip select control register allowing for a wider range of programmable bus timing options.

The user also gains greater control over burst operations by migrating to the MCF5208. The MCF5206e allows for enabling or disabling burst accesses for each chip select; however, the MCF5208's FlexBus has two independently programmable burst enable bits for each chip select—one for read cycles and one for write cycles.

### 2.7.2.2 DMA Timers vs. Timers

The basic functionality of the timer module on the MCF5206e and the DMA timers on the MCF5208 are the same. However, the DMA timers include some enhancements: the timers have been expanded to 32 bits instead of 16, and the timers can now be programmed to request DMA transfers in addition to, or instead of interrupt requests. This is useful for triggering DMA transfers on a periodic basis or on a capture edge.

### 2.7.2.3 GPIOs vs. Parallel Port

The biggest difference between the parallel port used to provide GPIO functionality on the MCF5206e and the GPIO on the MCF5208 is the number of pins that support GPIO functions. On the MCF5206e, only eight pins have optional GPIO capability; however, on the MCF5208, almost all of the module pins can optionally be used as GPIOs. On the MCF5208, 50 pins have multiplexed GPIO capability, and on the

MCF5207, 30 pins can be used as GPIOs. The large number of GPIO pins allows for flexibility in deciding which pin functions are used. The large number of programmable options makes the trade-offs between using a pin's primary function or GPIO less of a factor.

### 2.7.2.4 Watchdog Timers

Both devices have watchdog timer capability, but the modules used to provide these functions are different. In fact, the MCF5208 incorporates two different watchdog timers with different features and capabilities. Table 11 compares the main features of all three watchdog timers. The user has the option to use either, both, or none of the watchdog timers on the MCF5208.

**Table 11. MCF5206e Watchdog, MCF5208 Core Watchdog, and MCF5208 Watchdog Comparison**

| Feature | MCF5206e Watchdog | MCF5208 Core Watchdog | MCF5208 Watchdog | Comments |
|---|---|---|---|---|
| Programmable enable/disable | Yes | Yes | Yes (enabled by default) | |
| Reset/Interrupt Selection | Yes | Yes | No | |
| Timeout range | 8 timeout options ranging from $2^9$ to $2^{24}$ system clocks | $2^8$ to $2^{31}$ system clocks | 1 to $2^{16}$ system clocks | |
| Counts while processor halted | Yes | Programmable | Programmable | |
| Programmable low power operation | No | No | WAIT and DOZE mode | The MCF5208 watchdog has programmable independent options to be enabled or disabled for WAIT and DOZE low power modes. |
| Control register locking | Yes. The SYPCR can only be written once after system reset. | Yes. Setting the RO bit in the core watchdog control register (CWCR) makes the contents of the register read only until the next system reset. | No. The watchdog control register (WCR) is read/write. | |

### 2.7.2.5 JTAG

Like the MCF5206e, the MCF5208 has a JTAG module that allows for boundary scan testing. The functionality of the JTAG modules is the same, but the scan chain is different to account for the differences in signals, pins, and packages.

## 2.7.3 Reused Modules

Some of the modules found on the MCF5208 are reused from the MCF5206e design. These modules should pose little or no concern during migration.

#### 2.7.3.1 I²C and M-Bus

Despite the difference in names, the I²C module found on the MCF5208 is the same as the M-Bus module used on the MCF5206e. The functionality and programming model are identical, so low-level drivers written for the MCF5206e can easily be reused for the MCF5208. The base address for accessing the modules is different, and the value used for the frequency divider register (MFDR or I2FDR) may need to be adjusted to account for any change in the system frequency. Otherwise, few (if any) changes should be required.

#### 2.7.3.2 Universal Asynchronous/Synchronous Receiver/Transmitter (UART)

The UART module used for the MCF5208 is the same as the UART module on the MCF5206e. As with the I²C, low-level drivers should be fairly simple to port. Again, the base address for accessing the module registers should be updated, and the baud rate divider register (UBG1 and UBG2) values may need to be adjusted to maintain the same baud rate if the system frequency has changed.

### 2.7.4 New/Additional Modules

There are three modules on the MCF5208 (two on the MCF5207) that have no functional equivalent on the MCF5206e. They are the Fast Ethernet controller (FEC; not included on the MCF5207), queued serial peripheral interface (QSPI), and programmable interrupt timers (PITs). The following sections give a high-level description of these modules. The only migration issue with these modules is whether or not they could be of use in an existing design when porting to the MCF5208. If an existing MCF5206e design has similar functionality implemented off-chip, then it may be beneficial to eliminate the external components and bring the functionality on chip.

#### 2.7.4.1 Fast Ethernet Controller (FEC)

The MCF5208's integrated Fast Ethernet controller (FEC) performs the full set of IEEE® 802.3/Ethernet CSMA/CD media access control and channel interface functions. The FEC supports connection and functionality for the 10/100 Mbps 802.3 media independent interface (MII). It requires an external transceiver (PHY) to complete the interface to the media.

**NOTE**

The FEC is only available on the MCF5208. The MCF5207 does not include this functionality.

#### 2.7.4.2 Queued Serial Peripheral Interface (QSPI)

The queued serial peripheral interface module provides a high-speed synchronous serial peripheral interface with queued transfer capability. It allows up to 16 transfers to be queued at once, eliminating CPU intervention between transfers.

### 2.7.4.3 Periodic Interrupt Timers (PITs)

The two periodic interrupt timers are 16-bit timers that provide precise interrupts at regular intervals with minimal processor intervention. Each timer can either count down from the value written in its PIT modulus register, or it can be a free-running down-counter.

## 2.8 Chip Configuration

As on the MCF5206e, the MCF5208 needs some basic configuration information at reset. The MCF5206e latches the state of the IRQ pins at reset to determine the boot port size used for $\overline{CS0}$ and if the auto-acknowledge is used. The MCF5208 uses a similar mechanism to determine the boot chip select options, clocking mode, and bus drive strength. Table 12 compares the reset configuration on both devices.

**Table 12. MCF5206e vs. MCF5208 Reset Chip Configuration Comparison**

| Feature | MCF5206e | MCF5208 | Comments |
|---|---|---|---|
| Number of signals used for reset configuration | 3 interrupt signals | 8 data bus signals | |
| Latching configuration values | Configuration values are latched on the rising clock edge before $\overline{RSTI}$ negates. | Configuration values are latched on the rising clock edge before $\overline{RSTOUT}$ negates. | |
| Reset configuration required | Yes. | No. Reset configuration only needs to be used to override the default configuration values. | On the MCF5208 reset configuration is optional. If the default chip configuration values reflect the desired mode of operation at reset, then external circuitry to drive the reset configuration values is not needed. |

Table 13 shows the reset configuration options used for the MCF5208 including the default options used if the reset configuration pin ($\overline{RCON}$) is negated.

**Table 13. Configuration During Reset[1]**

| Pin(s) Affected | Default Configuration | Override Pins in Reset[2,3] | Function |
|---|---|---|---|
| None | RCON[1] = 0 | D1 | PLL Mode |
| | | 0 | 88/44 MHz operation (default) |
| | | 1 | 166.67/83.33 MHz operation |
| None | RCON[2] = 0 | D2 | Oscillator Mode |
| | | 0 | Crystal oscillator mode (default) |
| | | 1 | Oscillator bypass mode |

**Table 13. Configuration During Reset[1] (continued)**

| Pin(s) Affected | Default Configuration | Override Pins in Reset[2,3] | Function |
|---|---|---|---|
| None | RCON[4:3] = 00 | D[4:3] | Boot Device |
| | | 00 | External with 32-bit port[4] (default) |
| | | 01 | External with 16-bit port |
| | | 10 | External with 8-bit port |
| | | 11 | External with 32-bit port[4] |
| All output pins | RCON[5] = 0 | D5 | Output pad drive strength |
| | | 0 | Low Drive Strength (default) |
| | | 1 | High Drive Strength |
| None | RCON[6] = 0 | D6 | Limp Mode |
| | | 0 | PLL mode (default) |
| | | 1 | Limp mode |
| None | RCON[7] = 0 | D7 | Oscillator Frequency Select |
| | | 0 | 16 MHz (default) |
| | | 1 | 16.67 MHz |
| A[23:22]/$\overline{\text{FB\_CS}}$[5:4] | RCON[9] = 0 | D9 | Chip Select Configuration |
| | | 0 | A[23:22] = A[23:22] |
| | | 1 | A[23:22] = $\overline{\text{FB\_CS}}$[5:4] (default) |

NOTES:
1. Modifying the default configurations is possible only if the external $\overline{\text{RCON}}$ pin is asserted.
2. The D[31:10,8,0] pins do not affect reset configuration.
3. The external reset override circuitry drives the data bus pins with the override values while $\overline{\text{RSTOUT}}$ is asserted. It must stop driving the data bus pins within one FB_CLK cycle after $\overline{\text{RSTOUT}}$ is negated. To prevent contention with the external reset override circuitry, the reset override pins are forced to inputs during reset and do not become outputs until at least one FB_CLK cycle after $\overline{\text{RSTOUT}}$ is negated.
4. 32-bit port size is not available when DRAMSEL = 0. 16-bit port size is used as the default configuration for DRAMSEL = 0.

Figure 5 shows a block diagram of the recommended circuit used to drive the reset configuration values for the MCF5208.

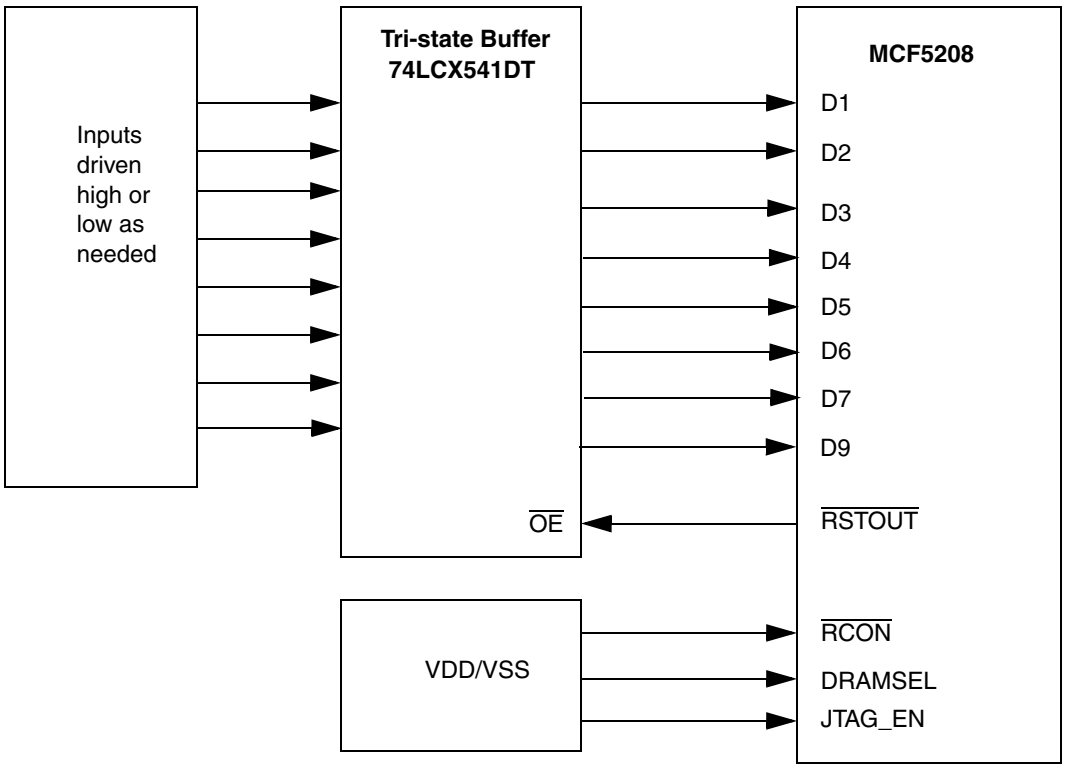


**Figure 5. Recommended Reset Configuration Circuit**

## 2.9 Interrupt Processing

In general, interrupts are handled by the ColdFire core in much the same way on the MCF5208 and they are on the MCF5206e. There are still seven interrupt levels with level seven being the highest priority unmaskable interrupt. However, vector generation and external interrupts function differently.

### 2.9.1 Interrupt Controller vs. System Integration Module Vector Generation

There are three different ways that the MCF5206e could determine the vector for an interrupt:

1. Autovectoring: the SIM generates the vector number (24 + the interrupt level) and returns the vector in response to the CPU's interrupt acknowledge (IACK) cycle.
2. Module interrupt vector register (IVR): some modules on the MCF5206e (the DMA, software watchdog, and UART) contain interrupt vector registers. The contents of the IVR will be returned in response to an IACK cycle for that module.
3. External IACK: the CPU runs a special external bus cycle and reads in a vector number from external memory or logic.

On the MCF5208, all vectors are generated by the interrupt controller so that IACK cycles are never issued to on-chip peripherals or the external bus. Since the interrupting module or device has no way of seeing the IACK cycle, appropriate actions should be taken to clear the interrupt request within the interrupt

service routine. This approach helps to simplify interrupt handling since all interrupts are treated the same way and there is never any need for external hardware or logic to supply vectors. In addition, each interrupt source is hard coded to a unique vector, unlike on the MCF5206e where if multiple interrupts are autovectored and have the same interrupt level, then they have the same vector. The MCF5206e approach means that multiple interrupts could share the same vector number, and therefore the interrupt service routine has to poll each of the possible interrupt sources to determine which one actually requested the interrupt.

### 2.9.2 EPORT vs. External Interrupts

The MCF5206e has three interrupt inputs that can be used as discrete active low external interrupt pins ($\overline{\text{IRQ}}$[7,4,1]) or as encoded interrupt priority levels (IPL[2:0]). The MCF5208 does not support encoded interrupt priority interrupt level inputs, but otherwise the MCF5208 has increased external interrupt functionality. The MCF5208 has up to three interrupt signals $\overline{\text{IRQ}}$[7,4,1] as part of the edge port (EPORT) module. The EPORT provides additional programmability for interrupt triggering. The MCF5206e only level detects interrupts, but each of the MCF5208's external interrupt pins can be configured for level-sensitive, rising edge triggered, falling edge triggered, or falling and rising edge triggered interrupts on a pin-by-pin basis.

Table 14 contrasts the features of the MCF5206e's interrupts pins and the MCF5208's EPORT.

**Table 14. External Interrupt Pin and EPORT Differences**

| Feature | Interrupt Pins | EPORT | Comments |
|---|---|---|---|
| Number of external interrupts | 3 | Up to 3 | $\overline{\text{IRQ4}}$ signal is muxed with $\overline{\text{DREQ0}}$ on the MCF5208 |
| Interrupt priority level inputs | 3 inputs decode to 7 interrupt levels | — | This feature is not supported on the MCF5208 |
| Programmable interrupt triggering | — | √ | Level, rising or falling edge, or both rising and falling edges |
| GPIO functionality | — | √ | The EPORT provides optional GPIO capability on external interrupt signal pins. |

## 2.10 Software Porting

One of the primary advantages of migrating an existing MCF5206e or design to the MCF5208 is the ease of software migration. Retargeting C or other high level programming language code to ColdFire is relatively easy, and the risk of introducing errors into the code during the process is low. In addition, the ISA_A+ used by the MCF5208 is a superset of the ISA_A used by the MCF5206e, therefore most assembly code should not require any changes. This makes porting code from MCF5206e to MCF5208 much easier than porting to a completely new architecture.

# 3 Summary

The MCF5208 offers a significant increase in performance and more functionality at a lower price point than the MCF5206e. There is little or no lost functionality, memory, or performance when migrating from the MCF5206e to the MCF5208/7. Furthermore, the additional set of MCF5208/7 on-chip peripheral modules can enable off-chip functions to be incorporated on chip, reducing hardware costs.

If the advantages gained from more functional integration and higher performance make this migration beneficial, this document should serve as a valuable reference to make the transition as smooth as possible.

# 4 References

**Table 15. References**

| Freescale Document Number | Title | Revision |
|---|---|---|
| MCF5208RM | MCF5208 Microprocessor Reference Manual | A.1 |
| MCF5208EC | MCF5208 Integrated Microprocessor Hardware Specifications | 0.5 |
| MCF5206eUM/D | MCF5206e User's Manual | 1 |
| CFPRM/D | ColdFire Family Programmer's Reference Manual | 3 |

# 5 Document Revision History

Table 16 provides a revision history for this application note.

**Table 16. Document Revision History**

| Rev. No. | Substantive Change(s) | Date of Release |
|---|---|---|
| 0 | Initial release | 05/2005 |

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

## 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






AN2981
Rev. 0
05/2005