Freescale Semiconductor
Application Note

Document Number: AN3613
Rev. 0, 03/2008

Using the MC9S12XS Family as a Development Platform for the MC9S12P Family

by: Lela Garofolo
Automotive Applications
Austin, TX

1 Introduction

The MC9S12P family will be the newest member of the automotive 16-bit microcontroller families. This family is intended to bridge the gap between high-end 8-bit microcontrollers and low-end 16-bit microcontrollers of the MC9S12XS family. The MC9S12P family uses many of the same features found on the MC9S12XS family, including error correction code (ECC) on flash memory, a separate data-flash module for diagnostic or data storage, an analog-to-digital converter (ATD), and a frequency-modulated phase-locked loop (IPLL).

The MC9S12P family is not available yet, but designers can begin development on applications intended for an MC9S12P family member using an MC9S12XS128 device. This document will help designers maintain compatibility when they use a member of the MC9S12XS family as an emulation tool for developing an application targeted at a member of the MC9S12P family. Allowing for minor constraints, as detailed in

Contents

freescale semiconductor

each of the following sections, code can be written for an MC9S12XS family device that can then run with minimal or no change on MC9S12P family devices.

This document is intended for use with the documents listed on the product summary page for S12 and S12X automotive products (see http://www.freescale.com/automotive).

Documents:

- Data sheet for the MC9S12P family
- Data sheet for the MC9S12XS family
- Product preview for the MC9S12P family
- Product brief for the MC9S12XS family
- S12XCPUV2 Reference Manual

Due to the timeline of devices launching from the MC9S12XS family and the MC9S12P family, this document is mainly written with the assumption that an MC9S12XS128 will be used for a development part that will transfer to an MC9S12P family member.

Throughout this document the following terms are used to describe similarities of the MC9S12XS and MC9S12P family:

- equivalent: like for like — there are no differences
- backwards-compatible: differences due to the removal of extra functionality, where one can be described as a subset of the other
- compatible: similar, with only a few differences

2 Overview of Families

Table 1 shows a summary of feature sets for the MC9S12XS128 and MC9S12P family devices. All MC9S12XS family devices are listed in the MC9S12XS family product brief. When using the MC9S12XS128 as a development platform for MC9S12P devices, some key differences must be taken into account.

Key considerations:

- Peripheral count
  - The MC9S12P family has a reduced peripheral set from the MC9S12XS family.
  - MCS12P family devices have a single SPI module, a ten-channel ATD module, and a six-channel PWM
- Memory maps
  - The MC9S12P family devices have less RAM and less data flash than a corresponding MC9S12XS family device. Mapping of these modules is backwards-compatible.
- Flash memory
  - The MC9S12XS family offers flash ranges from 256K to 64K; the MC9S12P family offers flash ranges from 128K to 32K.

- CPU
  - The MC9S12XS contains a CPU12X core and supports GPAGE, EPAGE, RPAGE, PPAGE, and DIRECT paging registers. It also contains instructions for directly accessing data in the global address map.
  - The MC9S12P contains a CPU12 core and only supports PPAGE and DIRECT paging registers.

Table 1. Family Summary

Device	Package	Flash	D-flash	RAM	MSCAN	SCI	SPI	TIM	PIT	PWM	ADC
MC9S12XS256	112 LQFP	256K	8K	8K	1	2	1	8 ch	4 ch	8 ch	16 ch
	80 QFP										8 ch
	64 LQFP										8 ch
MC9S12XS128	112 LQFP	128K	8K	8K	1	2	1	8 ch	4 ch	8 ch	16 ch
	80 QFP										8 ch
	64 LQFP										8 ch
MC9S12XS64	112 LQFP	64K	4K	4K	1	2	1	8 ch	4 ch	8 ch	16 ch
	80 QFP										8 ch
	64 LQFP										8 ch
MC9S12P128	80 LQFP	128K	4K	6K	1	1	1	8 ch	4 ch	6 ch	10 ch
	64 LQFP										
	48 QFN										
MC9S12P96	80 LQFP	96K	4K	6K	1	1	1	8 ch	4 ch	6 ch	10 ch
	64 LQFP										
	48 QFN										
MC9S12P64	64 LQFP	64K	4K	4K	1	1	1	8 ch	4 ch	6 ch	10 ch
	48 QFN										
MC9S12P32	64 LQFP	32K	4K	2K	1	1	1	8 ch	4 ch	6 ch	10 ch
	48 QFN										

3 MC9S12XS and MC9S12P Module Summary

3.1 MMC: Memory Mapping Control

The MMC module is backwards-compatible between the MC9S12XS family and MC9S12P family. The MMC module for the MC9S12P family is a subset of the MMC module on the MC9S12XS family. Consideration of the key differences will enable an easy transition from the MC9S12XS128 to a MC9S12P device.

The main differences are that the MC9S12P devices have a reduced global memory map of 128 KB, and the registers needed to support an 8 MB global map have been removed. The MC9S12P family features a 128 KB global memory address space accessible by translation of the PPAGE register, whereas the

MC9S12XS family has an 8 MB global memory space accessible by translation of the GPAGE and corresponding PPAGE, RPAGE, or EPAGE register. The GPAGE, EPAGE, and RPAGE register locations on the MC9S12XS devices become reserved address locations on the MC9S12P devices. Any accesses to these register addresses will be ignored on MC9S12P devices.

The MMCCTL control register for the MC9S12XS allows separate control of visibility in the global memory map of scratch RAM, D-flash IFRs, and P-flash IFRs via bits 7, 5, and 4, respectively. The MMCCTL register on the MC9S12P is changed, allowing control of D-flash and P-flash visibility in the global memory map or in PPAGE 0x01 via bit 0.

3.1.1 Device Memory Maps

The memory maps for the MC9S12XS and MC9S12P families are backwards-compatible. Careful consideration of the key differences will ensure minimal code changes, or even no changes, depending on usage. The majority of changes may be manageable with appropriate linker and compiler configurations.

Key differences:

- The total size of the global memory map for the MC9S12P family is smaller than the MC9S12XS family. The upper address boundary of the global map for the MC9S12P family is 0x03_FFFF — in the MC9S12XS family it is 0x7F_FFFF.
- The local PPAGE to global mapping for the P-flash has a global offset (or bias) of 0x40_0000 on the MC9S12XS and 0x00_0000 on the MC9S12P.
- The local memory map for the MC9S12P family does not have paging windows for the D-flash or RAM. These memories are fully accessible in the 64 KB local map.
- Unpaged P-flash is located at 0x1400 in the local map for the MC9S12P device, where as this is part of the RAM paging window for the MC9S12XS devices.
- On the MC9S12P the global map access is limited to flash programming and erase operations and for debug support. On the MC9S12XS it is fully readable and writable for data through use of dedicated global access instructions.

Figure 1 and Figure 2 illustrate a side-by-side comparison of the local memory map and the global memory maps.

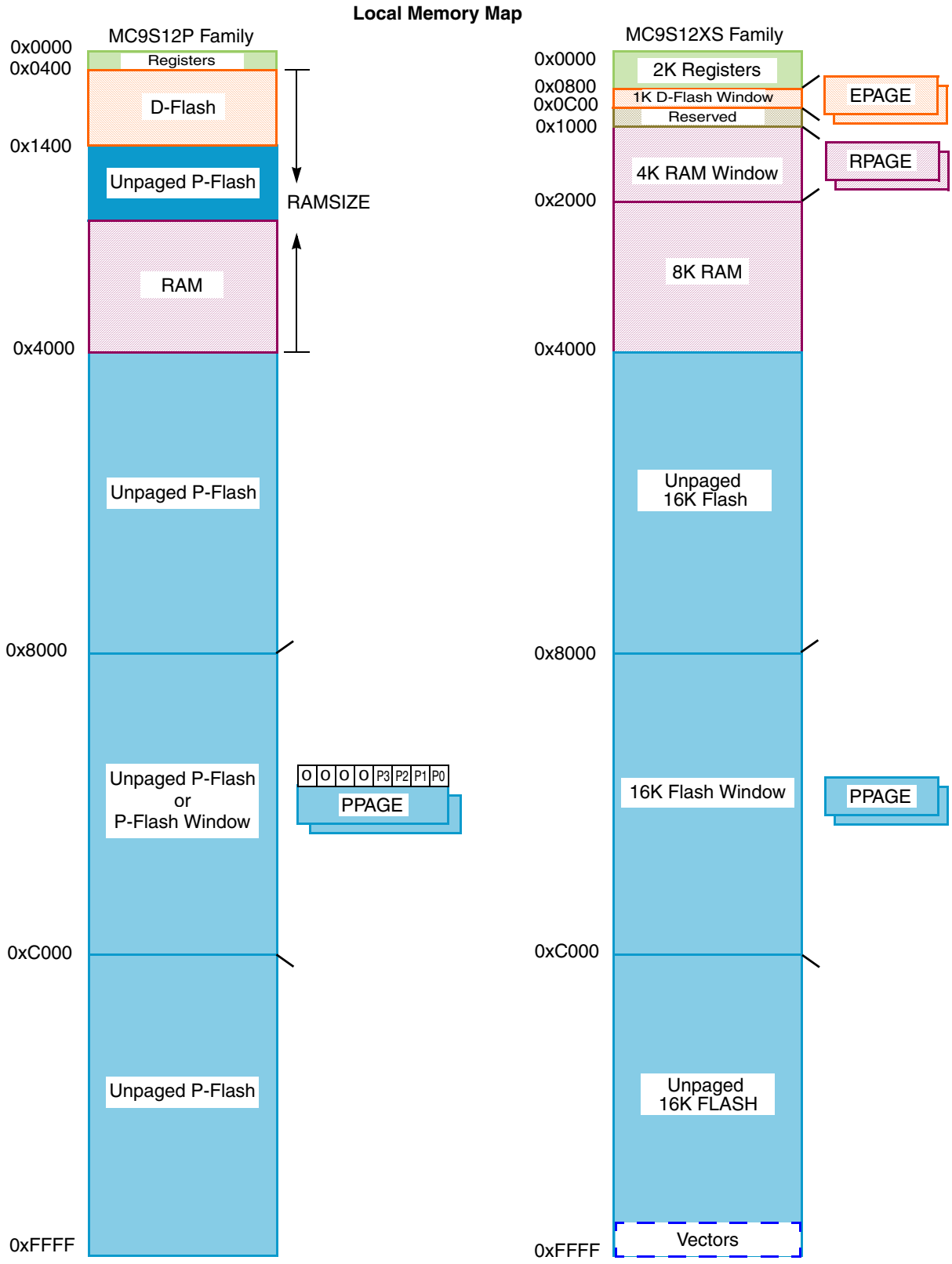

Figure 1. Comparison of Local Memory Map in MC9S12P and MC9S12XS

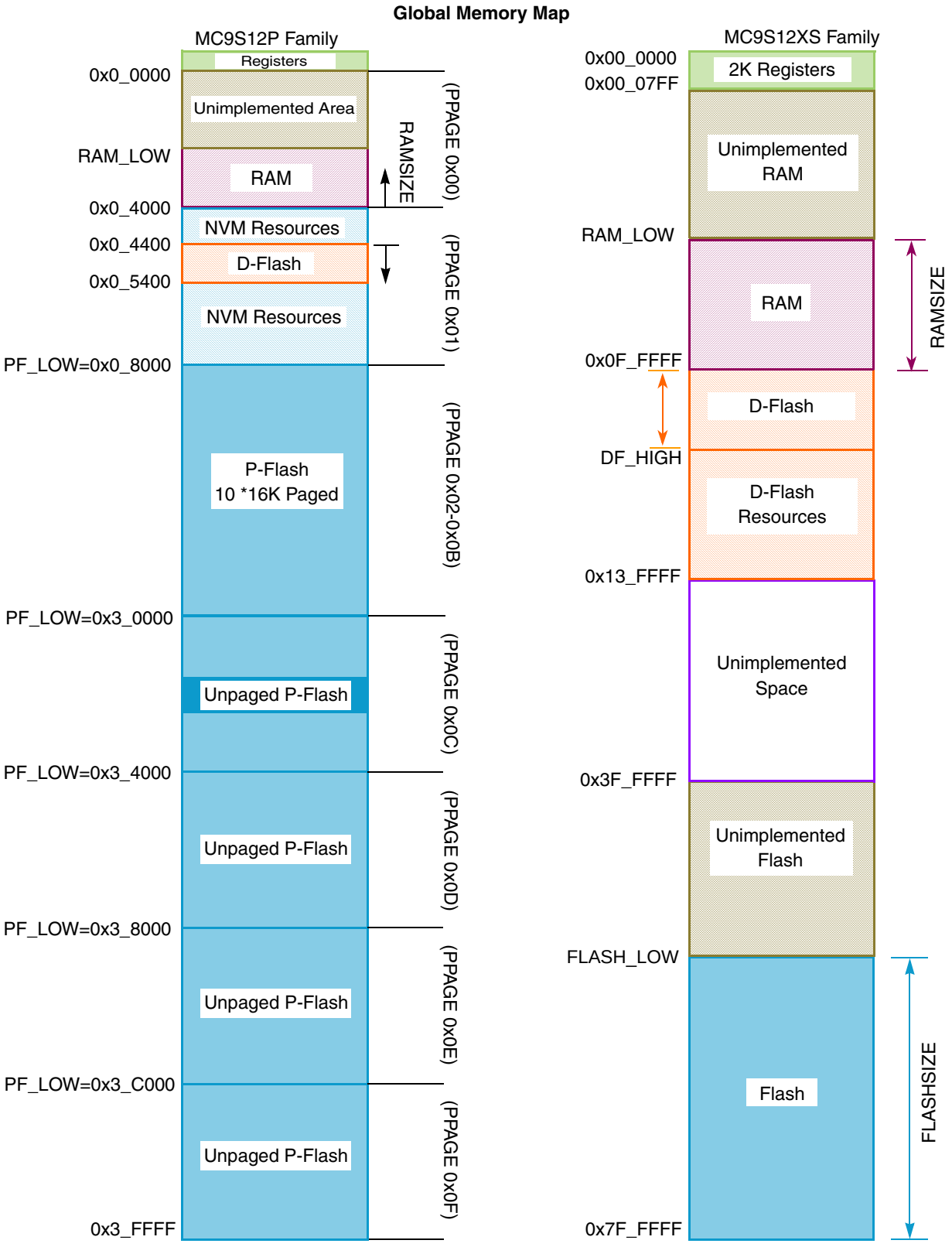

Figure 2. Comparison of Global Memory Map in MC9S12P and MC9S12XS

3.1.1.1 P-Flash Memory Maps

The MC9S12XS family offers larger flash sizes than the MC9S12P family.

On the MC9S12P family there is an unpaged region of P-flash in the lower address section of the local memory map from 0x1400 to (RAM_SIZE – 1). When developing code on an MC9S12XS device to move to an MC9S12P device the following points should be considered:

- Do not use global instructions to access the P-flash, as the global instruction set has been removed. This can be managed by configuring the compiler to generate code for an S12 CPU target when using the MC9S12XS as a development target for the MC9S12P.
- Do not access the unpaged P-flash in the local map in address locations 0x1400 to (RAM_SIZE – 1), which are available only on the MC9S12P. Accessing this via PPAGE, by writing to the PPAGE register and then via the page address in the region 0x8000 to 0xBFFF, will ensure code compatibility.
- The MC9S12P cannot access the global map directly. It is logical for an application to work in the local map when reading the P-flash. The CALL and RTC instructions (which manage PPAGE automatically) are equivalent on the MC9S12P and MC9S12XS, so code execution inherently operates in the local map. The different address map sizes can be managed by appropriate linker configurations.
- The relationship between the PPAGE addresses and global addresses (to get the global address take the PPAGE and append the lower fourteen bits of the local address) is compatible on the MC9S12P and the MC9S12XS, although the MC9S12XS has an additional global map offset or bias for the P-flash region of 0x40_0000. In other words, PPAGE 00 starts at global address 0x40_0000 on the MC9S12XS and at 0x00_0000 on the MC9S12P. This is primarily a consideration for flash programming routines, as the memory controller commands work with global map addresses. Figure 3 shows MC9S12P PPAGE to global addressing and Figure 4 the same for MC9S12XS.

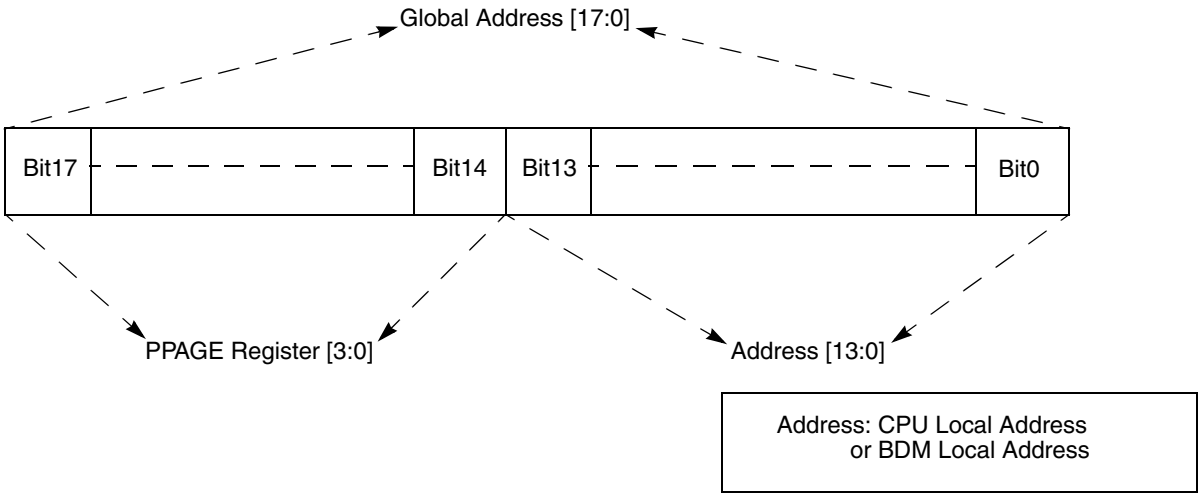

Figure 3. MC9S12P PPAGE to Global Addressing

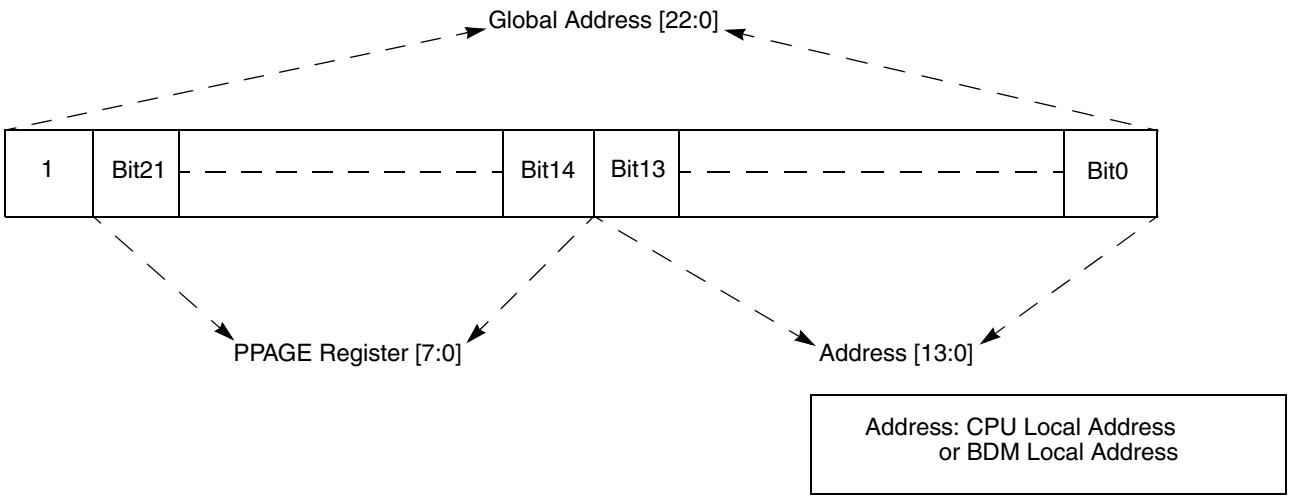

Figure 4. MC9S12XS PPAGE to Global Addressing

- Due to the smaller size of the MC9S12P global map (only 4 x 64K pages total) the location of the P-flash is effectively offset from the S12X family by 0xF0 PPAGEs, such that PPAGE 0xFF on the MC9S12XS is relative to PPAGE 0x0F on the MC9S12P. The four most significant bytes of the PPAGE register on the MC9S12P family are fixed to 0. Any write to these four bits will be ignored. Table 2 shows the comparative P-flash mapping side by side.

Table 2. MC9S12XS128 P-Flash Memory Boundaries Relative to the MC9S12P128

S12XS128			S12P128		
PPAGE	local	global	PPAGE	local	global
–	(non-banked) 0xC000–0xFFFF	0x7F_C000–0x7F_FFFF	0x0F	(non-banked) 0xC000–0xFFFF	0x03_C000–0x03_FFFF
0xFF	ppage window 0x8000–0xBFFF		0x0F	ppage window 0x8000–0xBFFF	
0xFE	ppage window 0x8000–0xBFFF	0x7F_8000–0x7F_BFFF	0x0E	ppage window 0x8000–0xBFFF	0x03_8000–0x03_BFFF
–	(non-banked) 0x4000–0x7FFF	0x7F_4000–0x7F_7FFF	–	(non-banked) 0x4000–0x7FFF	0x03_4000–0x03_7FFF
0xFD	ppage window 0x8000–0xBFFF		0x0D	ppage window 0x8000–0xBFFF	
0xFC	ppage window 0x8000–0xBFFF	0x7F_0000–0x7F_3FFF	–	(non-banked) 0X1400–RAM_LOW	0x03_1400–(0x03_1400 + (RAM_LOW-0x1400))
			0x0C	ppage window 0x8000–0xBFFF	0x03_0000–0x03_3FFF

Table 2. MC9S12XS128 P-Flash Memory Boundaries Relative to the MC9S12P128 (continued)

S12XS128			S12P128		
0xFB	ppage window 0x8000–0xBFFF	0x7E_C000–0X7E_FFFF	0x0B	ppage window 0x8000–0xBFFF	0x02_C000–0X02_FFFF
0xFA	ppage window 0x8000–0xBFFF	0x7E_8000–0X7E_BFFF	0x0A	ppage window 0x8000–0xBFFF	0x02_8000–0x02_BFFF
0xF9	ppage window 0x8000–0xBFFF	0x7E_4000–0X7F_FFFF	0x09	ppage window 0x8000–0xBFFF	0x02_4000–0x02_FFFF
0xF8	ppage window 0x8000–0xBFFF	0x7E_0000–0X7E_3FFF	0X08	ppage window 0x8000–0xBFFF	0x02_0000–0x02_3FFF

3.1.1.2 D-Flash Memory Maps

The D-flash mapping for the MC9S12P family is another difference from the MC9S12XS family. The global map for the MC9S12XS family is address range 0x10_0000 to D-flash high, while the MC9S12P is programmable at 0x00_4400 to 0x00_5400 in the global map and readable at 0x1400 to 0x13FF in the local (non-paged) map.

The MC9S12XS devices can access only 1K of D-flash at a time in the local map through the EPAGE window at 0x0C00–0x0FFF. MC9S12P devices can read the whole 4K.

3.1.1.3 RAM Memory Maps

The RAM on the MC9S12XS family and MC9S12P family is backwards-compatible, with only slight differences in accessing memory maps and available sizes. The MC9S12P family does not have a RPAGE window, and does not support global access due to the absence of global instructions. The MC9S12P family supports RAM sizes of 6K, 4K, and 2K. When using an MC9S12XS128 for development, the key points to consider are:

- RAM addresses are compatible where implemented areas are equivalent. For maximizing compatibility between the MC9S12XS and MC9S12P families, do not use the 4K RAM window on the MC9S12XS device.
- Do not use the RPAGE register when converting code from a MC9S12XS128 to MC9S12P family device. All writes to the RPAGE register will be ignored on the MC9S12P devices.
- Do not access RAM using the global map. When code is converted to run on an MC9S12P device, all global commands and GPAGE register writes will be ignored.

3.2 Peripheral Set

The peripheral set on the MC9S12P family is a reduced set from the MC9S12XS family. The MC9S12P family does not have a PIT, has two fewer PWM channels, and one less SCI. The MC9S12P device's ADC is a 10-channel 12-bit resolution module, whereas the MC9S12XS device has the same ADC module with up to 16 channels available, depending on the package used. The total amount of GPIO is improved due to the reduced power supply pairs on the MC9S12P.

3.3 PIM

The PIM module on the MC9S12P family is a reduced version of that on the MC9S12XS family. The modules are compatible with the exception of the absence of some of the ports and peripherals on the MC9S12P devices. For those ports that remain on the MC9S12P, the register address locations, functionality of GPIO, signal routing, and electrical pin properties are similar for both families. For ports not available on the MC9S12P family, the corresponding register address locations on the MC9S12XS devices are reserved. Any accesses to a reserved address location in the PIM will be ignored. Consideration of the reserved register address locations will enable code to be transitioned directly from a MC9S12XS device to a MC9S12P device.

The register differences to be considered between the MC9S12XS and MC9S12P family are listed below. The format used below is the register abbreviation used in the data sheets and bit numbers, with a colon used to signify ‘through’ and a comma used to designate ‘singularity.’

- Port T, Addresses: 0x240–0x246
  - S12XS:
    PTT7:6 PWM channels 7:6 if enabled and bits 7:4 in the PTTRR register are set
  - S12P:
    PTT7:6 reserved
  - To maintain compatibility, do not use PTT7:6 for PWM functions.
- Port T routing register (PTTRR), Address: 0x247
  - S12XS:
    PTTRR7:4 reroutes PWM channels 7:6
    PTTRR2:1 reroutes Timer channels 2:1
    PTTRR0 reroutes Timer channel 0
  - S12P:
    PTTRR7:6 reserved
    PTTRR2:1 reserved
    PTTR0 reroutes PWM channel 0
  - To maintain compatibility, do not use PTTRR.
- Port S, Addresses: 0x248–0x24E
  - S12XS:
    PTS7:4 SPI0 if enabled, or GPIO
    PTIS7:4 GPIO
    DDRS7:4 GPIO
    RDRS7:4 GPIO
    PERS7:4 GPIO
    PPSS7:4 GPIO
    WOMS7:4 GPI
  - S12P:

PTS7:4 reserved
PTIS7:4 reserved
DDRS7:4 reserved
RDRS7:4 reserved
PERS7:4 reserved
PPSS7:4 reserved
WOMS7:4 reserved

— To maintain compatibility, do not use GPIO functionality on PTS7:4. Details for maintaining compatibility across families for SPI0 is given below in Port M data register.

- Port M, Addresses: 0x250 – 0x256
  — S12XS:
  PTM7:6 GPIO
  PTM5:2 SPI0 if enabled and routed by setting MODRR bit 4
  PTM1:0 CAN if enabled, and SCI0 if enabled and routed by setting MODRR bit 7
  PTIM7:6 GPIO
  DDRM7:6 GPIO
  RDRM7:6 GPIO
  PERM7:6 GPIO
  PPSM7:6 GPIO
  WOMM7:6 GPIO
  — S12P:
  PTM7:6 reserved
  PTM5:2 SPI if enabled or GPIO
  PTM1:0 CAN if enabled or GPIO
  PTIM7:6 reserved
  DDRM7:6 reserved
  RDRM7:6 reserved
  PERM7:6 reserved
  PPSM7:6 reserved
  WOMM7:6 reserved
  — Code compatibility can be maintained when developing on an MC9S12XS device by re-mapping the SPI0 to PTM5:2 via the MODDR bit 4. Writes to this reserved location on the MC9S12P will have no effect, so code that maps peripherals on the MC9S12XS to maximize peripheral compatibility can execute without change. SCI0 should not be re-mapped to PTM on the MC9S12XS, as re-mapping options are not available on the MC9S12P.
- Module routing register (MODRR), Address 0x257
  — S12XS:
  MODRR7:0 Controls alternative peripheral routing for SCI1 and SPI0.

  - S12P:

    MODRR7:0 Reserved
  - See details for PTM5:2 to maintain compatibility for SPI0.
- Port P, Addresses 0x258–0x25F
  - S12XS:

    PTP7:3 Controls PWM7:3 if enabled, or GPIO

    PTP2:0 Controls PWM2:0 if enabled, TIM2:3 if enabled and routed by setting PTTRR bits 2:0, SCI1 if enabled (PTP2,0) and routed by setting MODRR bit 6, or GPIO

    PTIP6 GPIO

    DDRP6 GPIO

    RDRP6 GPIO

    PERP6 GPIO

    PPSP6 GPIO

    PIEP6 GPIO

    PIFP6 GPIO
  - S12P:

    PTP7 GPIO

    PTP6 reserved

    PTP5:0 PWM5:0 if enabled, or GPIO

    PTIP6 reserved

    DDRP6 reserved

    RDRP6 reserved

    PERP6 reserved

    PPSP6 reserved

    PIEP6 reserved

    PIFP6 reserved
  - To maintain compatibility between the two families, only use PWM5:0 (which corresponds to PTP5:0) and PTP7 for GPIO functions.
- Port H, Addresses 0x260–0x267
  - S12XS:

    All port H registers control GPIO functions
  - S12P:

    All port H registers are reserved
  - To maintain compatibility between the two families, don’t use port H.
- Port J, Addresses 0x268–0x26F
  - S12XS:

    For port J, registers bit 2 is reserved

— S12P:
For port J, registers bit controls GPIO

- Port AD, Addresses 0x270, 0x272, 0x274, 0x276
  — S12XS:
  PT0AD7:2 ATD channels 15:10 if enabled, or GPIO
  DDR0AD7:2 GPIO
  RDRAD7:2 GPIO
  PER0AD7:2 GPIO
  — S12P:
  PT0AD7:2 reserved
  DDR0AD7:2 reserved
  RDRAD7:2 reserved
  PER0AD7:2 reserved
  — To maintain compatibility, use only ATD channels 9:0 that correspond to those in PT0AD1:0. Also, use only PT1AD7:0 for ATD or GPIO functions.

3.4 Interrupt Controller (INT)

The MC9S12P INT module has less functionality than the MC9S12XS INT module. The ability to assign the I bit to one of seven priority levels is not supported on the MC9S12P family. For MC9S12P devices, the interrupt priority decoder is dependent only on the vector address. If more than one interrupt request is pending, the interrupt request with a higher order vector address is serviced. For the MC9S12XS devices, the priority for interrupt requests with a vector address lower or equal to vector base + 0x00F2 will default to a priority level 1. To maintain compatibility when developing code to transition to the MC9S12P family, all interrupts should be left with a default value of level 1. This will result in behavior similar to the MC9S12P devices, because when there is more than one interrupt request with the same priority level, the higher vector order address wins prioritization.

Some of the vector address locations are different in the MC9S12XS family than they are in the MC9S12P family. Table 3 summarizes the differences side by side. Minor modifications may be necessary when transitioning code from an MC9S12XS device to an MC9S12P device. Note that vector differences between the families that result from the removal of a module on the MC9S12P are not listed in this table.

Table 3. Vector Address Comparison Between MC9S12XS and MC9S12P

Vector Address	Interrupt Source MC9S12XS Family	Interrupt Source MC9S12P Family
vector base + 0x10	Spurious interrupt	reserved
vector base + 0x3E	ATD0 compare interrupt	reserved
vector base + 0x7C	High temperature interrupt	reserved
vector base + 0x7E	Autonomous periodic interrupt (API)	reserved
vector base + 0x80	Low-voltage interrupt (LVI)	Spurious interrupt
vector base + 0x84	reserved	ATD compare interrupt

Table 3. Vector Address Comparison Between MC9S12XS and MC9S12P (continued)

Vector Address	Interrupt Source MC9S12XS Family	Interrupt Source MC9S12P Family
vector base + 0x86	reserved	High temperature interrupt
vector base + 0x88	reserved	Autonomous periodic interrupt (API)
vector base + 0x8A	reserved	Low-voltage interrupt (LVI)
vector base + 0xC8	reserved	CPMU oscillator noise (S12P only)

3.5 Debug Module (DBG)

The DBG module on the MC9S12P family is backwards-compatible with the MC9S12XS family. The minor changes are to incorporate the new global memory map to address a linear space of 256 KB on the MC9S12P family, a reduction from the 8 MB linear global map on the MC9S12XS family. The end user's main concern with changes to these modules will be to use development tools that have been updated to incorporate the changes.

The notable changes are:

- S12XS:
  - There are four comparators: A, B, C, and D.
  - The trace buffer is sixty-four lines of sixty-four bits.
  - The state sequencing is comprehensive.
  - Range filtering is supported (TRANGE bits).
  - Start-, Mid-, and End-aligned triggering is supported.
- S12P:
  - There are three comparators: A, B, and C.
  - The trace buffer is sixty-four lines of twenty bits.
  - The state sequencer has fewer state change possibilities than on MC9S12XS.
  - Range filtering is not supported.
  - Start- and End-aligned triggering is supported.

3.6 Clocks and Reset Generator (CRG) versus Clock, Reset, and Power Management Unit (CPMU)

The MC9S12P family has a CPMU module in place of the CRG module that is on the MC9S12XS family. The CPMU module controls all of the clocking, reset, and power management functions of the device. The key features of the CPMU module are a Pierce oscillator (OSCLCP) for external clock source, voltage regulator (IVREG) and voltage monitoring features, phase locked loop (PLL), and a trimmable internal reference clock (IRC1M) of 1 MHz. The MC9S12XS family contains the clock and reset function in the CRG module. Power management functions are contained in a separate module, the voltage regulator (VREGL3V3).

Other key features of the CPMU on the MC9S12P are:

- Inherent fast startup from reset and stop on the internal clock source. By default, the device starts running from the IPLL with the internal IRC1M selected as its input reference. The external oscillator is not enabled by default following reset, and needs to be enabled by software (if required).
- An Adaptive spike filter to improve PLL stability in the presence of noise on the oscillator clock.

For CAN communications, an external oscillator is required. Where CAN is not required, MC9S12P designs that save the cost of the external oscillator can be easily managed.

Combining the power management functions with the clock and reset functions, along with the addition of the internal reference clock into the CPMU module on the MC9S12P, makes the family ideal for optimizing low-cost applications. While much of the CPMU peripheral functionality is compatible with the MC9S12XS, the enhanced clocking scheme means that the differences between the two targets are too detailed for the scope for this document. A separate application note, *Comparison of the S12X Family CRG Module with MC9S12P CPMU Module* (AN3622), is available online at www.freescale.com. AN3622 describes in depth the key differences to be considered.

3.7 Periodic Interrupt Timer (PIT)

The functionality of the PIT module is not available on the MC9S12P family. Therefore, to maintain compatibility between the MC9S12P and MC9S12XS families, do not use the PIT.

3.8 Pulse-Width Modulator (PWM)

The functionality of the PWM module on the MC9S12P family is compatible with the MC9S12XS family. The MC9S12P family has six PWM channels available across all package options, whereas the MC9S12XS has eight PWM channels on all package options. To maintain compatibility, use only channels PWM0:5. Additional routing options are available on the MC9S12P family for PWM0 that are not available on the MC9S12XS. To maintain compatibility between the families, PWM0 should be left to default (reset value) routing.

3.9 Scalable Controller Area Network (MSCAN)

The functionality of the MSCAN module on the MC9S12P family is compatible with the MC9S12XS family. Signal properties, routing, and register locations are equivalent.

3.10 Serial Peripheral Interface (SPI)

The functionality of the SPI module on the MC9S12P family is compatible with the MC9S12XS family. The default routing of the SPI between the families is different. The MC9S12P defaults to port M, PTM5:2. The MC9S12XS defaults to port S, PTS7:4, and has the option of re-routing to port M, PTM5:2, via the MODRR register. To maintain compatibility when developing code on an MC9S12XS, the SPI should be re-routed to PTM5:2 by writing to the MODRR register. Code transferred to an MC9S12P device will ignore the MODRR write, because it is a reserved address location on the MC9S12P devices.

3.11 Serial Communication Interface (SCI)

The functionality of SCI module on the MC9S12P family is compatible with the MC9S12XS family. The MC9S12P family has only one SCI module, whereas the MC9S12XS has two SCI modules. When developing code on an MC9S12XS device to transition it to an MC9S12P device, use SCI0. Signal properties, routing, and register locations are equivalent for SCI0.

3.12 Analog to Digital Converter (ATD)

The functionality of the ATD module on the MC9S12P family is compatible with the MC9S12XS family. The MC9S12P devices have ten channels available across all package options. The MC9S12XS family has sixteen channels available on the 112 LQFP package, and eight channels available on the 80 QFP and 64 LQFP packages. If all ten channels will be needed on the MC9S12P family, then development should be done on the MC9S12XS 112 LQFP package. Also, to maintain compatibility, only channels AN0:9 should be used. In addition, writing the ATDCTL0 register with 0x09 will cause the ATD on the MC9S12XS to wrap around to channel 0 once channel 9 has been converted in a multi-channel conversion sequence.

As there is no PIT module on the MC9S12P, there are no options for triggering ATD conversions from the PIT, as there are on the MC9S12XS. To maintain compatibility, do not use the PIT to trigger ATD conversion sequences.

3.13 Timer Module (TIM)

The functionality of the TIM module on the MC9S12P family is compatible with the MC9S12XS family. The MC9S12XS family has additional routing options via the PTTRR register that are not available on the MC9S12P family. Both families have the same default (reset value) routing. Therefore, to maintain compatibility the TIM should be left with default routing.

3.14 Background Debug Module (BDM)

On the MC9S12XS, the default clock for the BDM module is the external oscillator / 2 ($f_{osc}/2$). The bus clock can be selected as an alternative clock by setting the CLKSW bit in the BDMSTS register via a BDM hardware command.

On the MC9S12P the default clock for the BDM module is the PLL f_{VCOCLK} / 8. There is no alternate BDM clock or CLKSW bit. The valid range of f_{VCOCLK} is 32 to 64 MHz, so the BDM clock source range will be 4 to 8 MHz. Out of reset the PLL is configured for fVCOCLK = 64 MHz, and the BDM clock = 8 MHz.

The SYNC command is supported by both the MC9S12XS and the MC9S12P, and BDM cables can use this to determine the BDM clock rate.

3.15 Oscillator

The MC9S12XS Pierce oscillator module supports Loop-Controlled Pierce, Full-Swing Pierce, and External Clock configurations. The MC9S12P supports the Loop-Controlled Pierce configuration. For

maximum compatibility when using an MC9S12XS, configure it for Loop-Controlled Pierce operation between 4 to 16 MHz.

4 CPU

The MC9S12P family has the CPU12-1 module, which is a reduced version of the CPU12X-1 on the MC9S12XS family. Both CPU12-1 and CUP12X-1 are covered in the same manual, *CPU12 & CPU12X Reference Manual*. The key differences to note are:

- Instructions set
- Programmer's model and stacking operations
- Bus signature — access details
- Global map access

The instruction set for the MC9S12P family is a sub-set of the MC9S12XS family. Compatibility can be maintained by using only those instructions available on the MC9S12P. This can be achieved configuring the compiler / assembler to generate code for an S12 core as opposed to an S12X core. The MC9S12P core is missing the fuzzy logic instructions from the original S12 core — as these are not available on the MC9S12XS or used by C compilers, this should not be an issue.

The condition code register (CCR) on the MC9S12P family does not have IPL2:0 control bits. The CCR is eight bits wide on the MC9S12P family, whereas the CCR is sixteen bits wide on the MC9S12XS. Therefore, the interrupt stack frame is nine bytes instead of ten bytes. Also, the bus access for some of the CPU12-1 instructions has more cycles than the CPU12X-1 (EMUL, EMULS, EMACS, and ETBL). In general, these noted differences should have little impact on the end user when transitioning code from an MC9S12XS to an MC9S12P, but are worth mentioning.

5 NVM Compatibility

The P-flash and D-flash are functionally compatible for the MC9S12P and MC9S12XS devices for modes of operation and for flash command operations. The functionality of the ECC fault detection is equivalent for both families, but the ECC reporting registers (FECCR) have been removed from the MC9S12P devices.

The P-flash erase sector size is 1024 bytes on the MC9S12XS and 512 bytes on the MC9S12P.

On the MC9S12XS the FTM write once command can be executed from code in P-flash or in RAM, or directly via the BDM. On the MC9S12P it cannot be executed by code in flash. As this is something typically done during production programming or module testing, this is not expected to be a significant issue.

On the MC9S12XS the low level timing for flash programming / erase operations is derived from the external oscillator. On the MC9S12P it is derived from the bus clock (as the MC9S12P is able to operate without an external oscillator in some designs). This results in a different clock divider value being required to be written to FCLKDIV.

On the MC9S12XS the FCLKDIV register is write once after reset. On the MC9S12P the clock divider value is writable any time after reset until a new FDIVLCK bit in the FCLKDIV register is set. After FDIVLCK is set, the flash clock divider is fixed until a reset occurs.

6 Part IDs

The part ID is located in two 8-bit registers: PARTIDH and PARTIDL (addresses 0x001A and 0x001B). The read-only value is a unique part ID for each device revision. Table 4 shows the assigned part ID number and mask-set number for each device. The unique PARTID values can be used to enable target-aware source code for minimizing code changes when transferring to an MC9S12P.

Table 4. Part ID Number and Mask-Set Number by Device

Device	Mask Set Number	Part ID
MC9S12XS256	0M05M	0xC0C0
MC9S12XS128	0M04M	0xC1C0
MC9S12XS64	0M04M	0xC1C0
	TBD	0xC2C0
MC9S12P128	0M01N	0x3980
MC9S12P96	0M01N	0x3980
MC9S12P64	0M01N	0x3980
	TBD	TBD
MC9S12P32	0M01N	0x3980
	TBD	TBD

The MC9S12XS64 uses the MC9S12XS128 die tested for MC9S12XS64 functionality only. An actual MC9S12XS64 device may be tested at a later time. The MC9S12P96, MC9S12P64, and MC9S12P32 use the MC9S12P128 die tested for MC9S12P96, MC9S12P64, and MC9S12P32 functionality only, respectively. An actual MC9S12P96, MC9S12P64, or MC9S12P32 device may be tested at a later time. A product change notification will be posted in such an event. The PartlD register should be verified for the actual device.

7 Electrical Characteristics

The electrical characteristics for the MC9S12P family values are currently TBD (to be determined). They will remain TBD until the final silicon is validated and characterized. However, it should be noted that the values will be targeted to be the same or similar to the MC9S12XS family.

8 Packages and Pin Assignments

Table 1 summarizes the package options for the MC9S12XS and MC9S12P families. The common package options for the two families are 80 QFP and 64 LQFP. With the changes in the PIM, and the change from a CRG to a CMPU, there are differences in the pinouts. Table 5 below shows a side-by-side

comparison of the differences between the 80 QFP and 64 LQFP. The pins not listed can be assumed equivalent.

When you are preparing to begin development and are ready to select a package option from the MC9S12XS family, one key point to consider is the number of ATD channels and GPIO ports. The 80 QFP for the MC9S12XS does not have all ten ATD channels available, and the change to the CMPU frees up some of the voltage and oscillator pins to be used as GPIO. The MC9S12P package options have five additional GPIO pins. Therefore, to take advantage of all ten ATD channels and the five additional GPIO pins, development will have to be done on a 112 LQFP MC9S12XS part. Details of PIM differences are given in Section 3.3, "PIM."

Table 5. Pinout Differences for Equivalent Packages

	S12XS	S12P
Pin	80 QFP	
2	TXD1/IOC2/PWM2/KWP2/PP2	PWM2/KWP2/PP2
3	IOC1/PWM1/KWP1/PP1	PWM1/KWP1/PP1
4	RXD1/IOC0/PWM0/KWP0/PP0	PWM0/KWP0/PP0
5	IOC0/PT0	PWM0/IOC0/PT0
9*	V_{DDF}	KWJ0/PJ0
10*	V_{SS1}	KWJ1/PJ1
12	V_{reg_API}/PWM5/IOC5/PT5	PWM5/IOC5/PT5
13	PWM6/IOC6/PT6	IOC6/PT6
14	PWM7/IOC7/PT7	IOC7/PT7
24	XCLKS/ECLKX2/PE7	PE7
36	V_{DDPLL}	KWJ2/PJ2
49*	V_{DD}	PAD08/AN08
50*	V_{SS2}	PAD09/AN09
65	RXD1/PS2	PS2
66	TXD1/PS3	PS3
74	TXD1/TXCAN0/PM1	TXCAN0/PM1
75	RXD1/RXCAN0/PM0	RXCAN0/PM0
78	PWM7/KPW7/PP7	KPW7/PP7

	S12XS	S12P
Pin	64 LQFP	
2	TXD1/IOC2/PWM2/KWP2/PP2	PWM2/KWP2/PP2
3	IOC1/PWM1/KWP1/PP1	PWM1/KWP1/PP1
4	RXD1/IOC0/PWM0/KWP0/PP0	PWM0/KWP0/PP0
5	IOC0/PT0	PWM0/IOC0/PT0
9*	V_{DDF}	KWJ0/PJ0
10*	V_{SS1}	KWJ1/PJ1
12	V_{reg_API}/PWM5/IOC5/PT5	PWM5/IOC5/PT5
13	PWM6/IOC6/PT6	IOC6/PT6
14	PWM7/IOC7/PT7	IOC7/PT7
20	XCLKS/ECLKX2/PE7	PE7
30	V_{DDPLL}	KWJ2/PJ2
37*	V_{DD}	PAD08/AN08
38*	V_{SS2}	PAD09/AN09
52	RXD1/PS2	PS2
53	TXD1/PS3	PS3
59	TXD1/TXCAN0/PM1	TXCAN0/PM1
60	RXD1/RXCAN0/PM0	RXCAN0/PM0
63	PWM7/KPW7/PP7	KPW7/PP7

* Power supply pairs on MC9S12X requiring decoupling. GPIO on MC9S12P.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

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

Document Number: AN3613
Rev. 0
03/2008

freescale™
semiconductor