Freescale Semiconductor
Application Note

Document Number: AN3510
Rev. 0, 8/2007

S12HZ and S12XHZ Family Compatibility

by: Steve McAslan
Microcontroller Division, East Kilbride

1 Introduction

The 9S12XHZ512 MCU provides significant performance improvements over the existing 9S12HZ family through a combination of increased clock speed and enhanced functionality. You can take advantage of the increased speed of operation of the S12XHZ family almost immediately due to its very high level of backward compatibility. Further performance benefits are possible by optimizing the application design to take advantage of the new feature set.

This document describes the notable differences between the S12HZ family and the S12XHZ family.

All differences noted are based on the feature set of the 9S12XHZ512. Note that other derivatives of the S12XHZ family might not feature all of the functions described here. Refer to the appropriate data sheet for details.

The following list summarizes the differences between the S12XHZ family and the S12HZ family.

Contents

freescale™ semiconductor

- 40 MHz operation
- 80 MHz XGATE peripheral coprocessor
- Extended CPU instruction set
- Programmable (eight-level) interrupt controller
- Enhanced memory management controller (MMC)
- Four additional motor control (MC) pulse width modulation (PWM) modules and two additional stepper stall detectors (SSD)
- New four-channel periodic interrupt timer (PIT)
- Enhanced capture timer (ECT) with modulus prescaler options
- Non-multiplexed 8 Mbyte expanded memory bus (EBI)
- New low-power autonomous periodical interrupt (API) and fast recovery from STOP modes
- Decimal prescaler for real time interrupt (RTI) module
- Improved serial communication interface (SCI) featuring hardware bit manipulation for LIN
- Enhanced trigger source options for analog to digital converters
- Amplitude-controlled Pierce oscillator
- Wider and deeper background debug module (BDM)

2 CPU and Instruction Set

The S12XHZ features the new S12XD CPU. This CPU includes enhancements to the programmer model, stacking operations, and the instruction set.

2.1 Programmers Model

The CPU features an enhanced Condition Code Register (CCR). This register has been extended to 16 bits to allow stacking of the interrupt priority.

The additional bits in the CCR are shown in Figure 1. IPL[2:0] indicates the interrupt level of the CPU for the current interrupt.

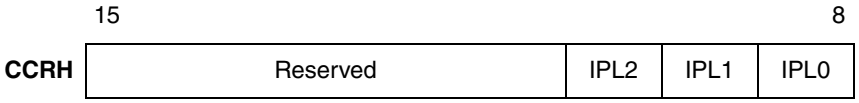

Figure 1. High Byte of Condition Code Register

The CPU automatically updates the value of IPL[2:0] to the value of the interrupt currently being serviced.

2.2 Interrupt Stacking Operation

The CCR has been extended from one to two bytes. This causes the interrupt stack frame to increase by one byte (from nine to ten bytes); therefore, all stack relative accesses are modified by one byte.

Figure 2 provides an example of a stack frame on an S12HZ after an interrupt has occurred.

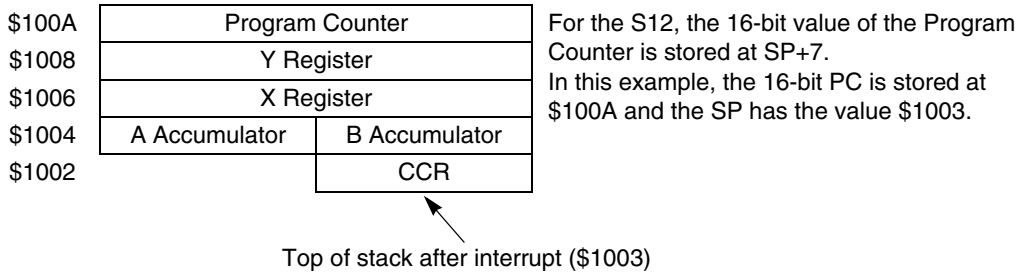

Figure 2. Stack Frame Example for S12

Figure 3 provides an example of a stack frame on an S12XHZ after an interrupt has occurred.

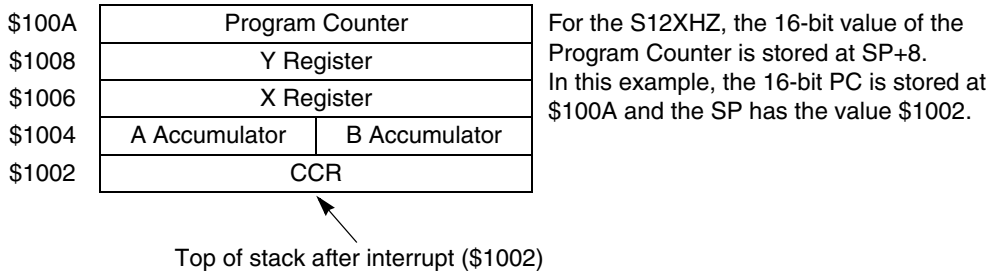

Figure 3. Stack Frame Example for S12XHZ

In practice, the requirement to extract information, such as the program counter from an interrupt stack frame, is an unusual activity (typically related to debug tools or perhaps, task schedulers); therefore, for the vast majority of users, this difference between the S12HZ and S12XHZ will have little impact.

2.3 Instruction Set

The S12XHZ features an enhanced instruction set over the S12HZ. The new CPU retains all of the existing S12HZ CPU instructions.

There are four classes of new instructions as follows:

1. New 16-bit, where only an 8-bit accumulator operation existed
2. New memory access instructions, allowing access to linear banks of up to 64 Kbytes
3. New instruction designed to optimize semaphore handling
4. New addressing modes for MOVe instructions

Class 1 improves the data manipulation capabilities of the CPU by allowing direct operation on larger data sizes. On the S12HZ, most arithmetic and logical operations, such as addition, can only take place by using the A, B or D accumulators. The S12XHZ extends this capability to the X and Y registers and adds new instructions for the D register. All arithmetic and logical functions using the A or B accumulator now have a 16-bit counterpart using the X and Y register. New instructions of this type are as follows:

- ADE (add with carry) and ADD (add without carry)
- SBE (subtract with carry) and SUB (subtract without carry)
- DEC (decrement) and INC (increment)
- AND (logical AND), OR (logical OR) and EOR (logical EXCLUSIVE OR)
- NEG (two's complement) and COM (one's complement)
- CLR (clear register)
- BIT (logical bit test) and TST (test register)
- LSL (logical shift left) and LSR (logical shift right)
- ASR (arithmetic shift right) and ASL (arithmetic shift right)
- ROR (rotate right) and ROL (rotate left)

These new instructions have the same addressing modes as their 8-bit counterparts.

To improve the 32-bit capability of the D-Accumulator, ADED (add with carry) and SBED (subtract with carry) are added. In addition, the CPU provides a set of compare instructions carrying forward the carry and zero flag (CPED, CPEX, CPEY, CPES). This improves the capability to perform 32-bit compares.

While the existing architecture allows 8-bit, read-modify-write instructions, the S12XHZ extends this capability to 16-bit words and provides the following:

- NEGW (two's complement) and COMW (one's complement)
- DECW (decrement 16-bit) and INCW (increment 16-bit)
- RORW (rotate right) and ROLW (rotate left)
- LSRW (logical shift right) and LSLW (logical shift left)
- ASRW (arithmetic shift right) and ASLW (arithmetic shift left)
- CLRW (clear memory) and TSTW (test memory)

Addressing modes are the same as for their 8-bit counterparts. In general, these new 16-bit operations allow significantly faster manipulation of data compared to the S12HZ CPU.

Class 2 provides access to a new mode available on the S12XHZ MMC. This allows access to any 64 Kbyte page in global memory based on a new MCU register called GPAGE. The new instructions include all available addressing modes and concatenate the GPAGE register with the 16-bit address data. Global instructions are available for the following instructions:

- GLDAA (load accumulator A) and GLDAB (load accumulator B)
- GLDD (load accumulator D)
- GLDX (load X register) and GLDY (load Y register)
- GLDS (load stack pointer)
- GSTAA (store accumulator A) and GSTAB (store accumulator B)
- GSTD (store accumulator D)
- GSTX (store X register) and GSTY (store Y register)
- GSTS (store stack pointer)

The GPAGE register is 7 bits wide, so global memory runs from 0x00_0000 to 0x7F_FFFF, and each location is accessible with a single instruction from anywhere in a program (once the GPAGE register is configured for that 64 Kbyte page).

Class 3 allows more efficient use of semaphores, which are important for real time operating systems (RTOS) and for sharing resources between tasks on the CPU. The new instruction is BTAS (bit test and set). Since this is a single instruction, it cannot be interrupted; therefore, it is useful when requesting access to resources.

Software usually locks resources via a status bit in RAM—when the bit is set the resource is in use. On the S12HZ, you must take care that both tasks do not appear to have allocated the resource. This can occur if one task interrupts another immediately after a bit-test instruction; therefore, tasks typically disable interrupts while checking and allocating resources. The BTAS instruction removes this need, as it tests and sets the resource bit in a single instruction step. BTAS follows the same syntax and allows the same addressing modes as the BSET instruction, except that the test is based on the original data and not on the data written back.

A typical use for a BTAS instruction is shown in Figure 4.

```
BTAS $1020, #$20
BNE ResourceNotAvailable

<ResourceLocked>
```

Figure 4. Typical Use of BTAS Instruction

Class 4 is designed to improve the opportunity for compilers to use the memory-to-memory move instructions by allowing the use of all relevant S12XHZ addressing modes, and not only those fitting in a single postbyte xb. See the CPU Manual for more information on the newly added modes.

3 Interrupt Controller

The S12XHZ features a new interrupt controller module which provides you with seven interrupt priority levels (I-bit). The S12XHZ no longer requires the HPRIO (High Priority) interrupt function on the S12HZ, so it is completely removed. The XIRQ, SWI, BDM, unimplemented opcode, and system reset interrupts are available as before. In addition, the S12XHZ introduces a new interrupt vector to allow handling of ‘spurious’ interrupts which can occur if an interrupt source is removed before the interrupt is handled.

The S12XHZ also provides improved detection of invalid software operations which access areas of the MCU's memory which contain no resources. This enhancement applies in single-chip mode and causes a reset if the CPU accesses a memory location which does not address an on-chip memory or peripheral module. The reset vector fetched is the system reset at 0xFFFE.

On the S12HZ, the priority of any interrupt is determined by its position in the interrupt vector table. Vectors closer to the top of memory (0xFFFF) have a higher priority than those lower down. An exception to this is that the HPRIO register can be used to promote a single interrupt above its vector position; however, this only applies to the point at which interrupts are taken—once an interrupt is being serviced,

there is no way to automatically tell if any other pending interrupt has a higher priority than the current level.

On the S12XHZ, you can allocate each interrupt source to one of seven possible interrupt levels. You can change these levels at any time (although this is not recommended while interrupts are active). The seven active levels are complemented by an eighth level, which disables the interrupt.

An interrupt can be taken only if it is enabled, and the global mask (I-bit) is clear, and it is a higher level than the current working interrupt. The CPU is aware of and stacks the interrupt level at which it is working. For example, this means that the CPU cannot take a level 1 interrupt if it has not returned from processing a level 5 interrupt, even if the I-bit is clear. Conversely, the CPU cannot take a level 5 interrupt if it is working at level 1. Of course, the CPU will not take the level 5 interrupt if the I-bit is set. As with the S12HZ, the I-bit is set automatically on entry to an interrupt, so the code within each Interrupt Service Routine (ISR) must explicitly clear the I-bit using the CLI instruction, if nested interrupts are desired. Customers who do not want to use nested interrupts still benefit from the seven different priority levels, as the highest priority interrupt will always be selected from those pending.

When the CPU returns from an interrupt, part of its new functionality is to recover the interrupt level at which it was working before the interrupt was taken. This is stored in the upper byte of the CCR (see “Section 2.1, “Programmers Model”).

An additional feature of the interrupt module is the ability to specify the location of the interrupt vector table in memory. This is achieved by using the Interrupt Vector Base Register (IVBR). The IVBR specifies the top 8 bits of the vector table 16-bit address, and can be changed at any time. (Again. this is not recommended while interrupts are active.) The vector table always exists in the main 64 Kbyte map of the CPU. This ability to move the vector table allows you to have multiple vector tables for multiple mode operating systems, debugging systems, and bootloaders. The IVBR defaults to 0xFF out of reset for compatibility with the S12HZ. Finally, the interrupt module has the ability to route interrupts to and from the XGATE. You can allocate each interrupt to the CPU or XGATE and give each a priority. The XGATE will service the highest priority request first; however, it is not possible to interrupt an XGATE thread which is already running.

The interrupt module also handles interrupts created in the XGATE. You can raise these under software control and typically use them to request the CPU to perform some action after the XGATE has completed some function. All XGATE interrupts have the same level of priority; however, again, you can determine this level by programming the interrupt module. Figure 5 shows the architecture of the interrupt module.

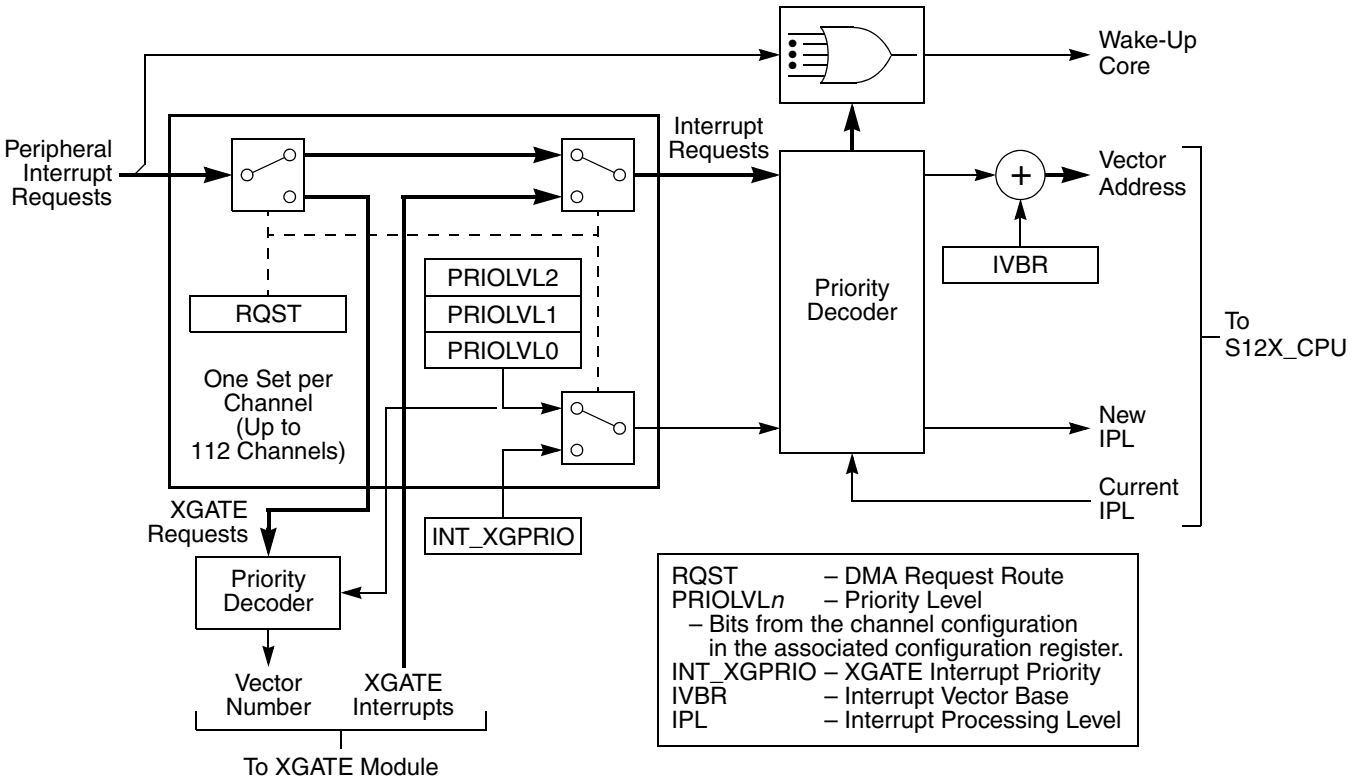

Figure 5. Interrupt Module

Figure 6 illustrates a typical profile of the interrupt processing levels possible when using the interrupt controller.

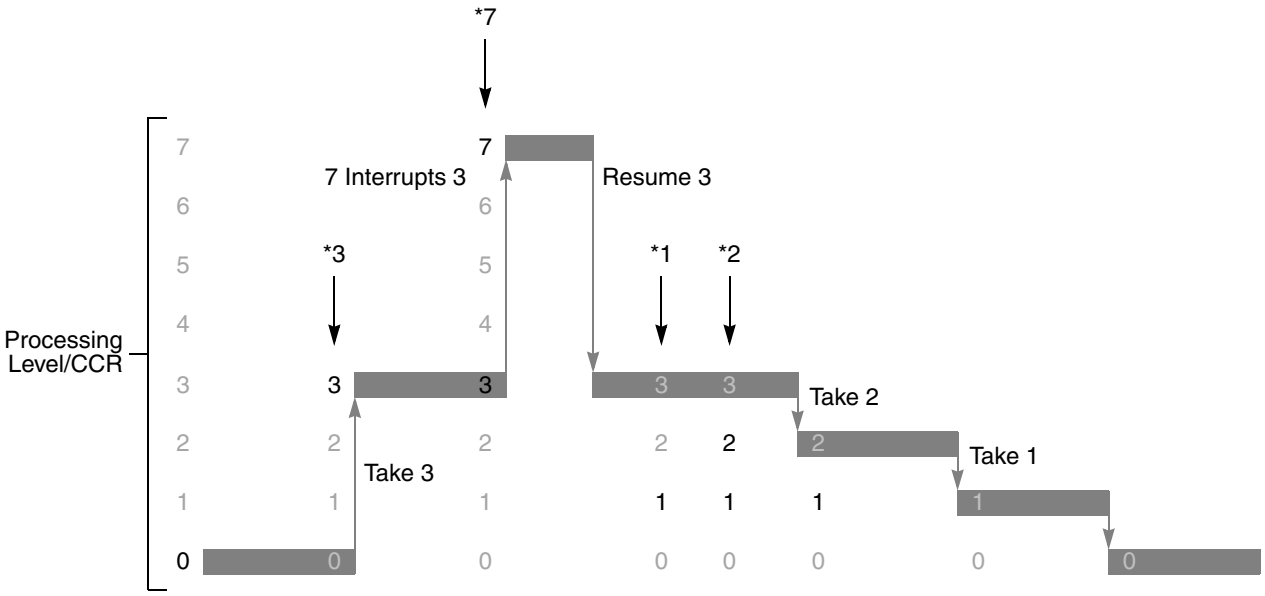

Figure 6. Interrupt Profile

In this example, the CPU is initially not executing an interrupt service routine. At the point marked by *3, the CPU recognizes an interrupt with a priority level of three and the CPU begins executing the Interrupt Service Routine (ISR). While executing the level three ISR, an interrupt with a level 7 priority occurs, and the CPU begins to execute the ISR at the point marked by *7. This ISR runs to completion and the level 3 ISR then resumes execution.

During the remaining execution time of level 3 ISR, a level 1 and level 2 interrupt, marked by *1 and *2 respectively, occur.

NOTE

Because these interrupts have a lower priority than the currently executing ISR, the CPU does not execute their ISRs; instead, the interrupts simply remain pending.

At the completion of level 3 ISR, the CPU executes level 2 ISR first because it has the highest priority of the two pending interrupts. Finally, at the completion of level 2 ISR, the CPU executes level 1 ISR and runs to completion.

NOTE

For interrupt nesting to occur as shown in this example, the software must clear the I-bit in the CPU's CCR at the start of each ISR.

You must take care not to set the interrupt level to 0 in the interrupt module. Doing so disables all interrupts from that interrupt source regardless of the settings of the peripheral's local interrupt enable bits.

4 XGATE

The XGATE is a completely new module on the S12XHZ. It is a highly integrated, but separately programmable coprocessor targeted at I/O management. The XGATE can access all of the peripheral modules and RAM, manipulate the data, and interact with the interrupt module, all independently of the CPU. The XGATE can also read part of the flash.

Specifically, the XGATE is a 16-bit RISC processor which runs up to double the CPU bus speed. Its faster speed allows it to access RAM at least once every CPU cycle. This allows a typical sustained performance of 70 MIPS. For those cycles where the CPU does not access RAM, then this performance is further increased.

The instruction set is optimized for handling data movement and logic operations. See Figure 7 for a block diagram of the programmer's model. There are eight 16-bit registers, of which R0 and R1 have special functions.

If you wish to maximize the performance of XGATE, store its software in RAM. XGATE software exists as threads which are initiated by a service request from the interrupt controller. Each request loads the start address of each thread and automatically initializes the R1 register to a value in the interrupt table. XGATE threads use R1 to point to a thread's variables and data which allows multiple peripherals to share a single instance of the code. For example, a simple XGATE thread may copy duty-cycle variables from RAM into the MC PWM registers. By writing this code carefully, it is possible to have the same thread handle all 12

MC PWMs on the 9S12XHZ512, thereby saving program space. It is equally possible to have different threads for each PWM.

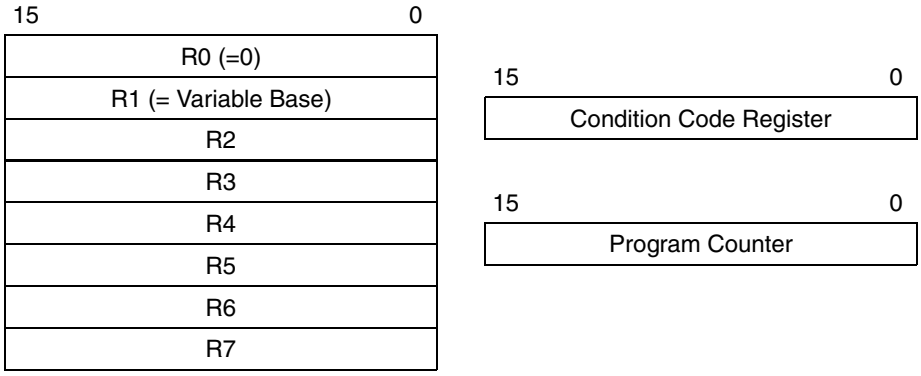

Figure 7. XGATE Programmer's Model

4.1 System Partitioning

The addition of the XGATE presents new possibilities and challenges for a software designer. Choice of tasks to allocate to a particular core requires careful consideration and since both cores effectively have simultaneous access to the RAM and peripherals, the system must be managed in such a manner that both cores do not attempt to utilize the same resource at the same time. Software designers can approach this challenge in a number of ways, and the S12XHZ architecture provides specific features to assist in this task. For example, the XGATE provides a set of eight dedicated semaphore flags which software designers can use to share resources. It also has eight software interrupts which allow the CPU to control the execution of the XGATE while the XGATE also has the ability to raise multiple interrupts on the CPU.

Additionally, since the XGATE code typically resides in RAM, the MCU provides a configurable protection system backward can be used to prevent accidental corruption of the XGATE or CPU code and variables. The protection mechanism permits the on-chip RAM to be partitioned into three areas: XGATE RAM, shared RAM, and CPU RAM.

4.2 XGATE RAM

XGATE RAM is typically used to store XGATE code. The CPU will copy XGATE code into RAM then set a protection bit. After the CPU completes the copy process. it enables the protection mechanism preventing it from performing any further writes to this area.

NOTE

The CPU can still read from this area.

4.3 Shared RAM

The shared RAM area allows full read/write access to both the XGATE and CPU. Typically, both cores use this area of RAM to exchange data.

4.4 CPU RAM

The CPU RAM area is defined for exclusive access by the CPU to prevent XGATE from unintentionally corrupting CPU data.

4.5 More Information on XGATE

XGATE is a formidable addition to the feature set of the S12XHZ family. Careful design will allow its significant performance to be released in a wide variety of applications. Freescale has created various application notes which focus on the use of XGATE as well as a library of software drivers to run on it. See www.freescale.com for more information.

TFT LCD driver is a useful application for XGATE in automotive cluster applications. See AN3493, *XGATE Library: Driving a TFT LCD Panel*, at www.freescale.com for details.

5 Memory Management Controller

The S12XHZ contains a new MMC which provides features not available on the S12HZ, and creates a standard memory map which is compatible for devices with large or small memory blocks.

The MMC defines locations in memory of all five memory blocks: registers, RAM, EEPROM, flash and external peripherals. In doing so, it fixes not only local 64 Kbytes accessible directly to the CPU, but also the full 8 Mbyte global memory space supported by the S12XHZ. For all S12XHZ devices, there is a direct correlation between the address of a memory block in the local map (0x0000 to 0xFFFF) and the global map (0x00_0000 to 0x7F_FFFF). In the global map, the most significant byte corresponds to the value in the GPAGE register.

5.1 RAM, EEPROM and Register Mapping

In creating the new MMC, some features found on the S12HZ were removed. This includes removing the capability to change the location of memory blocks. This is unlike the S12HZ where portions of RAM or registers can overlie the EEPROM and prevent its use; therefore, the S12XHZ no longer has the INITRM, INITRG and INITEE registers which define the starting location of the RAM, registers, and EEPROM, respectively. The functionality provided by these registers is implemented in an alternate way on the S12XHZ.

The INITRG register has no direct replacement function in the S12XHZ because registers are always fixed at address 0x0000; however, since in practice users only move the register block because they have a requirement to use the direct page area of the map, an alternative solution is available.

The S12HZ architecture fixes the direct page at 0x0000 to 0x00FF, and in that space the CPU can use the optimized direct-page addressing mode. Instead of this approach, the S12XHZ MMC can move the direct page location using a register called DIRECT. This register defines the upper 8 bits of the direct page; therefore, the direct page address is 0xZZ00 to 0xZZFF, where 0xZZ is the contents of the DIRECT register. Software can only write to the DIRECT register once.

The S12XHZ MMC fixes the EEPROM block in the local map at address 0x0800 to 0x0FFF. This 2 Kbyte block size is divided into two sections. The upper 1 Kbyte from 0x0C00 to 0x0FFF is always present in

the memory map. The lower 1 Kbyte acts as a page window in a similar fashion to that offered in the flash on the S12HZ and S12XHZ. Like the flash window, the EEPROM window has a control register to determine the 1 Kbytes of EEPROM which is visible. For the EEPROM, this register is called EPAGE (compare with PPAGE).On the 9S12XHZ512 there is a total of 4 Kbytes of EEPROM, so there is a single fixed 1 Kbyte and three further swappable pages of 1 Kbyte. The fixed 1 Kbyte space can also be made visible in this window if this is required.

The EEPROM is fixed in the global map at address 0x10_0000 to 0x13_FFFF giving a maximum EEPROM size of 256 Kbytes. On the 9S12XHZ512, the EEPROM exists at address 0x13_F000 to 0x13_FFFF.

The S12XHZ MMC fixes the RAM block in the local map at address 0x1000 to 0x3FFF. This 12 Kbyte block size is divided into two sections. The upper 8 Kbytes from 0x2000 to 0x3FFF are always present in the memory map. The lower 4 Kbyte space acts as a page window in similar fashion to the EEPROM and flash. RAM has a control register to determine the 4 Kbyte space which is to be visible. For the RAM, this register is called RPAGE.

On the 9S12XHZ512, there is a total of 32 Kbytes of RAM, so there is a single fixed 8 Kbyte block and six further swappable pages of 4 Kbytes. The fixed 8 Kbyte space can also be swapped into this window as two 4 Kbyte pages, if this is required. The RAM is fixed in the global map at address 0x00_1000 to 0x0F_FFFF, giving a maximum RAM size of just under 1 Mbytes. On the 9S12XHZ512, the RAM exists at address 0x0F_8000 to 0x0F_FFFF.

5.2 Flash Mapping

The S12XHZ has a more linear arrangement of flash pages than the S12HZ. This leads to a slight change in the selection of flash pages using the PPAGE register. No change to code is required; however, depending on use, the linker configuration may have to be slightly modified.

The flash is fixed in the global map at address 0x40_0000 to 0x7F_FFFF, giving a maximum flash size of 4 Mbytes. On the 9S12XHZ512, the flash exists at address 0x78_0000 to 0x7F_FFFF.

The PPAGE register selects a block of 16 Kbytes to be visible within a window from 0x8000 and 0xBFFF in the local map—this is unchanged from the S12HZ. The PPAGE value maps into the global map such that page 0x00 corresponds to the 16 Kbytes at the lower addresses in the global map (0x40_0000 to 0x40_3FFF). Page 0x01 corresponds to the next 16 Kbyte page (0x40_4000 to 0x40_7FFF), and so on until page 0xFF, which corresponds to the top 16 Kbytes in the global map (0x7F_C000 to 0x7F_FFFF).

In addition to the PPAGE window, and like the S12HZ, the S12XHZ has 32 Kbytes fixed to certain addresses in the global map. This means that these flash pages are always visible in the local map. In the local map, the 16 Kbytes from 0x4000 to 0x7FFF are fixed to the global map from 0x7F_4000 to 0x7F_7FFF, and the 16 Kbytes from 0xC000 to 0xFFFF are fixed to the global map from 0x7F_C000 to 0x7F_FFFF This linear arrangement is different to that found on the S12HZ, where the PPAGE register maps pages in a slightly different order.

The PPAGE register is limited to 6 bits on the S12HZ, and the second top two pages are swapped in order. For the top 64 Kbytes of flash, page 0x3F will map to the top page of the 64 Kbytes (0xpp_C000 to 0xpp_FFFF), as on the S12XHZ; however, the second bottom page (0xpp_4000 to 0xpp_7FFF) is

accessed at PPAGE 0x3E, unlike the S12XHZ where the value of 0xFD would be used. Typically then, the first page of flash used in the window is from page 0x3D, whereas on the S12XHZ this would be 0xFE.

In practice, this means that the flash page fixed at address 0x4000 to 0x7FFF on the S12HZ is mapped into the PPAGE window using a value of 0x3E, where the same fixed page on the S12XHZ is mapped using the value of 0xFD. Of course, since the flash pages are not accessible on the S12HZ through any other method, the actual topology of the flash is irrelevant when selecting pages.

5.3 Global Memory Mapping

As discussed in previous sections, the MMC maps all on-chip resources into a unified 8 Mbyte global, linear address space. Software can use the new global load and store instructions in conjunction with the GPAGE register to access on-chip flash, RAM, EEPROM, I/O registers and even external memory in a unified manner. The primary benefit of this alternate address space is that unlike the S12HZ, the S12XHZ allows code in any page of flash to access data in any other page in flash, removing the need for page swaps via intermediate memory. In addition, software can access large data tables (<= 64K) while executing from anywhere in the flash, RAM or EEPROM.

See application note AN2734, *HCS12X Family Memory Organization,* for more information on the MMC.

6 External Bus Interface

The External Bus Interface (EBI) on the S12XHZ is completely different from the S12HZ and is only available on the 144-pin package. This different external bus allows best use of the increase in bus speed and provides a glueless interface to asynchronous memories, display controllers, serial bus controllers, etc. The S12XHZ bus is non-multiplexed and features a 23-bit address bus, a 16-bit data bus, four chip selects, read and write enables, an external wait control input line, and an enable clock. Many applications can use the expanded bus with no additional glue logic between the MCU and the external peripheral.

On the S12XHZ expanded bus, the E-clock is a free-running clock. Software can configure the bus to enable or disable the E-clock at the internal bus speed or divided by 2, 3 or 4. The minimum number of cycles for each access on the EBI is two cycles. The S12XHZ replaces the R/W signal functionality on the S12HZ by RE (read enable) and WE (write enable) signals.

Like the S12HZ, the S12XHZ can accommodate slow peripherals on the EBI by the insertion of up to three wait states on the interface timing. This is an acceptable solution if the peripheral has a fixed access time; however, there are peripherals, such as graphics controllers, which have varying access times. To support connection to such peripherals, the expanded bus also features the EWAIT signal. A peripheral connected to the EWAIT input can suspend CPU operation until it is ready to accept or write the required data. The inserted wait states continue indefinitely until the EWAIT signal is released. The EWAIT signal does not affect the operation of the XGATE.

Since the EBI on the S12XHZ is a non-multiplexed bus, it uses many more General Purpose I/O pins than the S12HZ; however, it is possible to reduce the number of pins if the full address and data busses are not required. When used in expanded mode, the EBI on the S12HZ uses Port A, Port B, Port E and Port K. On the S12XHZ, the EBI uses a maximum of Port A, Port B, Port C, Port D, Port E and Port K.

If the EBI is connected to an 8-bit peripheral, then you can reduce the data bus to 8 bits which allows Port C to become available as a GPIO port. If peripherals connected to the EBIs do not use the full external 4 MByte address map, then it is possible to reduce the number of address lines on the external address bus which allow Port K, Port B and Port A to be used as GPIO. For more information on the usage and configuration of the expanded bus, please refer to AN2708, *An Introduction to the External Bus Interface on the HCS12X*.

7 Debug Module

The debug module on the S12XHZ supports the new features of the processor including the full 8 Mbyte memory map and XGATE. It is possible to capture software flow on both cores simultaneously or each core individually. The extended trace buffer and trigger configurations on the S12XHZ allow more complex debugging possibilities than on the S12HZ.

8 Oscillator

The S12XHZ features an oscillator module which is different from that provided on the S12HZ. The oscillator module provides an amplitude controlled Pierce configuration. The S12HZ features a choice of an amplitude controlled Colpitts or a full amplitude Pierce configuration.

The Pierce configuration improves on the current and EMC performance of the Pierce on the S12HZ and approaches the current supply requirements of the Colpitts.

As with all oscillator designs, the choice of external components, and the layout of the PCB is critical. It is strongly advised that you carefully consult the advice given in the data sheet chapter on the Oscillator.

9 Enhanced Capture Timer

The TIM module on the S12HZ is replaced by the ECT on the S12XHZ. This contains a number of new features. In addition, the ECT on the S12XHZ contains enhancements to the prescalers, glitch-free initialization of output compare (OC) pins, and a more deterministic method of setting OC flags.

The ECT timer prescaler is fully backward-compatible with the S12HZ but now allows more precise timing measurement and creation. The new prescaler applies to the main counter as well as the modulus down counter and the input capture filter delay counter. The prescaler on the S12HZ allows steps of 1, 2, 4, 8, 16, 32, 64, and 128. The additional prescalers on the S12XHZ allow steps of 1, 2, 3, 4, 5, and 6, up to 256. You can select the prescaler in use under software control.

The ECT on the S12XHZ has an additional Output Compare Pin Disconnect (OCPD) register, which can support glitch-free initialization of the output state of any OC channel. On the S12HZ, the OC configuration, OMx = 0, OLx = 0, is described as “Timer disconnected from output pin logic.” This is a typical configuration where the application uses the OC interrupt only for internal interrupt timing. In this case, control of the associated port pin automatically reverts to the PIM port control registers.

On the S12XHZ, the OC configuration, OMx = 0, OLx = 0, is more specifically described as “No output compare action on the timer output signal.” The internal interrupt behavior is the same as on the S12HZ,

but the OCPD register explicitly controls the connection of either the output/compare (OC) channel logic or the input/output (I/O) port control logic to the port pin.

The reset state of the OCPD register connects the OC logic to the pin when a channel is configured for OC. For compatibility with S12HZ user software must set the relevant OCPD bit during configuration if it requires GPIO functionality while using the OC channel as an internal interrupt timer.

On the S12XHZ family the ECT sets the OC flags (CnF) under different conditions in the timer cycle. On the S12HZ the TIM sets a CnF flag when the main timer (TCNT) value matches the value in the OC register (IOCn). This means that if software writes the IOCn register to the current value of TCNT then the CnF flag is set. On the S12XHZ the ECT sets the CnF flag only when TCNT increments to the value in the IOCn register. This means that software must write the desired IOCn value before TCNT reaches that count.

10 Real Time Interrupt

Like the ECT, the RTI timer also features an enhanced prescaler. Also like the ECT, it is fully backward-compatible.

The S12XHZ RTI features an option to use a decimal rather than a binary prescaler. This is useful where, for example, a 5 µs tick is required from a 4 MHz oscillator. Using a binary divide, the closest tick value is 5.12 µs (= 4 MHz/(5*4096)), whereas the decimal prescaler can achieve exactly 5 µs (= 4 MHz/(2*10000)).

The decimal RTI prescaler also allows divisions up to 15 times greater than the binary prescaler.

11 COP—Watchdog

The COP rate and type (window or non-window COP) are loaded from the flash during the reset procedure; therefore, the COP can be enabled out of reset without CPU intervention for safety critical systems. Full backward compatibility to the existing COP is maintained when the reserved flash byte is erased.

12 Periodic Interrupt Timer

The PIT is a new module on the S12XHZ. You can use the PIT to create interrupts or trigger the ADC peripheral at very precise time intervals. The PIT operates independently of the ECT and allows 24-bit count depths. This provides extremely large count values for long timeout periods or shorter periods with high resolution. The PIT is a completely internal module which does not take up any external pins.

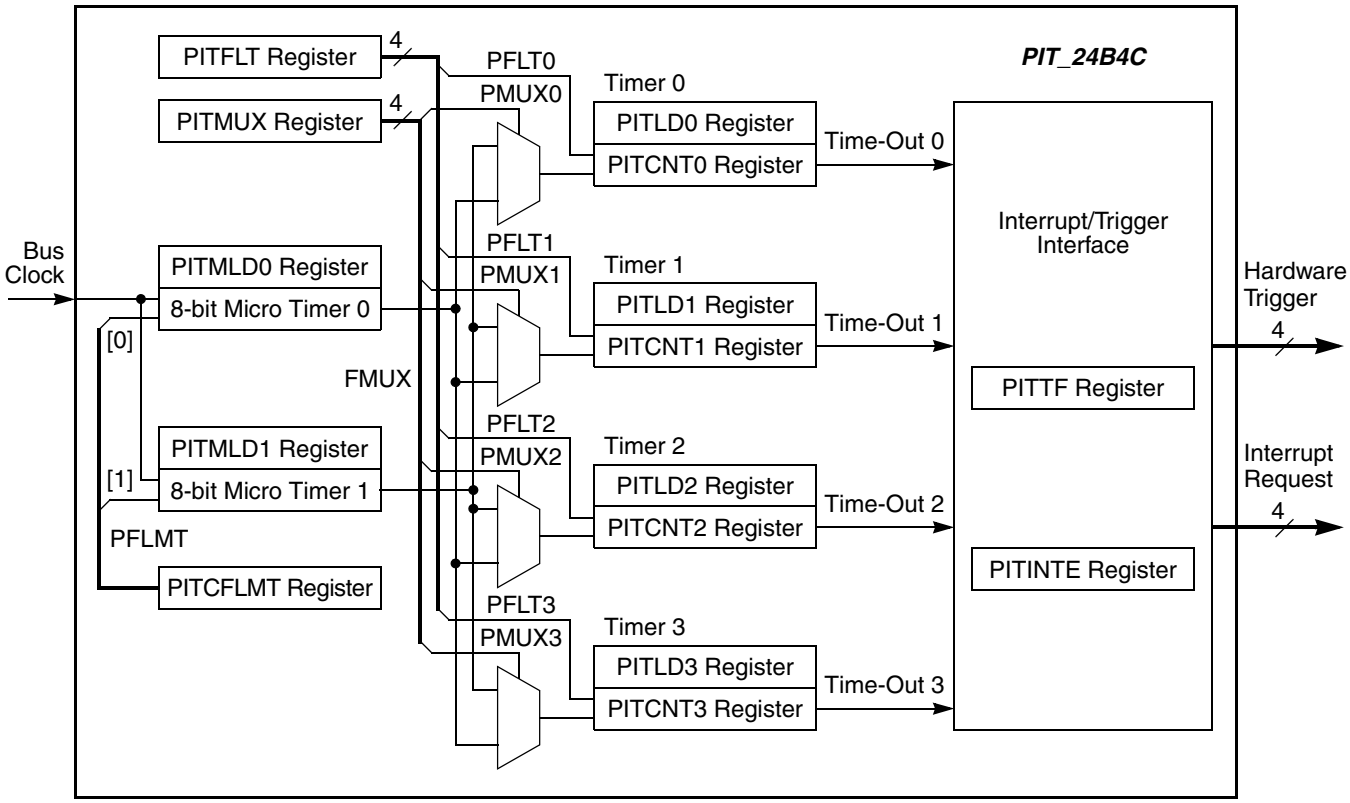

Figure 8. Periodic Interrupt Timer

As can be seen in Figure 8, the PIT consists of two stages with four 16-bit modulus down-counters and two 8-bit modulus down-counters. The output of each 8-bit counter can drive any of the 16-bit counters. This allows for a wide variety of timeout periods which may or may not have an integer relation to each other. For example, if an application has the need to execute several tasks at different time intervals, then interrupts of 1 µs, 4 µs and 10 µs are possible, in addition to a regular 823 µs tick (to activate an A/D conversion, for example). All of these timeouts are possible (at 40 MHz bus clock), with no further intervention by the CPU. Equally, the PIT could generate interrupts of 21.75 µs, 3.93 µs, 29.1 µs and 0.419 s (at 40 MHz bus clock).

13 Analog to Digital Converter

On the S12XHZ, you can trigger ATD conversions from additional sources. The new triggers are in addition to the conversion modes on the S12HZ. On the 9S12XHZ512, you can trigger the ATD from PWM channel 1 or 3, PIT timer channel 0 or 1, or any of the external ATD pins.

NOTE

Due to differences in the internal design layout, the ATD channels located on Port L require an extra initialization for equivalent operation. The difference means that default pulldown resistors on Port L affect the accuracy of the ATD more significantly than on the S12HZ. For best operation on both S12HZ and S12XHZ, we recommend that you disable the pulldown resistors on those pins which are in use as ATD channels.

14 SCI

At reset, SCI1 connects to Port P0 and Port P2 which is backward-compatible with the S12HZ. It can optionally connect to Port S[2:3].

The SCI modules feature new hardware support for the following: IrDA protocol, LIN detection of break signal and bus signal collisions, wake up interrupts on active edges, and alternate pin polarity on the receive and transmit pins.

15 IIC

The S12XHZ features an additional IIC and relabels the IIC on S12HZ as IIC0. At reset, IIC0 connects to Port P[4:5] which is compatible with the S12HZ. It can also be connected to Port T[4:5]. At reset IIC1 connects to Port P[6:7]. It can also be connected to Port T[6:7].

On the S12XHZ, the IIC modules have two additional bits to support new General Call Address and 10-bit Addressing features, which are controlled via a new control register IBCR2 located at module base +0x0005. This is a reserved register on S12HZ.

The new features default to disabled and the default IIC is fully compatible with S12HZ.

16 msCAN

The µsCAN module provides a new manual control for recovery from bus-off conditions. In the manual mode, the Bus-off recovery is under software control not earlier than after 128 * 11 received bits. The S12HZ automatic recovery after 128 * 11 received bits is still available, and is the default configuration.

17 Serial Peripheral Interface

The SPI is unchanged from the S12HZ to the S12XHZ.

18 Liquid Crystal Display Module

The LCD module is unchanged from the S12HZ to the S12XHZ.

19 Background Debug Module

The BDM module on the S12XHZ includes changes to account for the new MMC and also to take advantage automatically of the blank check capability of the flash and EEPROM hardware interfaces.

20 Low Power and STOP Mode Support

The S12XHZ features new support for those applications where low power is important and which make frequent use of the STOP mode.

Many applications require very low power usage over an extended period, while not losing the ability of the processor to respond to external events. STOP mode is very often used for this type of application, and the S12XHZ provides two new features to reduce power consumption to a minimum as follows:

- Fast response to external events by starting external oscillator only as required
- Low-power internal interval timer to allow regular timed wake-ups

The S12XHZ provides a fast response from STOP mode by allowing you to run code from the internal VCO (in the PLL module) very soon after an event is detected. Since the VCO begins oscillating very quickly, this avoids the delay in starting the external crystal circuit. In many cases, the processor can examine the event and a full recovery can be avoided if not required. If a recovery is required, then the user code simply restarts the external oscillator and begins running code as normal.

Where a regular check on external activity is required, the S12XHZ can make use of a low-power internal RC oscillator called the autonomous periodical interrupt (API) which is inside the Voltage Regulator. This has a programmable interrupt period of between 0.2 µs to 819 µs. When enabled, the API oscillator wakes the CPU at the specified interval and allows user code to examine the state of the application. Again you can decide to start up using the VCO or the external crystal.

For more detail on the approaches available on the S12XHZ refer to application note AN3289, *Low-Power Techniques for the S12X Family.*

21 Motor Control Ports

The S12XHZ is available in an additional 144-pin package which allows access to an additional MC port. Port W has exactly the same functionality as Ports U and V including higher current port pins, and additional MC modules and SSD modules.

22 General Purpose I/O (GPIO)

The GPIO is backward-compatible from the S12HZ to S12XHZ. There are some additional control registers to enhance performance on the S12XHZ as well as some new GPIO registers.

New registers are as follows:

- Port C – Port C, DDRC, pullup control in PUCR
- Port D – Port D, DDRD, pullup control in PUCR
- Port W – Port W, Input register, DDRW, slew rate control, pull device enable and polarity select

- Slew rate control registers for Port L, Port M, Port P, Port S, and Port T
- Port T includes a wired-or control register which affects bits 4 to 7. This register also contains the module routing control bits which can redirect the ports used by SCI1, IIC0 and IIC1.

Additional bits are added to the Port K, Port M, Port P and Port S.

The new slew-rate control registers provide an extra level of I/O configurability especially when combined with the existing reduce-drive control registers.

23 Pinout Compatibility

The S12HZ is available in 80-pin and 112-pin packages while the S12XHZ512 is available in 112-pin and 144-pin packages.

The 112-pin package is pinout-compatible, but the following differences must be noted:

- The S12XHZ contains additional GPIO ports which are not bonded-out in this package. It is important that these unbonded GPIO pins are set either to outputs or inputs with pullup or pulldown devices enabled. The unbonded ports are Port C, Port D, Port W, Port K[4:6], Port M [1], Port P[6:7], and Port S[2:3].
- On the S12XHZ, there are pull devices connected to the following pins: RESET (with an internal pullup) and TEST (with an internal pulldown).
- As already noted, the S12XHZ features either a full-swing Pierce or a loop-controlled Pierce oscillator configuration. Depending on the configuration in use on the S12HZ, it may be necessary to change the layout of crystal or resonator components to match the oscillator in use.

24 Register Map Compatibility

The number of new and enhanced features on the S12XHZ, and the removal of some redundant features of the S12HZ, mean that the register map is not 100% compatible between MCUs. You must ensure that any direct register accesses are to the correct address. This is done relatively easy by updating the symbolic definition of register names with the correct address and nomenclature for the S12XHZ.

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: AN3510
Rev. 0
8/2007

freescale™
semiconductor