1 TMS320VC5421 Features

- 200-MIPS Dual-Core DSP Consisting of Two Independent Subsystems
- Each Core Has an Advanced Multibus Architecture With Three Separate 16-Bit Data Memory Buses and One Program Bus
- 40-Bit Arithmetic Logic Unit (ALU) Including a 40-Bit Barrel-Shifter and Two 40-Bit Accumulators Per Core
- Each Core Has a 17-Bit × 17-Bit Parallel Multiplier Coupled to a 40-Bit Adder for Non-Pipelined Single-Cycle Multiply/ Accumulate (MAC) Operations
- Each Core Has a Compare, Select, and Store Unit (CSSU) for the Add/Compare Selection of the Viterbi Operator
- Each Core Has an Exponent Encoder to Compute an Exponent Value of a 40-Bit Accumulator Value in a Single Cycle
- Each Core Has Two Address Generators With Eight Auxiliary Registers and Two Auxiliary Register Arithmetic Units (ARAUs)
- 16-Bit Data Bus With Data Bus Holder Feature
- 512K-Word × 16-Bit Extended Program Address Space
- Total of 256K-Word × 16-Bit Dual- and Single-Access On-Chip RAM (128K-Word x 16-Bit Two-Way Shared Memory)
- Single-Instruction Repeat and Block-Repeat Operations
- Instructions With 32-Bit-Long Word Operands
- Instructions With Two or Three Operand Reads
- Fast Return From Interrupts
- Arithmetic Instructions With Parallel Store and Parallel Load
- Conditional Store Instructions
- Output Control of CLKOUT
- Output Control of TOUT
- Power Consumption Control With IDLE1, IDLE2, and IDLE3 Instructions
- Dual 1.8-V (Core) and 3.3-V (I/O) Power Supplies for Low-Power, Fast Operations
- 10-ns Single-Cycle Fixed-Point Instruction
- Interprocessor Communication via Two Internal 8-Element FIFOs
- Twelve Channels of Direct Memory Access (DMA) for Data Transfers With No CPU Loading (Six Channels Per Subsystem With External Access)
- Six Multichannel Buffered Serial Ports (McBSPs) With 128-Channel Selection Capability (Three McBSPs per Subsystem)
- 16-Bit Host-Port Interface (HPI) Multiplexed With External Memory Interface Pins
- Software-Programmable Phase-Locked Loop (APLL) Provides Several Clocking Options (Requires External Oscillator)
- On-Chip Scan-Based Emulation Logic, IEEE Standard 1149-1† (JTAG) Boundary-Scan Logic
- Two Software-Programmable Timers (One Per Subsystem)
- Software-Programmable Wait-State Generator (14 Wait States Maximum)
- Provided in 144-pin MicroStar BGA™ Ball Grid Array (GGU Suffix) and 144-pin Low-Profile Quad Flatpack (LQFP) (PGE Suffix) Packages

MicroStar BGA is a trademark of Texas Instruments.

‡ IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

2 Introduction

This section describes the main features, gives a brief functional overview of the TMS320VC5421, lists the pin assignments, and provides a signal description table. This data manual also provides a detailed description section, electrical specifications, parameter measurement information, and mechanical data about the available packaging.

NOTE: This data manual is designed to be used in conjunction with the *TMS320C54x™ DSP Functional Overview* (literature number SPRU307).

2.1 Description

The TMS320VC5421 fixed-point digital signal processor (DSP) is a dual-core solution running at 200-MIPS performance. The 5421 consists of two DSP subsystems capable of core-to-core communications and a 128K-word zero-wait-state on-chip program memory shared by the two DSP subsystems. Each subsystem consists of one 54x DSP core, 32K-word program/data DARAM, 32K-word data SARAM, 2K-word ROM, three multichannel serial interfaces, xDMA logic, one timer, one APLL, and other miscellaneous circuitry.

The 5421 also contains a host-port interface (HPI) that allows the 5421 to be viewed as a memory-mapped peripheral to a host processor. The 5421 is pin-compatible with the TMS320VC5420.

Each subsystem has its separate program and data spaces, allowing simultaneous accesses to program instructions and data. Two read operations and one write operation can be performed in one cycle. Instructions with parallel store and application-specific instructions can fully utilize this architecture. Furthermore, data can be transferred between program and data spaces. Such parallelism supports a powerful set of arithmetic, logic, and bit-manipulation operations that can all be performed in a single machine cycle. The 5421 includes the control mechanisms to manage interrupts, repeated operations, and function calls. In addition, the 5421 has 128K words of on-chip program memory that can be shared between the two subsystems.

The 5421 is intended as a high-performance, low-cost, high-density DSP for remote data access or voice-over IP subsystems. It is designed to maintain the current modem architecture with minimal hardware and software impacts, thus maximizing reuse of existing modem technologies and development efforts.

2.2 Migration From the 5420 to the 5421

Customers migrating from the 5420 to the 5421 need to take into account the following:

- The memory structure of the 5421 has been changed to incorporate 128K x 16-bit words of two-way shared memory.
- The DMA of the 5421 has been enhanced to provide access to external, as well as internal memory.
- The HPI and DMA memory maps have been changed to incorporate the new memory 5421.
- 2K x 16-bit words of ROM have been added to the 5421 for bootloading purposes only.
- The VCO pin on the 5420 has been replaced with the $\overline{\text{HOLDA}}$ pin on the 5421 and the $\overline{\text{HOLD}}$ pin was added to the 5421 at a previously unused pin location.
- The McBSPs have been updated with a new mode that allows 128-channel selection capability.
- McBSP CLKX/R pins can be used as inputs to internal clock rate generator for CLKS-like function without the penalty of extra pins.
- The SELA/B pin on 5421 is changed to type I/O/Z for added functionality.

NOTE:

For additional information, see *TMS320VC5420 to TMS320VC5421 DSP Migration* (literature number SPRA621).

TMS320C54x is a trademark of Texas Instruments.

3 Functional Overview

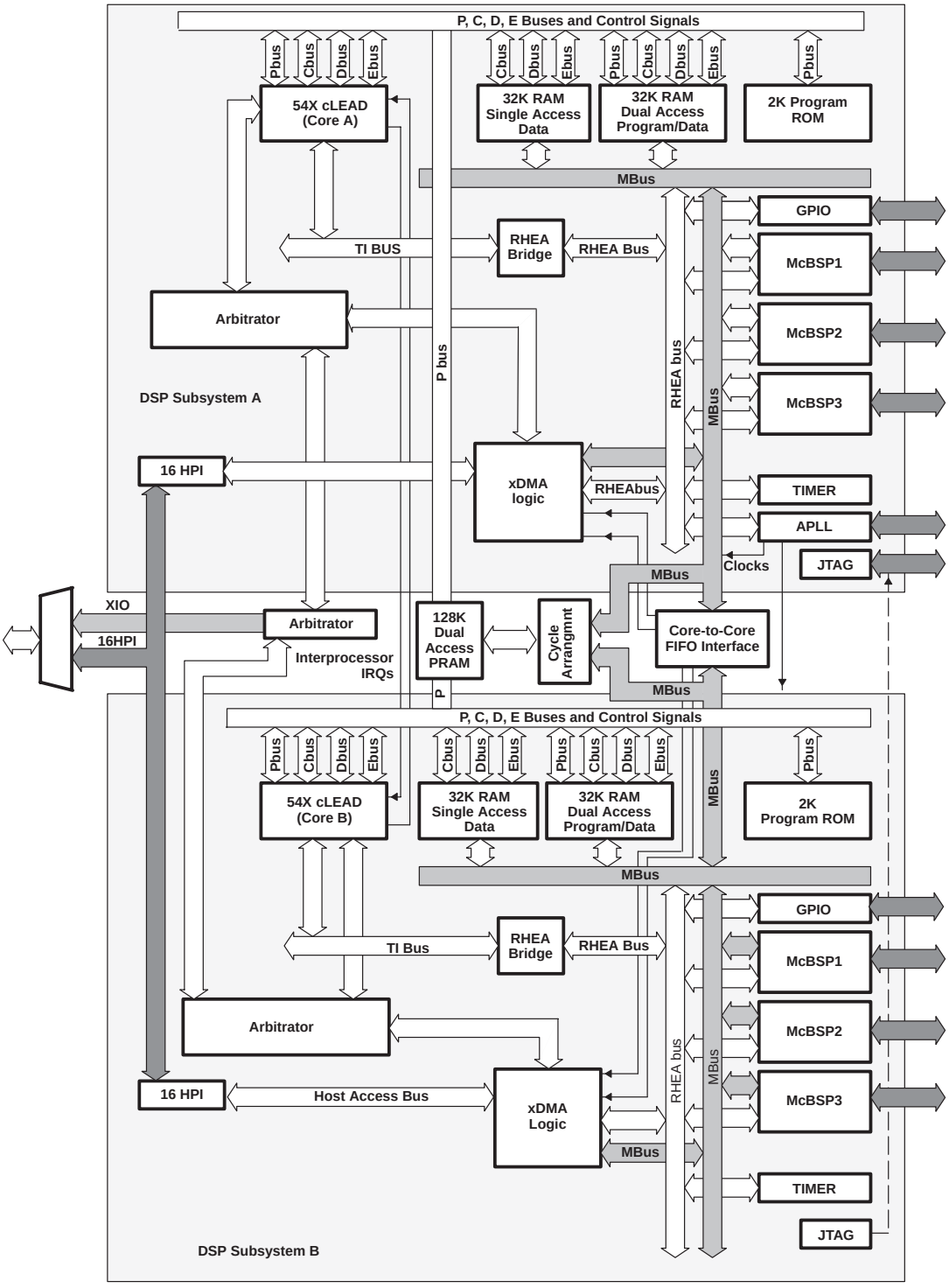
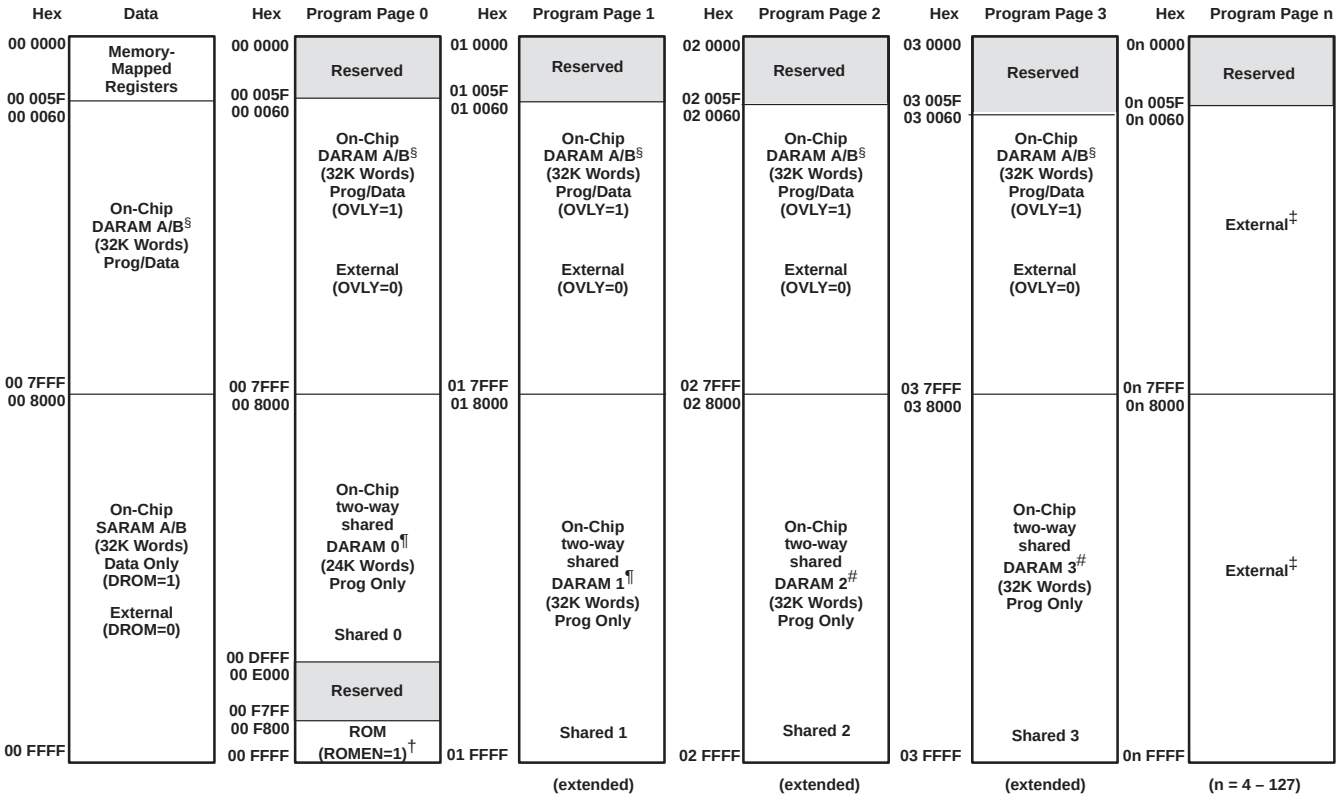


Figure 3–1. TMS320VC5421 Functional Block Diagram

3.1 Memory

Each 5421 DSP subsystem maintains the peripheral register memory map and interrupt location/priorities of the standard 5420. Figure 3–2 shows the size of the required memory blocks and their link map within the program and data space of the cLEAD core. The total on-chip memory for the 5421 devices is 256K-word data/program.



† ROM enabled after reset.

‡ When CPU PMST register bit MP/$\overline{MC}$=0 and an address is generated outside the on-chip memory bound or the address reach, i.e., XPC > 3h, access is always external, if XIO = 1. Pages 8–127 are mapped over pages 4–7. When XIO = 1 and MP/$\overline{MC}$ = 1, program pages 0, 1, 2, and 3 are external. Pages 4–127 are mapped over pages 0–3.

§ On-chip DARAM A and SARAM A are for subsystem A. Likewise, on-chip DARAM B and SARAM B are for subsystem B.

¶ On-chip DRAM 0 and DRAM 1 are owned by subsystem A and shared with subsystem B.

On-chip DRAM 2 and DRAM 3 are owned by subsystem B and shared with subsystem A.

NOTES: A. Clearing the ROMEN bit (GPIO[7]) enables an 8K-word block (0E000h – 0FFFFh) of DARAM .

B. All external accesses require the XIO pin to be high.

C. CPU I/O space is a single page of 64K words. Access is always external.

D. All internal memory is divided into 8K blocks.

Figure 3–2. Memory Map Relative to CPU Subsystems A and B

3.1.1 On-Chip Dual-Access RAM (DARAM)

The 5421 subsystems A and B each have 32K 16-bit words of on-chip DARAM (4 blocks of 8K words). Each of these DARAM blocks can be accessed twice per machine cycle. This memory is intended primarily to store data values; however, it can be used to store program as well. At reset, the DARAM is mapped into data memory space. The DARAM can be mapped into program/data memory space by setting the OVLY bit in the processor-mode status (PMST) register of the 54x CPU in each DSP subsystem.

3.1.2 On-Chip Single-Access RAM (SARAM)

The 5421 subsystems A and B each have 32K 16-bit words of on-chip SARAM (4 blocks of 8K words). Each of these SARAM blocks can be accessed once per machine cycle. This memory is intended to store data values only. At reset, the SARAM is disabled. The SARAM can be enabled in data memory space by setting the DROM bit in the PMST register.

3.1.3 On-Chip Two-Way Shared RAM (DARAM)

The 5421 has 128K 16-bit words of on-chip DARAM (16 blocks of 8K words) that is shared between the two DSP subsystems. This memory is intended to store program only. Each subsystem is able to make one instruction fetch from any location in two-way shared memory each cycle. Neither subsystem CPU can write to the two-way shared memory as only the DMA can write to two-way shared memory.

3.1.4 On-Chip Boot ROM

The 5421 subsystems A and B each have 2K 16-bit words of on-chip ROM. This ROM is used for bootloading functions only. Enabling the ROM maps out one 8K-word block of the shared program memory. The ROM can be disabled by clearing bit 7 (ROMEN) of the general-purpose I/O (GPIO) register. Table 3–1 shows the XIO/ROMEN modes. The ROM is enabled or disabled at reset for each subsystem depending on the state of the GPIO0 pin for that subsystem.

Table 3–1. XIO/ROMEN Modes

XIO	ROMEN/GPIO0	MODE
0	x	Fetch internal from RAM
1	0	Fetch external
1	1	ROM enabled

3.1.5 Extended Program Memory

The program memory space on the 5421 device addresses up to 512K 16-bit words. The 5421 device uses a paged extended memory scheme in program space to allow access of up to 512K of program memory . This extended program memory (each subsystem) is organized into eight pages (0–7), pages 0–3 are internal, pages 4–7 are external, each 64K in length. (Pages 8–127 as defined by the program counter extension register (XPC) are aliases for pages 4–7.) Access to the extended program memory is similar to the 5420. To implement the extended program memory scheme, the 5421 device includes the following feature:

- Two 54x instructions are extended to use the additional two bits in the 5421 device.
  - READA – Read program memory addressed by accumulator A and store in data memory
  - WRITA – Write data to program memory addressed by accumulator A
    (Writes not allowed for CPUs to shared program memory)

3.1.6 Program Memory

The program memory is accessible on multiple pages, depending on the XPC value. Within these pages, memory is accessible, depending on the address range.

- Access in the lower 32K of each page is dependent on the state of OVLY.
  - OVLY = 0 – Program memory is accessed externally for all values of XPC.
  - OVLY = 1 – Program memory is accessed from local data/program DARAM for all values of XPC.
- Access in the upper 32K of each page is dependent on the state of MP/$\overline{MC}$ and the value of XPC.
  - MP/$\overline{MC}$ = 0 – Program memory is accessed internally from two-way shared DARAM for XPC = 0–3. Program memory is accessed externally for XPC = 4–127.
  - MP/$\overline{MC}$ = 1 – Program memory is accessed externally for all values of XPC.

3.1.7 Data Memory

The data memory space is a single page of 64K. Access is dependent on the address range. Access in the lower 32K of data memory is always from local DARAM.

Access in the upper 32K of data memory is dependent on the state of DROM.

- DROM = 0 – Data memory is accessed externally
- DROM = 1 – Data memory is accessed internally from local SARAM

3.1.8 I/O Memory

The I/O space is a single page of 64K. Access is always external.

When XIO = 0 and an access to external memory is attempted, any write is ignored and any read is an unknown value.

3.2 Multicore Reset Signals

The 5421 device includes three reset signals: $\overline{A\_RS}$, $\overline{B\_RS}$, and $\overline{HPIRS}$. The $\overline{A\_RS}$ and $\overline{B\_RS}$ pins function as the CPU reset signal for subsystem A and subsystem B, respectively. These signals reset the state of the CPU registers and upon release, initiate the reset function. Additionally, the $\overline{A\_RS}$ signal resets the on-chip PLL and initializes the CLKMD register to bypass mode.

The HPI reset signal ($\overline{HPIRS}$) places the HPI peripheral into a reset state. It is necessary to wait three clock cycles after the rising edge of $\overline{HPIRS}$ before performing an HPI access. The $\overline{HPIRS}$ signal also resets the PLL by turning off the PLL and initializing the CLKMD register to bypass mode.

3.3 Bootloader

The on-chip bootloader is used to automatically transfer user code from an external source to anywhere in program memory after reset. The XIO pin is sampled during a hardware reset and the results indicate the operating mode as shown in Table 3–2.

Table 3–2. Bootloader Operating Modes

XIO	AFTER RESET
0	HPI mode, bootload is controlled by host. The external host holds the 5421 in reset while it loads the on-chip memory of one or both subsystems as determined by the SELA/B pin. The host can release the 5421 from reset by either of the following methods: 1. If the $\overline{A\_RS}$/$\overline{B\_RS}$ pins are held low while $\overline{HPIRS}$ transitions from low to high, the subsystem cores reset will be controlled by the $\overline{A\_RS}$/$\overline{B\_RS}$ pins. When the host has finished downloading code, it drives $\overline{A\_RS}$/$\overline{B\_RS}$ high to release the cores from reset. 2. If the $\overline{A\_RS}$/$\overline{B\_RS}$ pins are held high while $\overline{HPIRS}$ transitions from low to high, the subsystems stay in reset until a HPI data write to address 0x2F occurs. This means the host can download code to subsystem A and then release core A from reset by writing any data to core A address 0x2F via the HPI. The host can then repeat the sequence for core B. This mode allows the host to control the 5421 reset without additional hardware.
1	XIO mode. ROM is mapped in, if ROMEN pin = 1 during reset.

The 5421 bootloader provides the following options for the source of code to download:

- Parallel from 8-bit or 16-bit-wide EPROM
- Serial boot from McBSPs, 8-bit mode

GPIO register bit 7 (ROMEN) is used to enable/disable the ROM after reset. The ROMEN bit reflects the status of the ROMEN/GPIO0 pin for each core. ROMEN = 1 indicates that the ROM and the 8K-word program memory block (00 E000h–00 FFFFh) are not available for a CPU write. When ROMEN = 0, this 8K-word program memory is available and the ROM is disabled.

A combination of interrupt flags and the bit values of an external memory location determine the selection of the various boot options.

3.4 External Interface (XIO)

The external interface (XIO) supports the 5421 master boot modes and other external accesses. Its features include:

- Multiplexed with the HPI pins
- Selection of XIO or HPI mode is determined by a dedicated pin (XIO)
- Provides 512K words of external program space, 64K words of external data space, and 64K words of external I/O space.
- Different boot modes are selectable by the XIO, HMODE, and $\overline{\text{A\_RS}}$/$\overline{\text{B\_RS}}$ pins.
- After reset, the control register bit ROMEN is always preset to 1.

While XIO = 0 during reset, host HPI mode is on, the host sees all RAM, and ROM is disabled. A host write to 002Fh releases the CPUs from reset; the 002Fh write by the host clears the ROMEN bit in the GPIO register.

While XIO = 1 and ROMEN = 1 during reset, the CPU starts from ROM (0FF80h) to do boot selection. After branching to non-ROM area, the code changes the ROMEN bit to enable the RAM area occupied by ROM. While XIO = 1 and ROMEN = 0 during reset, the CPU starts from external (0FF80h) to do boot selection.

Table 3–3 provides a complete description of HMODE, SELA/B, and XIO pin functionality.

Table 3–3. XIO/HPI Modes

HMODE	SELA/B	HPI MODES (XIO = 0)	XIO MODES (XIO = 1)
0	0	HPI muxed address/data subsystem A slave to host	SELA/B pin is multiplexed as PPA18 output.
0	1	HPI muxed address/data subsystem B slave to host	SELA/B pin is multiplexed as PPA18 output.
1	0	HPI non-muxed address/data subsystem A slave to host	SELA/B pin is multiplexed as PPA18 output.
1	1	HPI non-muxed address/data subsystem B slave to host	SELA/B pin is multiplexed as PPA18 output.

3.5 On-Chip Peripherals

All the 54x devices have the same CPU structure; however, they have different on-chip peripherals connected to their CPUs. The on-chip peripheral options provided are:

- Software-programmable wait-state generator
- Programmable bank-switching
- Parallel I/O ports
- Multichannel buffered serial ports (McBSPs)
- A hardware timer
- A software-programmable clock generator using a phase-locked loop (PLL)

3.5.1 Software-Programmable Wait-State Generators

The software-programmable wait-state generator can be used to extend external bus cycles up to fourteen machine cycles to interface with slower off-chip memory and I/O devices. The software wait-state register (SWWSR) controls the operation of the wait-state generator. The SWWSR of a particular DSP subsystem (A or B) is used for the external memory interface, depending on the state of the xDMA/XIO arbitration logic (see Direct Memory Access (DMA) Controller section 3.8 and Table 3–4. The 14 least significant bits (LSBs) of the SWWSR specify the number of wait states (0–7) to be inserted for external memory accesses to five separate address ranges. This allows a different number of wait states for each of the five address ranges.

Additionally, the software wait-state multiplier (SWSM) bit of the software wait-state control register (SWCR) defines a multiplication factor of 1 or 2 for the number of wait states. At reset, the wait-state generator is initialized to provide seven wait states on all external memory accesses. The SWWSR bit fields are shown in Figure 3–3 and described in Table 3–4.

15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
XPA	I/O	Data	Data	Program	Program
R/W=0	R/W=111	R/W=111	R/W=111	R/W=111	R/W=111

LEGEND: R = Read, W = Write

Figure 3–3. Software Wait-State Register (SWWSR) [Memory-Mapped Register (MMR) Address 0028h]

Table 3–4. Software Wait-State Register (SWWSR) Bit Fields

BIT NO.	BIT NAME	RESET VALUE	FUNCTION
15	XPA	0	Extended program address control bit. XPA is used in conjunction with the program space fields (bits 0 through 5) to select the address range for program space wait states.
14–12	I/O	1	I/O space. The field value (0–7) corresponds to the base number of wait states for I/O space accesses within addresses 0000–FFFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
11–9	Data	1	Upper data space. The field value (0–7) corresponds to the base number of wait states for external data space accesses within addresses 8000–FFFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
8–6	Data	1	Lower data space. The field value (0–7) corresponds to the base number of wait states for external data space accesses within addresses 0000–7FFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
5–3	Program	1	Upper program space. The field value (0–7) corresponds to the base number of wait states for external program space accesses within the following addresses: ❑ XPA = 0: x8000–xFFFFh ❑ XPA = 1: The upper program space bit field has no effect on wait states. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
2–0	Program	1	Program space. The field value (0–7) corresponds to the base number of wait states for external program space accesses within the following addresses: ❑ XPA = 0: x0000–x7FFFh ❑ XPA = 1: 00000–3FFFFh The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.

The software wait-state multiplier bit of the software wait-state control register (SWCR) is used to extend the base number of wait states selected by the SWWSR. The SWCR bit fields are shown in Figure 3–4 and described in Table 3–5.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															SWSM
R/W=0															R/W=0

LEGEND: R = Read, W = Write

Figure 3–4. Software Wait-State Control Register (SWCR) [MMR Address 002Bh]

Table 3–5. Software Wait-State Control Register (SWCR) Bit Fields

PIN NO.	PIN NAME	RESET VALUE	FUNCTION
15–1	Reserved	0	These bits are reserved and are unaffected by writes.
0	SWSM	0	Software wait-state multiplier. Used to multiply the number of wait states defined in the SWWSR by a factor of 1 or 2. ❑ SWSM = 0: wait-state base values are unchanged (multiplied by 1). ❑ SWSM = 1: wait-state base values are multiplied by 2 for a maximum of 14 wait states.

3.5.2 Programmable Bank-Switching

Programmable bank-switching can be used to insert one cycle automatically when crossing memory-bank boundaries inside program memory or data memory space. One cycle can also be inserted when crossing from program-memory space to data-memory space (54x) or one program memory page to another program memory page. This extra cycle allows memory devices to release the bus before other devices start driving the bus, thereby avoiding bus contention. The size of the memory bank for the bank-switching is defined by the bank-switching control register (BSCR), as shown in Figure 3–5. The BSCR of a particular DSP subsystem (A or B) is used for the external memory interface based on the xDMA/XIO arbitration logic. The BSCR bit fields are described in Table 3–6.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BNKCMP				PS-DS	Reserved		IPIRQ	Reserved					BH	Reserved	EXIO
R/W				R/W	R/W		R/W						R/W		R/W

LEGEND: R = Read, W = Write

Figure 3–5. BSCR Register Bit Layout for Each DSP Subsystem

Table 3–6. BSCR Register Bit Functions for Each DSP Subsystem

BIT NO.	BIT NAME	RESET VALUE	FUNCTION
15–12	BNKCMP	1111	Bank compare. BNKCMP determines the external memory-bank size. BNKCMP is used to mask the four most significant bits (MSBs) of an address. For example, if BNKCMP = 1111b, the four MSBs (bits 12–15) are compared, resulting in a bank size of 4K words. Bank sizes of 4K words to 64K words are allowed.
11	PS-DS	1	Program read – data read access. PS-DS inserts an extra cycle between consecutive accesses of program read and data read or data read and program read. PS-DS = 0 No extra cycles are inserted by this feature. PS-DS = 1 One extra cycle is inserted between consecutive data and program reads.
10–9	Reserved	0	These bits are reserved and are unaffected by writes.
8	IPIRQ	0	The IPIRQ bit is used to send an interprocessor interrupt to the other subsystem. IPIRQ=1 sends the interrupt. IPIRQ must be cleared before subsequent interrupts can be made. Refer to the interrupts section for more details.
7–3	Reserved	0	These bits are reserved and are unaffected by writes.
2	BH	0	Bus holder. BH controls the bus holder feature: BH is cleared to 0 at reset. BH = 0 The bus holder is disabled. BH = 1 The bus holder is enabled. When not driven, PPD[15:0] pins are held at the previous logic level.
1	Reserved	0	This bit is reserved and is unaffected by writes.
0	EXIO	0	External bus interface off. The EXIO bit controls the external bus-off function. EXIO = 0 The external bus interface functions as usual. EXIO = 1 The address bus, data bus, and control signals become inactive after completing the current bus cycle. Note that the DROM, MP/$\overline{MC}$, and OVLY bits in the PMST and the HM bit of ST1 cannot be modified when the interface is disabled.

3.5.3 Parallel I/O Ports

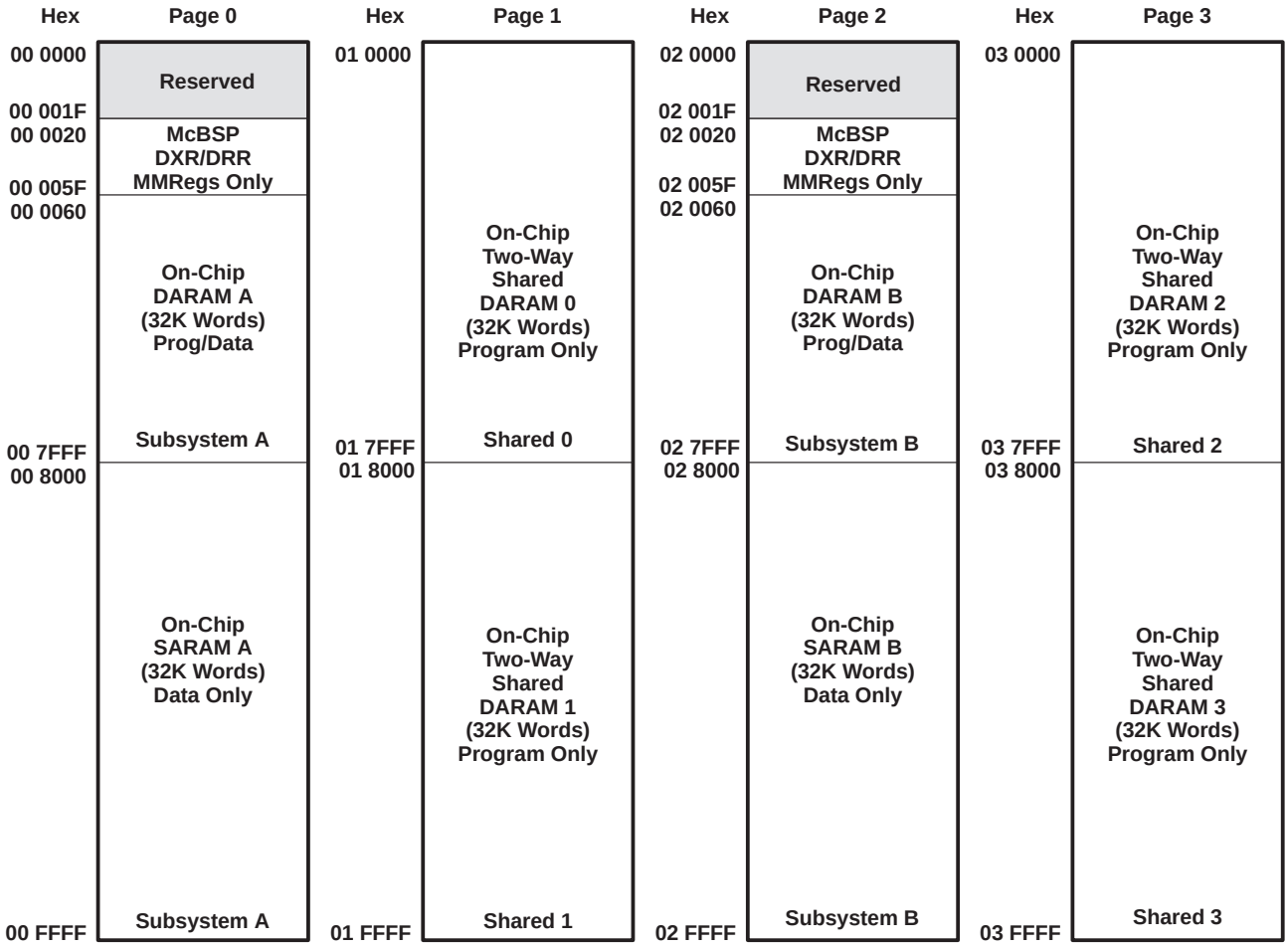
The 5421 has a total of 64K words of I/O port address space. These ports can be addressed by PORTR and PORTW. The $\overline{IS}$ signal indicates the read/write access through an I/O port. The devices can interface easily with external devices through the I/O ports while requiring minimal off-chip address-decoding logic. The SELA/B pin selects which subsystem is accessing the external I/O space.

3.6 16-Bit Bidirectional Host-Port Interface (HPI16)

The HPI16 is an enhanced 16-bit version of the TMS320C54x™ DSP 8-bit host-port interface (HPI). The HPI16 is designed to allow a 16-bit host to access the DSP on-chip memory, with the host acting as the master of the interface.

3.6.1 HPI16 Memory Map

Figure 3–6 illustrates the available memory accessible by the HPI. Neither the CPU nor DMA I/O spaces can be accessed using the host-port interface.



NOTES: A. All local memory is available to the HPI

B. The encoder maps CPU A Data Page 0 into the HPI Page 0. CPU B Data Page 0 is mapped into the HPI Page 2. Pages 1 and 3 are the on-chip shared program memory.

C. In pages 00 and 02, in the range of 0020–005F, only the following memory mapped registers are accessible: 20,21,30,31,40,41 (read only), 22,23,32,33,42,43 (write only).

Figure 3–6. Memory Map Relative to Host-Port Interface HPI16

3.6.2 HPI Features

Some of the features of the HPI16 include:

- 16-bit bidirectional data bus
- Multiple data strobes and control signals to allow glueless interfacing to a variety of hosts
- Multiplexed and nonmultiplexed address/data modes
- 18-bit address bus used in nonmultiplexed mode to allow access to all internal memory (including internal extended address pages)
- 18-bit address register used in multiplexed mode. Includes address autoincrement feature for faster accesses to sequential addresses
- Interface to on-chip DMA module to allow access to entire internal memory space
- HRDY signal to hold off host accesses due to DMA latency
- Control register available in multiplexed mode only. Accessible by either host or DSP to provide host/DSP interrupts, extended addressing, and data prefetch capability
- Maximum data rate of 33 megabytes per second (MBps) at 100-MHz DSP clock rate (no other DMA channels active)

The HPI16 acts as a slave to a 16-bit host processor and allows access to the on-chip memory of the DSP. There are two modes of operation as determined by the HMODE signal: *multiplexed* mode and *nonmultiplexed* mode.

3.6.3 HPI Multiplexed Mode

In *multiplexed* mode, HPI16 operation is very similar to that of the standard 8-bit HPI, which is available with other C54x™ DSP products. A host with a multiplexed address/data bus can access the HPI16 data register (HPID), address register (HPIA), or control register (HPIC) via the HD bidirectional data bus. The host initiates the access with the strobe signals ($\overline{\text{HDS1}}$, $\overline{\text{HDS2}}$, $\overline{\text{HCS}}$) and controls the type of access with the HCNTL, HR/$\overline{\text{W}}$, and $\overline{\text{HAS}}$ signals. The DSP can interrupt the host via the $\overline{\text{HINT}}$ signal, and can stall host accesses via the HRDY signal.

3.6.4 Host/DSP Interrupts

In *multiplexed* mode, the HPI16 offers the capability for the host and DSP to interrupt each other through the HPIC register.

For host-to-DSP interrupts, the host must write a “1” to the DSPINT bit of the HPIC register. This generates an interrupt to the DSP. This interrupt can also be used to wake the DSP from any of the IDLE 1,2, or 3 states. Note that the DSPINT bit is always read as “0” by both the host and DSP. The DSP cannot write to this bit (see Figure 3–7).

For DSP-to-host interrupts, the DSP must write a “1” to the $\overline{\text{HINT}}$ bit of the HPIC register to interrupt the host via the $\overline{\text{HINT}}$ pin. The host acknowledges and clears this interrupt by also writing a “1” to the $\overline{\text{HINT}}$ bit of the HPIC register. Note that writing a “0” to the $\overline{\text{HINT}}$ bit by either host or DSP has no effect.

3.6.5 Emulation Considerations

The HPI16 can continue operation even when the DSP CPU is halted due to debugger breakpoints or other emulation events.

3.6.6 HPI Nonmultiplexed Mode

In *nonmultiplexed* mode, a host with separate address/data buses can access the HPI16 data register (HPID) via the HD 16-bit bidirectional data bus, and the address register (HPIA) via the 18-bit HA address bus. The host initiates the access with the strobe signals ($\overline{\text{HDS1}}$, $\overline{\text{HDS2}}$, $\overline{\text{HCS}}$) and controls the direction of the access with the HR/$\overline{\text{W}}$ signal. The HPI16 can stall host accesses via the HRDY signal. Note that the HPIC register is not available in *nonmultiplexed* mode since there are no HCNTL signals available. All host accesses initiate a DMA read or write access. Figure 3–7 shows a block diagram of the HPI16 in *nonmultiplexed* mode.

C54x is a trademark of Texas Instruments.

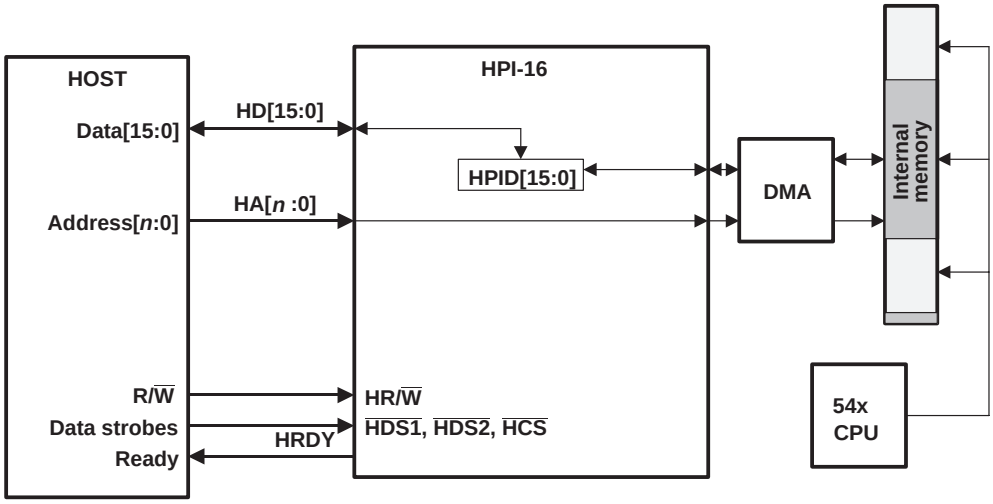


Figure 3–7. Interfacing to the HPI-16 in Non-Multiplexed Mode

3.6.7 Other HPI16 System Considerations

3.6.7.1 Operation During IDLE

The HPI16 can continue to operate during IDLE1 or IDLE2 by using special clock management logic that turns on relevant clocks to perform a synchronous memory access, and then turns the clocks back off to save power. The DSP CPU does not wake up from the IDLE mode during this process.

3.6.7.2 Downloading Code During Reset

The HPI16 can download code while the DSP is in reset. However, the system provides a pin ($\overline{\text{HPIRS}}$) that provides a way to take the HPI16 module out of reset while leaving the DSP in reset. The maximum HPI16 data rate is 33 MBps assuming no other DMA activity (100-MIPS DSP subsystem).

3.6.7.3 Performance Issues

On the 5421, the use of SELA/B is optional for access to all on-chip memory. However, with both the 5420 and 5421 implementation using two separate subsystems (subchips), the SELA/B pin is used to select the specific HPI16 used to access memory.

SELA/B PIN	SUBSYSTEM
0	A
1	B

Accesses to memory contained inside the same subsystem as the selected HPI16 will be faster. For accesses to an HPI16 in a subsystem different than the memory being addressed, reads take an additional six cycles and writes an extra five cycles. Therefore, for performance reasons, it is best to additionally decode SELA/B.

3.7 Multichannel Buffered Serial Port (McBSP)

The 5421 device provides high-speed, full-duplex serial ports that allow direct interface to other C54x/LC54x devices, codecs, and other devices in a system. There are six multichannel buffered serial ports (McBSPs) on board (three per subsystem).

The McBSP provides:

- Full-duplex communication
- Double-buffer data registers, which allow a continuous data stream
- Independent framing and clocking for receive and transmit

In addition, the McBSP has the following capabilities:

- Direct interface to:
  - T1/E1 framers
  - MVIP switching-compatible and ST-BUS compliant devices
  - IOM-2 compliant device
  - AC97-compliant device
  - Serial peripheral interface (SPI)
- Multichannel transmit and receive of up to 128 channels
- A wide selection of data sizes, including: 8, 12, 16, 20, 24, or 32 bits
- μ-law and A-law companding
- Programmable polarity for both frame synchronization and data clocks
- Programmable internal clock and frame generation

The 5421 McBSPs have been enhanced to provide more flexibility in the choice of the sample rate generator input clock source. On previous TMS320C5000™ DSP platform devices, the McBSP sample rate input clock can be driven from one of two possible choices: the internal CPU clock , or the external CLKS pin. However, most C5000™ DSP devices have only the internal CPU clock as a possible source because the CLKS pin is not implemented on most device packages.

To accommodate applications that require an external reference clock for the sample rate generator, the 5421 McBSPs allow either the receive clock pin (BCLKR) or the transmit clock pin (BCLKX) to be configured as the input clock to the sample rate generator. This enhancement is enabled through two register bits: pin control register (PCR) bit 7 – enhanced sample clock mode (SCLKME), and sample rate generator register 2 (SRGR2) bit 13 – McBSP sample rate generator clock mode (CLKSM). SCLKME is an addition to the PCR contained in the McBSPs on previous C5000 devices. The new bit layout of the PCR is shown in Figure 3–8. For a description of the remaining bits, see *TMS320C54x DSP Reference Set, Volume 5: Enhanced Peripherals* (literature number SPRU302).

TMS320C5000 and C5000 are trademarks of Texas Instruments.

15	14	13	12	11	10	9	8
Reserved		XIOEN	RIOEN	FSXM	FSRM	CLKXM	CLKRM
R,+0		RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0

7	6	5	4	3	2	1	0
SCLKME	CLKS_STAT	DX_STAT	DR_STAT	FSXP	FSRP	CLKXP	CLKRP
RW,+0	R,+0	R,+0	R,+0	RW,+0	RW,+0	RW,+0	RW,+0

Note: R = Read, W = Write, +0 = Value at reset

Figure 3–8. Pin Control Register (PCR)

The selection of the sample rate generator (SRG) clock input source is made by the combination of the CLKSM and SCLKME bit values as shown in Table 3–7.

Table 3–7. Sample Rate Generator Clock Source Selection

SCLKME	CLKSM	SRG Clock Source
0	0	CLKS (not available as a pin on 5421)
0	1	CPU clock
1	0	BCLKR pin
1	1	BCLKX pin

When either of the bidirectional pins, BCLKR or BCLKX, is configured as the clock input, its output buffer is automatically disabled. For example, with SCLKME = 1 and CLKSM = 0, the BCLKR pin is configured as the SRG input. In this case, both the transmitter and receiver circuits can be synchronized to the SRG output by setting the PCR bits (9:8) for CLKXM = 1 and CLKRM = 1. However, the SRG output is only driven onto the BCLKX pin because the BCLKR output is automatically disabled.

The McBSP supports independent selection of multiple channels for the transmitter and receiver. When multiple channels are selected, each frame represents a time-division multiplexed (TDM) data stream. In using time-division multiplexed data streams, the CPU may only need to process a few of them. Thus, to save memory and bus bandwidth, multichannel selection allows independent enabling of particular channels for transmission and reception. Up to a maximum of 128 channels in a bit stream can be enabled or disabled.

The 5421 McBSPs have two working modes that are selected by setting the RMCME and XMCME bits in the multichannel control registers (MCR1x and MCR2x, respectively). See Figure 3–9 and Figure 3–10. For a description of the remaining bits, see *TMS320C54x DSP Reference Set, Volume 5: Enhanced Peripherals* (literature number SPRU302).

- In the first mode, when RMCME = 0 and XMCME = 0, there are two partitions (A and B), with each containing 16 channels as shown in Figure 3–9 and Figure 3–10. This is compatible with the McBSPs used in the 5420, where only 32-channel selection is enabled (default).

15 14 13 12 11 10	9	8 7	6 5	4 3 2	1 0
Reserved	XMCME	XPBBLK	XPABLK	XCBLK	XMCM
R,+0	RW,+0	RW,+0	RW,+0	R,+0	RW,+0

Note: R = Read, W = Write, +0 = Value at reset; x = McBSP 0,1, or 2

Figure 3–9. Multichannel Control Register 2x (MCR2x)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						RMCME	RPBBLK		RPABLK		RCBLK			RMCM	
R,+0						RW,+0	RW,+0		RW,+0		R,+0			RW,+0	

Note: R = Read, W = Write, +0 = Value at reset; x = McBSP 0,1, or 2

Figure 3–10. Multichannel Control Register 1x (MCR1x)

- In the second mode, with RMCME = 1 and XMCME = 1, the McBSPs have 128 channel selection capability. Twelve new registers (RCERCx–RCERHx and XCERCx–XCERHx) are used to enable the 128 channel selection. The subaddresses of the new registers are shown in Table 3–21. These new registers, functionally equivalent to the RCERA0–RCERB1 and XCERA0–XCERB1 registers in the 5420, are used to enable/disable the transmit and receive of additional channel partitions (C,D,E,F,G, and H) in the 128 channel stream. For example, XCERH1 is the transmit enable for channel partition H (channels 112 to 127) of MCBSP1 for each DSP subsystem. See Figure 3–11, Table 3–8, Figure 3–12, and Table 3–9 for bit layout and function of the receive and transmit registers .

15	14	13	12	11	10	9	8
RCERyz15	RCERyz14	RCERyz13	RCERyz12	RCERyz11	RCERyz10	RCERyz9	RCERyz8
RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0
7	6	5	4	3	2	1	0
RCERyz7	RCERyz6	RCERyz5	RCERy4	RCERyz3	RCERyz2	RCERyz1	RCERyz0
RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0

Note: R = Read, W = Write, +0 = Value at reset; y = Partition A,B,C,D,E,F,G, or H; z = McBSP 0,1, or 2

Figure 3–11. Receive Channel Enable Registers Bit Layout for Partitions A to H

Table 3–8. Receive Channel Enable Registers for Partitions A to H

Bit	Name	Function	
15–0	RCERyz(15:0)	Receive Channel Enable Register	
		RCERyz *n* = 0	Disables reception of *n*th channel in partition y.
		RCERyz *n* = 1	Enables reception of *n*th channel in partition y.

Note: y = Partition A,B,C,D,E,F,G, or H; z = McBSP 0,1, or 2; n = bit 15–0

15	14	13	12	11	10	9	8
XCERyz15	XCERyz14	XCERyz13	XCERyz12	XCERyz11	XCERyz10	XCERyz9	XCERyz8
RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0
7	6	5	4	3	2	1	0
XCERyz7	XCERyz6	XCERyz5	XCERy4	XCERyz3	XCERyz2	XCERyz1	XCERyz0
RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0

Note: R = Read, W = Write, +0 = Value at reset; y = Partition A,B,C,D,E,F,G, or H; z = McBSP 0,1, or 2

Figure 3–12. Transmit Channel Enable Registers Bit Layout for Partitions A to H

Table 3–9. Transmit Channel Enable Registers for Partitions A to H

Bit	Name	Function	
15–0	XCERyz(15:0)	Transmit Channel Enable Register	
		XCERyz *n* = 0	Disables transmit of *n*th channel in partition y.
		XCERyz *n* = 1	Enables transmit of *n*th channel in partition y.

Note: y = Partition A,B,C,D,E,F,G, or H; z = McBSP 0,1, or 2; n = bit 15–0

The clock stop mode (CLKSTP) in the McBSP provides compatibility with the serial port interface (SPI) protocol. Clock stop mode works with only single-phase frames and one word per frame. The word sizes supported by the McBSP are programmable for 8-, 12-, 16-, 20-, 24-, or 32-bit operation. When the McBSP is configured to operate in SPI mode, both the transmitter and the receiver operate together as a master or as a slave.

The McBSP is fully static and operates at arbitrarily low clock frequencies. The maximum McBSP multichannel operating frequency on the 5421 is 9 MBps. Nonmultichannel operation is limited to 38 MBps.

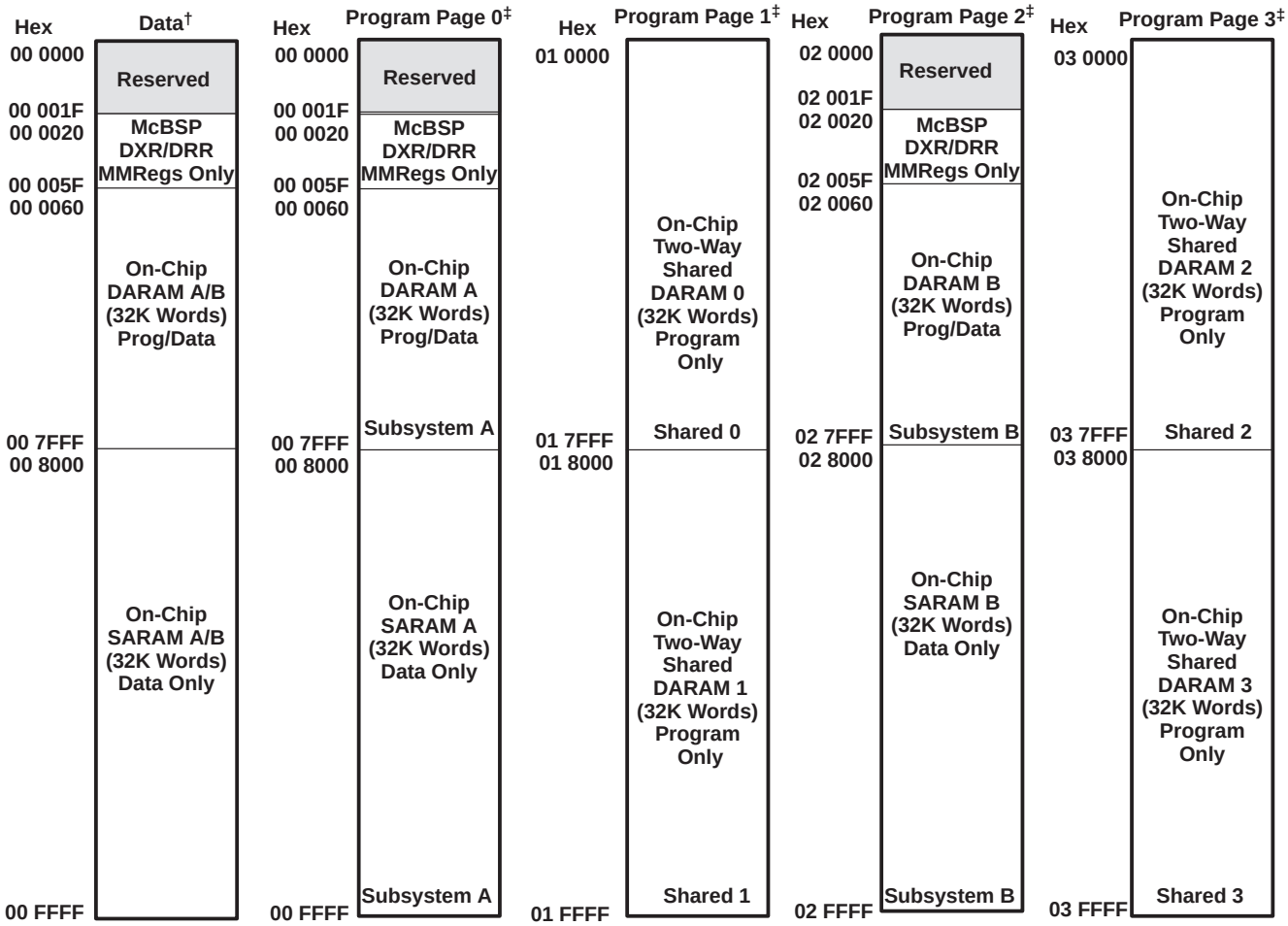
3.7.1 Emulation Considerations

The McBSP can continue operation even when the DSP CPU is halted due to debugger breakpoints or other emulation events.

3.8 Direct Memory Access (DMA) Controller

The 5421 includes two 6-channel direct memory access (DMA) controllers for performing data transfers independent of the CPU, one for each subsystem. The DMA controller controls accesses to off-chip program/data/IO and internal data/program memory. The primary function of the 5421 DMA controller is to provide code overlays and manage data transfers between on-chip memory, the peripherals, and off-chip memory.

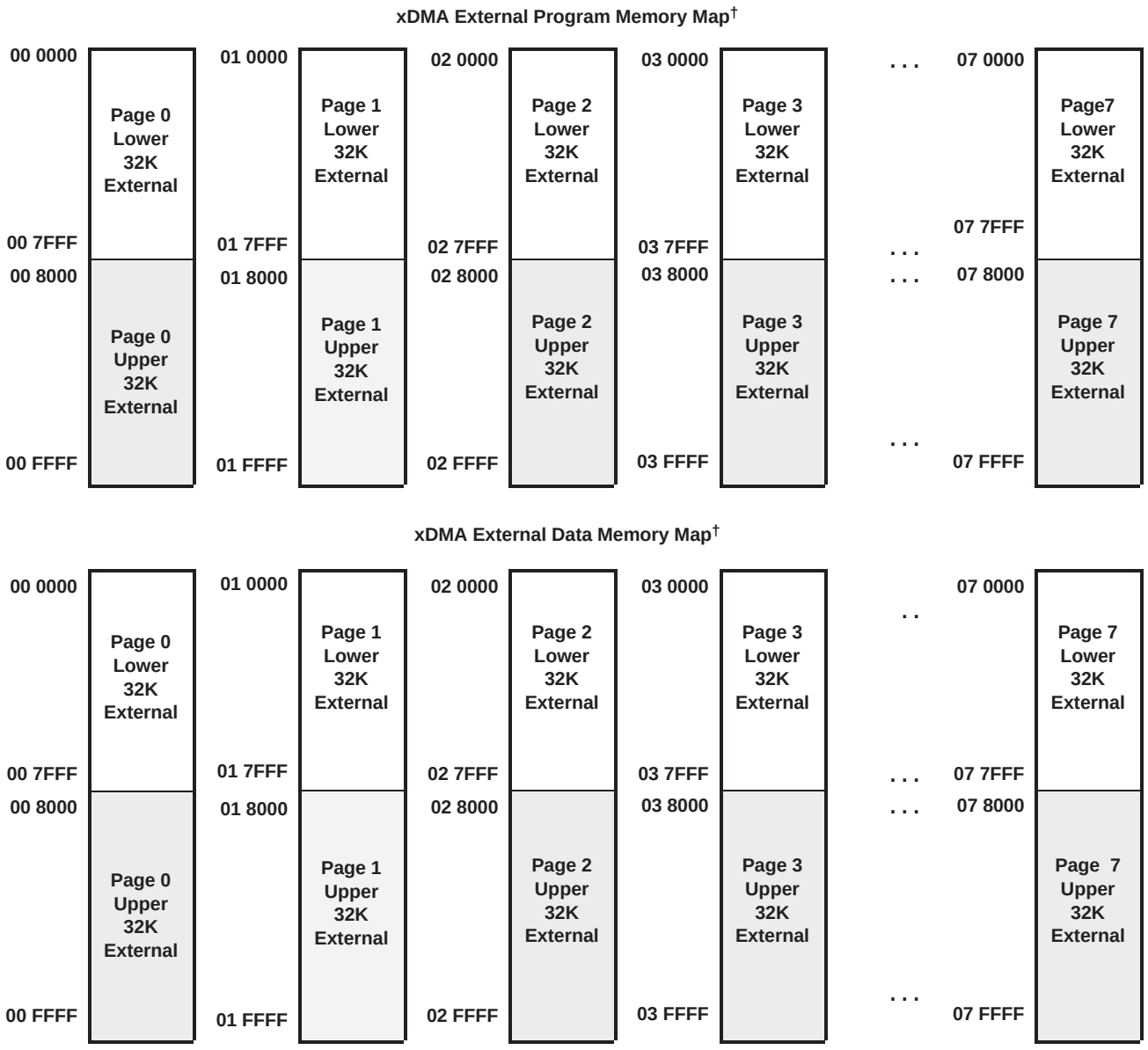
In the background of CPU operation, the 5421 DMA allows movement of data between internal and external program/data memory, and internal peripherals, such as the McBSPs and the HPI. Each subsystem has its own independent DMA with six programmable channels, which allows for six different contexts for DMA operation. The HPI has a dedicated auxiliary DMA channel. Figure 3–13 illustrates the memory map accessible by the DMA.



† DMD/DMS = 01
‡ DMD/DMS = 00

NOTES: A. All local memory is available to the DMA.
B. All I/O memory accesses by the DMA (DMD/DMS = 10) are mapped to the core-to-core FIFO.
C. In pages 00 and 02, in the range of 0020–005F, only the following memory mapped registers are accessible: 20,21,30,31,40,41 (read only), 22,23,32,33,42,43 (write only).

Figure 3–13. On-Chip Memory Map Relative to DMA (DLAXS/SLAXS = 0)



† Pages 8 – 127 are overlaid over pages 0 – 7.

Figure 3–14. DMA External Program Memory Map

3.8.1 DMA Controller Features

The 5421 DMA has the following features:

- The DMA operates independently of the CPU.
- The DMA has six channels. The DMA can keep track of the contexts of six independent block transfers.
- Two DMA channels are available for external accesses: one for reads and one for writes.
- The DMA has higher priority than the CPU for internal accesses.
- Each channel has independently programmable priorities.
- Each channel's source and destination address registers include configurable indexing modes. The address can be held constant, postincremented, postdecremented, or adjusted by a programmable value.
- For internal accesses, each read or write transfer can be initialized by selected events.
- Supports 32-bit transfers for internal accesses only.
- Single-word (16-bit) transfers are supported for external accesses.
- The DMA does not support transfers from peripherals to external memory.
- The DMA does not support transfers from external memory to the peripherals.
- The DMA does not support external to external transfers.

A 16-bit DMA transfer requires four CPU clock cycles to complete — two cycles for reads and two cycles for writes. This gives a maximum DMA throughput of 50 MBps. Since the DMA controller shares the DMA bus with the HPI module, the DMA access rate is reduced when the HPI is active.

3.8.2 DMA Accesses to External Memory

The 5421 DMAs supports external accesses to extended program, extended data, and extended I/O memory. These overlay pages are only visible to the DMA controller. A maximum of two channels (one for reads, one for writes) per DMA can be used for external memory accesses. The DMA external accesses require 9 cycles (minimum) for external writes and 13 cycles (minimum) for external reads.

The control of the bus is arbitrated between the two CPUs and the two DMAs. While one DMA or CPU is in control of the external bus, the other three components will be held off (via wait-states) until the current transfer is complete. The DMA takes precedence over XIO requests. The $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ feature of the 5421 affects external CPU transfers, as well as external DMA transfers. When an external processor asserts the $\overline{\text{HOLD}}$ pin to gain control of the memory interface, the $\overline{\text{HOLDA}}$ signal is not asserted until all pending DMA transfers are completed. To prevent a DMA from blocking out the CPUs or $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ feature from accessing the external bus, uninterrupted burst transfers are ***not*** supported by the DMAs. Subsequently, CPU and DMA arbitration testing is performed for each external bus cycle, regardless of the bus activity. With the completion of each block, the highest priority will be swapped.

For arbitration at the DSP subsystem level, the DMA requests (DMA_REQ_A or DMA_REQ_B) from either DMA will be sent to both CPUs as shown in Figure 3–15. Regardless of which CPU controls the external pin interface (XIO), both CPUs must send a grant (GRANT_A, GRANT_B) for control of the bus to be released to the DMAs.

Arbitration between CPUs is done using a request/grant scheme. Prior to accessing XIO of one of the CPUs, software is responsible for asserting a request for access to the device pins and polling grant status until the pins are granted to the requestor. If both CPUs request the bus simultaneously, subsystem A is granted priority. For details on memory-mapped register bits pertaining to CPU XIO arbitration, see the general-purpose I/O control register bits [6:4] (CORE SEL, XIO GRANT, XIO REQ) in Table 3–14.

At reset, the default is that subsystem A has access to the device pins. Accesses without a grant will be allowed, but do not show up on the device pins.

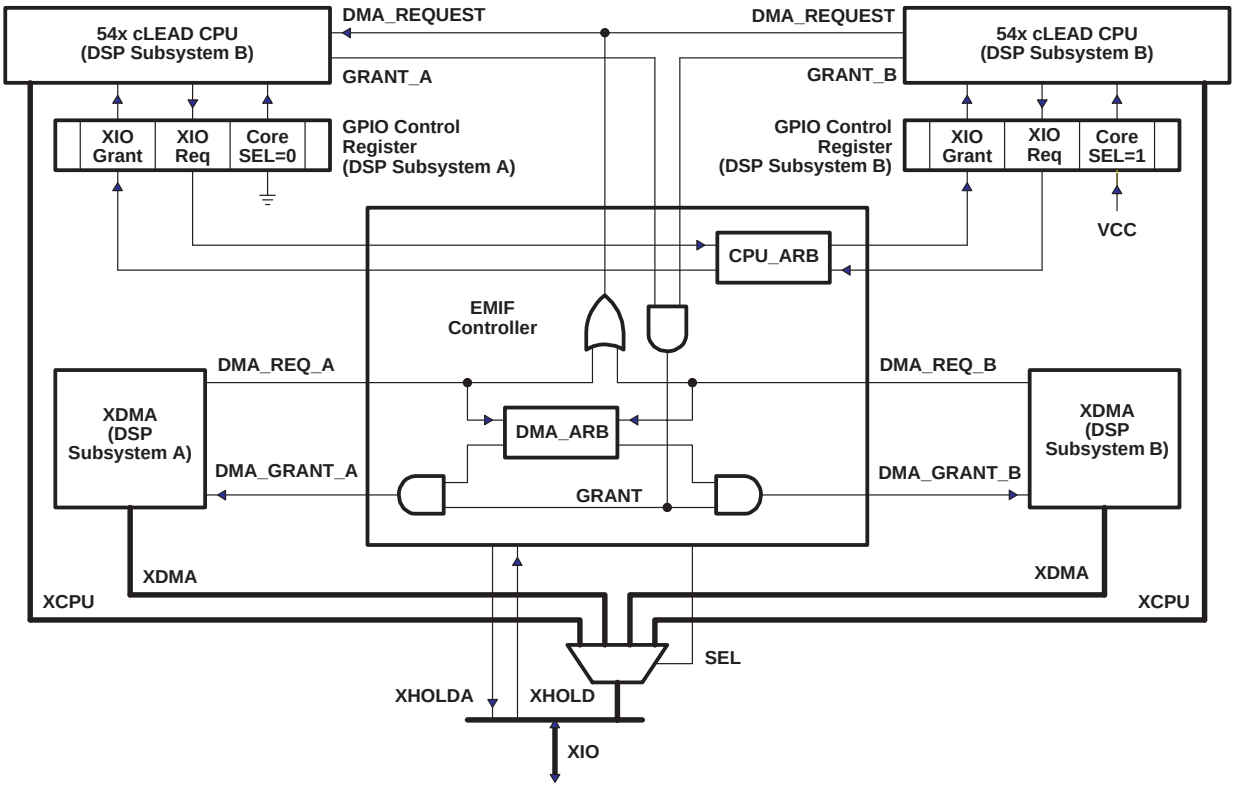


Figure 3–15. Arbitration Between XIO and xDMA for External Access

The HM bit in the ST1 indicates whether the processor continues internal execution when acknowledging an active $\overline{\text{HOLD}}$ signal.

- HM = 0, the processor continues execution from internal program memory but places its external interface in the high-impedance state.
- When HM = 1, the processor halts internal execution.

To ensure that proper arbitration occurs, the HM bit should be set to 0 in the memory-mapped ST1 registers for both CPUs.

To allow the DMA access to extended data pages, the SLAXS and DLAXS bits are added to the DMMCRn registers. For a description of the remaining bits, see *TMS320C54x DSP Reference Set, Volume 5: Enhanced Peripherals* (literature number SPRU302).

15	14	13	12	11	10 9 8	7 6	5	4 3 2	1 0
AUTO INIT	DINM	IMOD	CT MOD	SLAXS	SIND	DMS	DLAXS	DIND	DMD

Figure 3–16. DMA Transfer Mode Control Register (DMMCRn)

These new bit fields were created to allow the user to define the space-select for the DMA (internal/external). Also, a new extended destination data page (XDSTDP[6:0], subaddress 029h) and extended source data page (XSRCDP[6:0], subaddress 028h) have been created.

DLAXS(DMMCRn[5]) Destination	0 = No external access (default internal) 1 = External access
SLAXS(DMMCRn[11]) Source	0 = No external access (default internal) 1 = External access

For the CPU external access, software can configure the memory cells to reside inside or outside the program address map. When the cells are mapped into program space, the device automatically accesses them when their addresses are within bounds. When the program address generation (PAGEN) logic generates an address outside its bounds, the device automatically generates an external access. All DMA I/O space accesses are mapped to the core-to-core FIFO.

3.8.3 DMA Controller Synchronization Events

The transfers associated with each DMA channel can be synchronized to one of several events. The DSYN bit field of the DMA channel x sync select and frame count (DMSFCx) register selects the synchronization event for a channel. The list of possible events and the DSYN values are shown in Table 3–10.

Table 3–10. DMA Synchronization Events

DSYN VALUE	DMA SYNCHRONIZATION EVENT
0000b	No synchronization used
0001b	McBSP0 Receive Event
0010b	McBSP0 Transmit Event
0011b	McBSP2 Receive Event
0100b	McBSP2 Transmit Event
0101b	McBSP1 Receive Event
0110b	McBSP1 Transmit Event
0111b	FIFO Receive Buffer Not Empty Event
1000b	FIFO Transmit Buffer Not Full Event
1001b – 1111b	Reserved

3.8.4 DMA Channel Interrupt Selection

The DMA controller can generate a CPU interrupt for each of the six channels. However, channels 0, 1, 2, and 3 are multiplexed with other interrupt sources. DMA channels 0 and 1 share an interrupt line with the receive and transmit portions of McBSP2 (IMR/IFR bits 6 and 7), and DMA channels 2 and 3 share an interrupt line with the receive and transmit portions of McBSP1 (IMR/IFR bits 10 and 11). When the 5421 is reset, the interrupts from these four DMA channels are deselected. The INTSEL bit field in the DMA channel priority and enable control (DMPREC) register can be used to select these interrupts, as shown in Table 3–11.

Table 3–11. DMA Channel Interrupt Selection

INTSEL Value	IMR/IFR[6]	IMR/IFR[7]	IMR/IFR[10]	IMR/IFR[11]
00b (reset)	BRINT2	BXINT2	BRINT1	BXINT1
01b	BRINT2	BXINT2	DMAC2	DMAC3
10b	DMAC0	DMAC1	DMAC2	DMAC3
11b	Reserved			

3.8.5 DMA in Autoinitialization Mode

The DMA can automatically reinitialize itself after completion of a block transfer. Some of the DMA registers can be preloaded for the next block transfer through the DMA global reload registers (DMGSA, DMGDA, DMGCR, and DMGFR). Autoinitialization allows:

- Continuous operation: Normally, the CPU would have to reinitialize the DMA immediately after the completion of the current block transfers, but with the global reload registers, it can reinitialize these values for the next block transfer any time after the current block transfer begins.
- Repetitive operation: The CPU does not preload the global reload register with new values for each block transfer but only loads them on the first block transfer.

The 5421 DMA has been enhanced to expand the DMA global reload register sets. Each DMA channel now has its own DMA global reload register set. For example, the DMA global reload register set for channel 0 has DMGSA0, DMGDA0, DMGCR0, and DMGFR0 while DMA channel 1 has DMGSA1, DMGDA1, DMGCR1, and DMGFR1, etc.

To utilize the additional DMA global reload registers, the AUTOIX bit is added to the DMPREC register as shown in Figure 3–17.

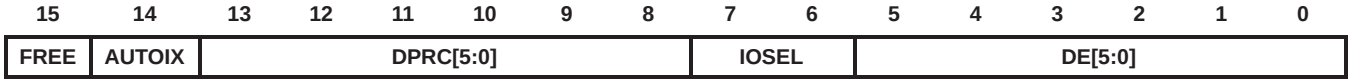


Figure 3–17. DMPREC Register

Table 3–12. DMA Global Reload Register Selection

AUTOIX	DMA GLOBAL RELOAD REGISTER USAGE IN AUTO INIT MODE
0 (default)	All DMA channels use DMGSA0, DMGDA0, DMGCR0 and DMGFR0
1	Each DMA channel uses its own set of global reload registers

3.8.6 Subsystem Communications

The 5421 device provides two options for efficient core-to-core communications:

- Core-to-core FIFO communications
- DMA global memory transfer

3.8.6.1 FIFO Data Communications

The subsystems' FIFO communications interface is shown in the 5421 functional block diagram (Figure 3–1). Two unidirectional 8-word-deep FIFOs are available in the device for efficient interprocessor communication: one configured for core A-to-core B data transfers, and the other configured for core B-to-core A data transfers. Each subsystem, by way of DMA control, can write to its respective output data FIFO and read from its respective input data FIFO. The FIFOs are accessed using the DMAs I/O space, which is completely independent of the CPU I/O space. The DMA transfers to or from the FIFOs can be synchronized to "receive FIFO not empty" and "transmit FIFO not full" events, providing protection from overflow and underflow. Subsystems can interrupt each other to flag when the FIFOs are either full or empty. The interprocessor interrupt request bit (IPIRQ) (bit 8 in the BSCR register (BSCR.8)) is set to 1 to generate a PINT in the other subsystem's IFR.14. See the *Interrupts* section (Section 3.13) for more information.

3.8.6.2 DMA Global Memory Transfers

The 5421 enables each subsystem to transfer data directly between the memories that are CPU local via DMA global memory transfers. The DMA global memory map is shown in Figure 3–13.

3.8.7 Chip Subsystem ID Register

The chip subsystem ID Register (CSIDR) is a read-only memory-mapped register located at 3Eh within each DSP subsystem. This register contains three elements for electrically readable device identification. The ChipID bits identify the type of 54x device (21h for 5421). The ChipRev bits contain the revision number of the device. Lastly, the SubSysID contains a unique subsystem identifier.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Chip ID								Chip Rev				SubSysID			

Figure 3–18. Chip Subsystem ID Register

Table 3–13. Chip Subsystem ID Register Bit Functions

BIT NO.	BIT FIELD NAME	FUNCTION
15–8	Chip ID	54x device type. Contains 21h for 5421.
7–4	Chip Rev	Revision number of device (i.e., 0h for revision 0).
3–0	SubSysID	Identifier for DSP subsystem: A = 0h, B = 1h.

3.9 General-Purpose I/O

In addition to the A_XF and B_XF pins, the 5421 has eight general-purpose I/O pins. These pins are:

A_GPIO0, A_GPIO1, A_GPIO2, A_GPIO3

B_GPIO0, B_GPIO1, B_GPIO2, B_GPIO3

Four general-purpose I/O pins are available to each core. Each GPIO pin can be individually selected as either an input or an output. Additionally, the timer output is selectable on GPIO pin 3. At core reset, all GPIO pins are configured as inputs. GPIO data and control bits are accessible through a memory-mapped register at 3Ch with the format shown in Figure 3–19.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOUT	Reserved			GPIO DIR3	GPIO DIR2	GPIO DIR1	GPIO DIR0	ROM EN	CORE SEL	XIO GRANT	XIO REQ	GPIO DAT3	GPIO DAT2	GPIO DAT1	GPIO DAT0
R/W+0				R/W+0	R/W+0	R/W+0	R/W+0	R/W†	R	R+0	R/W+0	R/W+0	R/W+0	R/W+0	R/W+0

† 1 denotes XIO = 1, 0 denotes XIO = 0

Note: R = Read, W = Write, +0 = Value at reset

Figure 3–19. General-Purpose I/O Control Register

Table 3–14. General-Purpose I/O Control Register Bit Functions

BIT NO.	BIT NAME	BIT VALUE	FUNCTION
15	TOUT	0	Timer output disable. Uses GPIO3 as general-purpose I/O.
		1	Timer output enable. Overrides DIR3. Timer output is driven on GPIO3 and readable in DAT3.
14-12	Reserved	X	Register bit is reserved. Read 0, write has no effect.
11–8	GPIO DIRn†	0	GPIOn pin is used as an input.
		1	GPIOn pin is used as an output.
7	ROMEN‡	0	ROM is mapped out (value at reset if XIO = 0)
		1	ROM is mapped in (value at reset if XIO = 1)
6	CORE SEL	0	cLEAD core A is selected for XIO REQ bit. DSP subsystem A is tied low internally for this bit.
		1	cLEAD core B is selected for XIO REQ bit. DSP subsystem B is tied high internally for this bit.
5	XIO GRANT	0	EMIF is not available to the cLEAD core determined by the CORE SEL bit.
		1	EMIF is granted to the cLEAD core determined by the CORE SEL bit.
4	XIO REQ	0	EMIF is not requested for the cLEAD core indicated by the CORE SEL bit.
		1	Request EMIF for the cLEAD core indicated by the CORE SEL bit.
3–0	GPIO DATn†	0	GPIOn is driven with a 0 (DIRn = 1). GPIOn is read as 0 (DIRn = 0).
		1	GPIOn is driven with a 1 (DIRn = 1). GPIOn is read as 1 (DIRn = 0).

† n = 3, 2, 1, or 0
‡ 1 denotes XIO = 1, 0 denotes XIO = 0

Register bit 7 is used as ROMEN to enable and disable ROM space. In XIO mode, ROM enable (ROMEN) reflects the state of the A_GPIO0 and B_GPIO0 pins (GPIODAT0 input) to enable the applicable on-chip ROM after reset. Register bits (6:4) are used for XIO arbitration of external memory interface (EMIF) control between DSP subsystems. The timer out (TOUT) bit is used to multiplex the output of the timer and GPIO3. All GPIO pins are programmable as an input or output by the direction bit (DIRn). Data is either driven or read from the data bit field (DATn). DIR3 has no affect when TOUT = 1.

GPIO2 is a special case where the logic level determines the operation of $\overline{\text{BIO}}$-conditional instructions on the CPU. GPIO2 is always mapped as a general-purpose I/O, but the $\overline{\text{BIO}}$ function exists when this pin is configured as an input.

3.9.1 Hardware Timer

The 54x devices feature a 16-bit timing circuit with a 4-bit prescaler. The timer counter decrements by one at every CLKOUT cycle. Each time the counter decrements to zero, a timer interrupt is generated. The timer can be stopped, restarted, reset, or disabled by specific status bits. The timer output pulse is driven on GPIO3 when the TOUT bit is set to one in the general-purpose I/O control register. The device must be in HPI mode (XIO = 0) to drive TOUT on the GPIO3 pin.

3.9.2 Software-Programmable Phase-Locked Loop (PLL)

The clock generator provides clocks to the 5421 device, and consists of a phase-locked loop (PLL) circuit. The clock generator requires a reference clock input, which must be provided by using an external clock source. The reference clock input is then divided by two (DIV mode) to generate clocks for the 5421 device. Alternately, the PLL circuit can be used (PLL mode) to generate the device clock by multiplying the reference clock frequency by a scale factor, allowing use of a clock source with a lower frequency than that of the CPU. Bypass (multiply by 1) is the default mode at reset. The PLL is an adaptive circuit that, once synchronized, locks onto and tracks an input clock signal. When the PLL is initially started, it enters a transitional mode during which the PLL acquires lock with the input signal. Once the PLL is locked, it continues to track and maintain synchronization with the input signal. Then, other internal clock circuitry allows the synthesis of new clock frequencies for use as master clock for the 5421 device. Only subsystem A controls the PLL. Subsystem B cannot access the PLL registers.

The software-programmable PLL features a high level of flexibility, and includes a clock scaler that provides various clock multiplier ratios, capability to directly enable and disable the PLL, and a PLL lock timer that can be used to delay switching to PLL clocking mode of the device until lock is achieved. Devices that have a built-in software-programmable PLL can be configured in one of two clock modes:

- PLL mode. The input clock (CLKIN) is multiplied by 1 of 31 possible ratios. These ratios are achieved using the PLL circuitry.
- DIV (divider) mode. The input clock is divided by 2 or 4. Note that when DIV mode is used, the PLL can be completely disabled in order to minimize power dissipation.

The software-programmable PLL is controlled using the 16-bit memory-mapped (address 0058h) clock mode register (CLKMD). The CLKMD register is used to define the clock configuration of the PLL clock module. Figure 3–20 shows the bit layout of the clock mode register and Table 3–15 describes the bit functions.

15 12	11	10 3	2	1	0
PLLMUL†	PLLDIV†	PLLCOUNT†	PLLON/OFF†	PLLNDIV	STATUS
R/W	R/W	R/W	R/W	R/W	R/W

† When in DIV mode (PLLSTATUS is low), PLLMUL, PLLDIV, PLLCOUNT, and PLLON/OFF are don't cares, and their contents are indeterminate.

LEGEND: R = Read, W = Write

Figure 3–20. Clock Mode Register (CLKMD)

Table 3–15. Clock Mode Register (CLKMD) Bit Functions

BIT NO.	BIT NAME	FUNCTION
15–12	PLLMUL†	PLL multiplier. PLLMUL defines the frequency multiplier in conjunction with PLLDIV and PLLNDIV. See Table 3–16.
11	PLLDIV†	PLL divider. PLLDIV defines the frequency multiplier in conjunction with PLLMUL and PLLNDIV. See Table 3–16. PLLDIV = 0 Means that an integer multiply factor is used PLLDIV = 1 Means that a noninteger multiply factor is used
10–3	PLLCOUNT†	PLL counter value. PLLCOUNT specifies the number of input clock cycles (in increments of 16 cycles) for the PLL lock timer to count before the PLL begins clocking the processor after the PLL is started. The PLL counter is a down-counter, which is driven by the input clock divided by 16; therefore, for every 16 input clocks, the PLL counter decrements by one. The PLL counter can be used to ensure that the processor is not clocked until the PLL is locked, so that only valid clock signals are sent to the device.
2	PLLON/OFF†	PLL on/off. PLLON/OFF enables or disables the PLL part of the clock generator in conjunction with the PLLNDIV bit (see Table 3–17). Note that PLLON/OFF and PLLNDIV can both force the PLL to run; when PLLON/OFF is high, the PLL runs independently of the state of PLLNDIV.
1	PLLNDIV	PLLNDIV configures PLL mode when high or DIV mode when low. PLLNDIV defines the frequency multiplier in conjunction with PLLDIV and PLLMUL. See Table 3–16.
0	STATUS	Indicates the PLL mode. STATUS = 0 Indicates DIV mode STATUS = 1 Indicates PLL mode

† When in DIV mode (PLLSTATUS is low), PLLMUL, PLLDIV, PLLCOUNT, and PLLON/OFF are don't cares, and their contents are indeterminate.

Table 3–16. Multiplier Related to PLLNDIV, PLLDIV, and PLLMUL

PLLNDIV	PLLDIV	PLLMUL	MULTIPLIER†
0	x	0–14	0.5
0	x	15	0.25
1	0	0–14	PLLMUL + 1
1	0	15	bypass (multiply by 1)‡
1	1	0 or even	(PLLMUL + 1)/2
1	1	odd	PLLMUL/4

† CLKOUT = CLKIN * Multiplier
‡ Indicates the default clock mode after reset

Table 3–17. VCO Truth Table

PLLON/OFF	PLLNDIV	VCO STATE
0	0	off
1	0	on
0	1	on
1	1	on

3.9.3 PLL Clock Programmable Timer

During the lockup period, the PLL should not be used to clock the 5421. The PLLCOUNT programmable lock timer provides a convenient method of automatically delaying clocking of the device by the PLL until lock is achieved.

The PLL lock timer is a counter, loaded from the PLLCOUNT field in the CLKMD register, that decrements from its preset value to 0. The timer can be preset to any value from 0 to 255, and its input clock is CLKIN divided by 16. The resulting lockup delay can therefore be set from 0 to 255 $\times$ 16 CLKIN cycles.

The lock timer is activated when the operating mode of the clock generator is switched from DIV to PLL. During the lockup period, the clock generator continues to operate in DIV mode; after the PLL lock timer decrements to zero, the PLL begins clocking the 5421.

Accordingly, the value loaded into PLLCOUNT is chosen based on the following formula:

$$PLLCOUNT = \frac{Lockup\ Time}{16 \times T_{CLKIN}}$$

where T_{CLKIN} is the input reference clock period and lockup time is the required VCO lockup time, as shown in Table 3–18.

Table 3–18. VCO Lockup Time

CLKOUT FREQUENCY (MHz)	LOCKUP TIME (μs)
5	23
10	17
20	16
40	19
60	24
80	29
100	35

3.10 Memory-Mapped Registers

The 5421 has 27 memory-mapped CPU registers, which are mapped in data memory space address 0h to 1Fh. Each 5421 device also has a set of memory-mapped registers associated with peripherals. Table 3–19 gives a list of CPU memory-mapped registers (MMRs) available. Table 3–20 shows additional peripheral MMRs associated with the 5421.

Table 3–19. Processor Memory-Mapped Registers for Each DSP Subsystem

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
IMR	0	0	Interrupt Mask Register
IFR	1	1	Interrupt Flag Register
—	2–5	2–5	Reserved for testing
ST0	6	6	Status Register 0
ST1	7	7	Status Register 1
AL	8	8	Accumulator A Low Word (15–0)
AH	9	9	Accumulator A High Word (31–16)
AG	10	A	Accumulator A Guard Bits (39–32)
BL	11	B	Accumulator B Low Word (15–0)
BH	12	C	Accumulator B High Word (31–16)
BG	13	D	Accumulator B Guard Bits (39–32)
TREG	14	E	Temporary Register
TRN	15	F	Transition Register
AR0	16	10	Auxiliary Register 0
AR1	17	11	Auxiliary Register 1
AR2	18	12	Auxiliary Register 2
AR3	19	13	Auxiliary Register 3
AR4	20	14	Auxiliary Register 4
AR5	21	15	Auxiliary Register 5
AR6	22	16	Auxiliary Register 6
AR7	23	17	Auxiliary Register 7
SP	24	18	Stack Pointer
BK	25	19	Circular Buffer Size Register
BRC	26	1A	Block-Repeat Counter
RSA	27	1B	Block-Repeat Start Address
REA	28	1C	Block-Repeat End Address
PMST	29	1D	Processor Mode Status Register
XPC	30	1E	Extended Program Counter
—	31	1F	Reserved

Table 3–20. Peripheral Memory-Mapped Registers for Each DSP Subsystem

NAME	ADDRESS DEC	ADDRESS HEX	DESCRIPTION
DRR20	32	20	McBSP 0 Data Receive Register 2
DRR10	33	21	McBSP 0 Data Receive Register 1
DXR20	34	22	McBSP 0 Data Transmit Register 2
DXR10	35	23	McBSP 0 Data Transmit Register 1
TIM	36	24	Timer Register
PRD	37	25	Timer Period Register
TCR	38	26	Timer Control Register
—	39	27	Reserved
SWWSR	40	28	Software Wait-State Register
BSCR	41	29	Bank-Switching Control Register
—	42	2A	Reserved
SWCR	43	2B	Software Wait-State Control Register
HPIC	44	2C	HPI Control Register (HMODE=0 only)
—	45–47	2D–2F	Reserved
DRR22	48	30	McBSP 2 Data Receive Register 2
DRR12	49	31	McBSP 2 Data Receive Register 1
DXR22	50	32	McBSP 2 Data Transmit Register 2
DXR12	51	33	McBSP 2 Data Transmit Register 1
SPSA2	52	34	McBSP 2 Subbank Address Register†
SPSD2	53	35	McBSP 2 Subbank Data Register†
—	54–55	36–37	Reserved
SPSA0	56	38	McBSP 0 Subbank Address Register†
SPSD0	57	39	McBSP 0 Subbank Data Register†
—	58–59	3A–3B	Reserved
GPIO	60	3C	General-Purpose I/O Register
—	61	3D	Reserved
CSIDR	62	3E	Chip Subsystem ID register
—	63	3F	Reserved
DRR21	64	40	McBSP 1 Data Receive Register 2
DRR11	65	41	McBSP 1 Data Receive Register 1
DXR21	66	42	McBSP 1 Data Transmit Register 2
DXR11	67	43	McBSP 1 Data Transmit Register 1
—	68–71	44–47	Reserved
SPSA1	72	48	McBSP 1 Subbank Address Register†
SPSD1	73	49	McBSP 1 Subbank Data Register†
—	74–83	4A–53	Reserved
DMPREC	84	54	DMA Priority and Enable Control Register
DMSA	85	55	DMA Subbank Address Register‡
DMSDI	86	56	DMA Subbank Data Register with Autoincrement‡
DMSDN	87	57	DMA Subbank Data Register‡
CLKMD	88	58	Clock Mode Register (CLKMD)
—	89–95	59–5F	Reserved

† See Table 3–21 for a detailed description of the McBSP control registers and their subaddresses.
‡ See Table 3–22 for a detailed description of the DMA subbank addressed registers.

3.11 McBSP Control Registers and Subaddresses

The control registers for the multichannel buffered serial port (McBSP) are accessed using the subbank addressing scheme. This allows a set or subbank of registers to be accessed through a single memory location. The McBSP subbank address register (SPSA) is used as a pointer to select a particular register within the subbank. The McBSP data register (SPSDx) is used to access (read or write) the selected register. Table 3–21 shows the McBSP control registers and their corresponding subaddresses.

Table 3–21. McBSP Control Registers and Subaddresses

McBSP0		McBSP1		McBSP2		SUB-ADDRESS	DESCRIPTION
NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS		
SPCR10	39h	SPCR11	49h	SPCR12	35h	00h	Serial port control register 1
SPCR20	39h	SPCR21	49h	SPCR22	35h	01h	Serial port control register 2
RCR10	39h	RCR11	49h	RCR12	35h	02h	Receive control register 1
RCR20	39h	RCR21	49h	RCR22	35h	03h	Receive control register 2
XCR10	39h	XCR11	49h	XCR12	35h	04h	Transmit control register 1
XCR20	39h	XCR21	49h	XCR22	35h	05h	Transmit control register 2
SRGR10	39h	SRGR11	49h	SRGR12	35h	06h	Sample rate generator register 1
SRGR20	39h	SRGR21	49h	SRGR22	35h	07h	Sample rate generator register 2
MCR10	39h	MCR11	49h	MCR12	35h	08h	Multichannel register 1
MCR20	39h	MCR21	49h	MCR22	35h	09h	Multichannel register 2
RCERA0	39h	RCERA1	49h	RCERA2	35h	0Ah	Receive channel enable register partition A
RCERB0	39h	RCERB1	49h	RCERB2	35h	0Bh	Receive channel enable register partition B
XCERA0	39h	XCERA1	49h	XCERA2	35h	0Ch	Transmit channel enable register partition A
XCERB0	39h	XCERB1	49h	XCERB2	35h	0Dh	Transmit channel enable register partition B
PCR0	39h	PCR1	49h	PCR2	35h	0Eh	Pin control register
RCERC0	39h	RCERC1	49h	RCERC2	35h	010h	Receive channel enable register partition C
RCERD0	39h	RCERD1	49h	RCERD2	35h	011h	Receive channel enable register partition D
XCERC0	39h	XCERC1	49h	XCERC2	35h	012h	Transmit channel enable register partition C
XCERD0	39h	XCERD1	49h	XCERD2	35h	013h	Transmit channel enable register partition D
RCERE0	39h	RCERE1	49h	RCERE2	35h	014h	Receive channel enable register partition E
RCERF0	39h	RCERF1	49h	RCERF2	35h	015h	Receive channel enable register partition F
XCERE0	39h	XCERE1	49h	XCERE2	35h	016h	Transmit channel enable register partition E
XCERF0	39h	XCERF1	49h	XCERF2	35h	017h	Transmit channel enable register partition F
RCERG0	39h	RCERG1	49h	RCERG2	35h	018h	Receive channel enable register partition G
RCERH0	39h	RCERH1	49h	RCERH2	35h	019h	Receive channel enable register partition H
XCERG0	39h	XCERG1	49h	XCERG2	35h	01Ah	Transmit channel enable register partition G
XCERH0	39h	XCERH1	49h	XCERH2	35h	01Bh	Transmit channel enable register partition H

3.12 DMA Subbank Addressed Registers

The direct memory access (DMA) controller has several control registers associated with it. The main control register (DMPREC) is a standard memory-mapped register. However, the other registers are accessed using the subbank addressing scheme. This allows a set or subbank of registers to be accessed through a single memory location. The DMA subbank address (DMSA) register is used as a pointer to select a particular register within the subbank, while the DMA subbank data (DMSD) register or the DMA subbank data register with autoincrement (DMSDI) is used to access (read or write) the selected register.

When the DMSDI register is used to access the subbank, the subbank address is automatically postincremented so that a subsequent access affects the next register within the subbank. This autoincrement feature is intended for efficient, successive accesses to several control registers. If the autoincrement feature is not required, the DMSDN register should be used to access the subbank. Table 3–22 shows the DMA controller subbank addressed registers and their corresponding subaddresses.

Table 3–22. DMA Subbank Addressed Registers

NAME	ADDRESS	SUB-ADDRESS	DESCRIPTION
DMSRC0	56h/57h	00h	DMA channel 0 source address register
DMDST0	56h/57h	01h	DMA channel 0 destination address register
DMCTR0	56h/57h	02h	DMA channel 0 element count register
DMSFC0	56h/57h	03h	DMA channel 0 sync event and frame count register
DMMCR0	56h/57h	04h	DMA channel 0 transfer mode control register
DMSRC1	56h/57h	05h	DMA channel 1 source address register
DMDST1	56h/57h	06h	DMA channel 1 destination address register
DMCTR1	56h/57h	07h	DMA channel 1 element count register
DMSFC1	56h/57h	08h	DMA channel 1 sync event and frame count register
DMMCR1	56h/57h	09h	DMA channel 1 transfer mode control register
DMSRC2	56h/57h	0Ah	DMA channel 2 source address register
DMDST2	56h/57h	0Bh	DMA channel 2 destination address register
DMCTR2	56h/57h	0Ch	DMA channel 2 element count register
DMSFC2	56h/57h	0Dh	DMA channel 2 sync event and frame count register
DMMCR2	56h/57h	0Eh	DMA channel 2 transfer mode control register
DMSRC3	56h/57h	0Fh	DMA channel 3 source address register
DMDST3	56h/57h	10h	DMA channel 3 destination address register
DMCTR3	56h/57h	11h	DMA channel 3 element count register
DMSFC3	56h/57h	12h	DMA channel 3 sync event and frame count register
DMMCR3	56h/57h	13h	DMA channel 3 transfer mode control register
DMSRC4	56h/57h	14h	DMA channel 4 source address register
DMDST4	56h/57h	15h	DMA channel 4 destination address register
DMCTR4	56h/57h	16h	DMA channel 4 element count register
DMSFC4	56h/57h	17h	DMA channel 4 sync event and frame count register
DMMCR4	56h/57h	18h	DMA channel 4 transfer mode control register
DMSRC5	56h/57h	19h	DMA channel 5 source address register
DMDST5	56h/57h	1Ah	DMA channel 5 destination address register
DMCTR5	56h/57h	1Bh	DMA channel 5 element count register
DMSFC5	56h/57h	1Ch	DMA channel 5 sync event and frame count register
DMMCR5	56h/57h	1Dh	DMA channel 5 transfer mode control register
DMSRCP	56h/57h	1Eh	DMA source program page address (common channel)

Table 3–22. DMA Subbank Addressed Registers (Continued)

NAME	ADDRESS	SUB-ADDRESS	DESCRIPTION
DMDSTP	56h/57h	1Fh	DMA destination program page address (common channel)
DMIDX0	56h/57h	20h	DMA element index address register 0
DMIDX1	56h/57h	21h	DMA element index address register 1
DMFRI0	56h/57h	22h	DMA frame index register 0
DMFRI1	56h/57h	23h	DMA frame index register 1
DMGSA0	56h/57h	24h	DMA channel 0 global source address reload register
DMGDA0	56h/57h	25h	DMA channel 0 global destination address reload register
DMGCR0	56h/57h	26h	DMA channel 0 global count reload register
DMGFR0	56h/57h	27h	DMA channel 0 global frame count reload register
XSRCDP	56h/57h	28h	DMA extended source data page
XDSTDP	56h/57h	29h	DMA extended destination data page
DMGSA1	56h/57h	2Ah	DMA channel 1 global source address reload register
DMGDA1	56h/57h	2Bh	DMA channel 1 global destination address reload register
DMGCR1	56h/57h	2Ch	DMA channel 1 global count reload register
DMGFR1	56h/57h	2Dh	DMA channel 1 global frame count reload register
DMGSA2	56h/57h	2Eh	DMA channel 2 global source address reload register
DMGDA2	56h/57h	2Fh	DMA channel 2 global destination address reload register
DMGCR2	56h/57h	30h	DMA channel 2 global count reload register
DMGFR2	56h/57h	31h	DMA channel 2 global frame count reload register
DMGSA3	56h/57h	32h	DMA channel 3 global source address reload register
DMGDA3	56h/57h	33h	DMA channel 3 global destination address reload register
DMGCR3	56h/57h	34h	DMA channel 3 global count reload register
DMGFR3	56h/57h	35h	DMA channel 3 global frame count reload register
DMGSA4	56h/57h	36h	DMA channel 4 global source address reload register
DMGDA4	56h/57h	37h	DMA channel 4 global destination address reload register
DMGCR4	56h/57h	38h	DMA channel 4 global count reload register
DMGFR4	56h/57h	39h	DMA channel 4 global frame count reload register
DMGSA5	56h/57h	3Ah	DMA channel 5 global source address reload register
DMGDA5	56h/57h	3Bh	DMA channel 5 global destination address reload register
DMGCR5	56h/57h	3Ch	DMA channel 5 global count reload register
DMGFR5	56h/57h	3Dh	DMA channel 5 global frame count reload register

3.13 Interrupts

Vector-relative locations and priorities for all internal and external interrupts are shown in Table 3–23.

Table 3–23. 5421 Interrupt Locations and Priorities for Each DSP Subsystem

NAME	LOCATION		PRIORITY	FUNCTION
	DECIMAL	HEX		
$\overline{RS}$, SINTR	0	00	1	Reset (Hardware and Software Reset)
$\overline{NMI}$, SINT16	4	04	2	Nonmaskable Interrupt
SINT17	8	08	—	Software Interrupt #17
SINT18	12	0C	—	Software Interrupt #18
SINT19	16	10	—	Software Interrupt #19
SINT20	20	14	—	Software Interrupt #20
SINT21	24	18	—	Software Interrupt #21
SINT22	28	1C	—	Software Interrupt #22
SINT23	32	20	—	Software Interrupt #23
SINT24	36	24	—	Software Interrupt #24
SINT25	40	28	—	Software Interrupt #25
SINT26	44	2C	—	Software Interrupt #26
SINT27	48	30	—	Software Interrupt #27
SINT28	52	34	—	Software Interrupt #28
SINT29	56	38	—	Software Interrupt #29
SINT30	60	3C	—	Software Interrupt #30
$\overline{INT0}$, SINT0	64	40	3	External User Interrupt #0
$\overline{INT1}$, SINT1	68	44	4	External User Interrupt #1
$\overline{INT2}$, SINT2	72	48	5	Reserved
TINT, SINT3	76	4C	6	External Timer Interrupt
BRINT0, SINT4	80	50	7	BSP #0 Receive Interrupt
BXINT0, SINT5	84	54	8	BSP #0 Transmit Interrupt
BRINT2, DMAC0	88	58	9	BSP #2 Receive Interrupt or DMA Channel 0
BXINT2, DMAC1	92	5C	10	BSP #2 Receive Interrupt or DMA Channel 1
$\overline{INT3}$, SINT8	96	60	11	Reserved
HPINT, SINT9	100	64	12	HPI Interrupt (from DSPINT in HPIC)
BRINT1, DMAC2	104	68	13	BSP #1 Receive Interrupt or DMA Channel 2
BXINT1, DMAC3	108	6C	14	BSP #1 transmit Interrupt or DMA channel 3
DMAC4, SINT12	112	70	15	DMA Channel 4
DMAC5, SINT13	116	74	16	DMA Channel 5
IPINT, SINT14	120	78	17	Interprocessor Interrupt
—	124–127	7C–7F	—	Reserved

The interprocessor interrupt (IPINT) bit of the interrupt mask register (IMR) and the interrupt flag register (IFR) allows the subsystem to perform interrupt service routines based on the other subsystem activity. Incoming IPINT interrupts are latched in IFR.14. Generating an interprocessor interrupt is performed by writing a “1” to the IPIRQ field of the bank-switching control register (BSCR). Subsequent interrupts must first clear the interrupt by writing “0” to the IPIRQ field. Figure 3–21 shows the bit layout of the IMR and the IFR. Table 3–24 describes the bit functions.

For example, if subsystem A is required to notify subsystem B of a completed task, subsystem A must write a “1” to the IPIRQ field to generate a IPINT interrupt on subsystem B. On subsystem B, the IPINT interrupt is latched in IFR.14. Figure 5 shows the bit layout of the BSCR and Table 6 describes the bit functions.

15	14	13	12	11	10	9	8
Reserved	IPINT	DMAC5	DMAC4	XINT1 or DMAC3	RINT1 or DMAC2	HPINT	Reserved
	R/W		R/W	R/W	R/W	R/W	

7	6	5	4	3	2	1	0
XINT2 or DMAC1	RINT2 or DMAC0	XINT0	RINT0	TINT	Reserved	INT1	INT0
R/W	R/W	R/W	R/W	R/W		R/W	R/W

LEGEND: R = Read, W = Write

Figure 3–21. Bit Layout of the IMR and IFR Registers for Subsystems A and B

Table 3–24. Bit Functions for IMR and IFR Registers for Each DSP Subsystem

BIT NO.	BIT NAME	BIT VALUE	FUNCTION
15	Reserved	X	Register bit is reserved.
14	IPINT	0	IFR/IMR: Interprocessor IRQ has no interrupt pending/is disabled (masked).
		1	IFR/IMR: Interprocessor IRQ has an interrupt pending/is enabled.
13	DMAC5	0	IFR/IMR: DMA Channel 5 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: DMA Channel 5 has an interrupt pending/is enabled.
12	DMAC4	0	IFR/IMR: DMA Channel 4 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: DMA Channel 4 has an interrupt pending/is enabled.
11	XINT1	0	IFR/IMR: McBSP_1 has no transmit interrupt pending/is disabled (masked).
		1	IFR/IMR: McBSP_1 has a transmit interrupt pending/is enabled.
	DMAC3	0	IFR/IMR: DMA Channel 3 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: DMA Channel 3 has an interrupt pending/is enabled.
10	RINT1	0	IFR/IMR: McBSP_1 has no receive interrupt pending/is disabled (masked).
		1	IFR/IMR: McBSP_1 has a receive interrupt pending/is enabled.
	DMAC2	0	IFR/IMR: DMA Channel 2 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: DMA Channel 2 has an interrupt pending/is enabled.
9	HPINT	0	IFR/IMR: Host-port interface has no DSPINT interrupt pending/is disabled (masked).
		1	IFR/IMR: Host-port interface has an DSPINT interrupt pending/is enabled.
8	Reserved	X	Register bit is reserved.
7	XINT2	0	IFR/IMR: McBSP_2 has no transmit interrupt pending/is disabled (masked).
		1	IFR/IMR: McBSP_2 has a transmit interrupt pending/is enabled.
	DMAC1	0	IFR/IMR: DMA Channel 1 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: DMA Channel 1 has an interrupt pending/is enabled.
6	RINT2	0	IFR/IMR: McBSP_2 has no receive interrupt pending/is disabled (masked).
		1	IFR/IMR: McBSP_2 has a receive interrupt pending/is enabled.
	DMAC0	0	IFR/IMR: DMA Channel 0 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: DMA Channel 0 has an interrupt pending/is enabled.
5	XINT0	0	IFR/IMR: McBSP_0 has no receive interrupt pending/is disabled (masked).
		1	IFR/IMR: McBSP_0 has a receive interrupt pending/is enabled.
4	RINT0	0	IFR/IMR: McBSP_0 has no receive interrupt pending/is disabled (masked).
		1	IFR/IMR: McBSP_0 has a receive interrupt pending/is enabled.

Table 3–24. Bit Functions for IMR and IFR Registers for Each DSP Subsystem (Continued)

BIT NO.	BIT NAME	BIT VALUE	FUNCTION
3	TINT	0	IFR/IMR: Timer has no interrupt pending/is disabled (masked).
		1	IFR/IMR: Timer has an interrupt pending/is enabled.
2	Reserved	X	Register bit is reserved.
1	INT1	0	IFR/IMR: Ext user interrupt pin 1 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: Ext user interrupt pin 1 has an interrupt pending/is enabled.
0	INT0	0	IFR/IMR: Ext user interrupt pin 0 has no interrupt pending/is disabled (masked).
		1	IFR/IMR: Ext user interrupt pin 0 has an interrupt pending/is enabled.

3.14 IDLE3 Power-Down Mode

The IDLE1 and IDLE2 power-down modes operate as described in the *TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals* (literature number SPRU131). The IDLE3 mode is special in that the clocking circuitry is shut off to conserve power. The 5421 cannot enter an IDLE3 mode unless both subsystems execute an IDLE3 instruction. The power-reduced benefits of IDLE3 cannot be realized until both subsystems enter the IDLE3 state and the internal clocks are automatically shut off. The order in which subsystems enter IDLE3 does not matter.

3.15 Emulating the 5421 Device

The 5421 is a single device, but actually consists of two independent subboundary systems that contain register/status information used by the emulator tools. The emulator tools must be informed of the multicore device by modifying the **board.cfg** file. The board.cfg file is an ASCII file that can be modified with most editors. This provides the emulator with a description of the JTAG chain. The board.cfg file must identify two processors when using the 5421. The file contents would look something like this:

“CPU_B” TI320C5xx

“CPU_A” TI320C5xx

Use Code Composer Studio to convert this file into a binary file (**board.dat**), readable by the emulation tools. Place the board.dat file in the directory that contains the emulator software.

The subsystems are serially connected together internally. Emulation information is serially transmitted into the device using the TDI pin. The device responds with serial scan information transmitted out the TDO pin.