OMAP5912 Multimedia Processor Peripheral Interconnects Reference Guide

Literature Number: SPRU758A
March 2004

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Preface

Read This First

About This Manual

This document describes various peripheral interconnects of the OMAP5912 multimedia processor.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.

Related Documentation From Texas Instruments

The following documents describe the OMAP5910 device and related peripherals. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

OMAP5912 Multimedia Processor Device Overview and Architecture Reference Guide (literature number SPRU748) introduces the setup, components, and features of the OMAP5912 multimedia processor and provides a high-level view of the device architecture.

OMAP5912 Multimedia Processor OMAP 3.2 Subsystem Reference Guide (literature number SPRU749) introduces and briefly defines the main features of the OMAP3.2 subsystem of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor DSP Sybsystem Reference Guide (literature number SPRU750) describes the OMAP5912 multimedia processor DSP subsystem. The digital signal processor (DSP) subsystem is built around a core processor and peripherals that interface with: 1) The ARM926EJS via the microprocessor unit interface (MPUI); 2) Various standard memories via the external memory interface (EMIF); 3) Various system peripherals via the TI peripheral bus (TIPB) bridge.

OMAP5912 Multimedia Processor Clocks Reference Guide (literature number SPRU751) describes the clocking mechanisms of the OMAP5912 multimedia processor. In OMAP5912, various clocks are created from special components such as the digital phase locked loop (DPLL) and the analog phase-locked loop (APLL).

OMAP5912 Multimedia Processor Initialization Reference Guide (literature number SPRU752) describes the reset architecture, the configuration, the initialization, and the boot ROM of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Power Management Reference Guide (literature number SPRU753) describes power management in the OMAP5912 multimedia processor. The ultralow-power device (ULPD) generates and manages clocks and reset signals to OMAP3.2 and to some peripherals. It controls chip-level power-down modes and handles chip-level wake-up events. In deep sleep mode, this module is still active to monitor wake-up events.This book describes the ULPD module and outline architecture.

OMAP5912 Multimedia Processor Security Features Reference Guide (literature number SPRU754) describes the security features of teh OMAP5912 multimedia processor. The OMAP5912 security scheme relies on the OMAP3.2 secure mode. The distributed security on the OMAP3.2 platform is a Texas Instruments solution to address m-commerce and security issues within a mobile phone environment. The OMAP3.2 secure mode was developed to bring hardware robustness to the overall OMAP5912 security scheme.

OMAP5912 Multimedia Processor Direct Memory Access (DMA) Support Reference Guide (literature number SPRU755) describes the direct memory access support of the OMAP5912 multimedia processor. The OMAP5912 processor has three DMAs:

- The system DMA is embedded in OMAP3.2. It handles DMA transfers associated with MPU and shared peripherals.
- The DSP DMA is embedded in OMAP3.2. It handles DMA transfers associated with DSP peripherals.
- The generic distributed DMA (GDD) is an OMAP5912 resource attached to the SSI peripheral. It handles only DMA transfers associated with the SSI peripheral.

OMAP5912 Multimedia Processor Memory Interfaces Reference Guide (literature number SPRU756) describes the memory interfaces of the OMAP5912 multimedia processor.

- SDRAM (external memory interface fast, or EMIFF)
- Asynchronous and synchronous burst memory (external memory interface slow, or EMIFS)
- NAND flash (hardware controller or software controller)
- CompactFlash on EMIFS interface
- Internal static RAM

OMAP5912 Multimedia Processor Interrupts Reference Guide (literature number SPRU757) describes the interrupts of the OMAP5912 multimedia processor. Three level 2 interrupt controllers are used in OMAP5912:

- One MPU level 2 interrupt handler (also referred to as MPU interrupt level 2) is implemented outside of OMAP3.2 and can handle 128 interrupts.
- One DSP level 2 interrupt handler (also referred to as DSP interrupt level 2.1) is instantiated outside of OMAP3.2 and can handle 64 interrupts.
- One OMAP3.2 DSP level 2 interrupt handler (referenced as DSP interrupt level 2.0) can handle 16 interrupts.

OMAP5912 Multimedia Processor Peripheral Interconnects Reference Guide (literature number SPRU758) describes various periperal interconnects of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Timers Reference Guide (literature number SPRU759) describes various timers of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Serial Interfaces Reference Guide (literature number SPRU760) describes the serial interfaces of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Universal Serial Bus (USB) Reference Guide (literature number SPRU761) describes the universal serial bus (USB) host on the OMAP5912 multimedia processor. The OMAP5912 processor provides several varieties of USB functionality. Flexible multiplexing of signals from the OMAP5912 USB host controller, the OMAP5912 USB function controller, and other OMAP5912 peripherals allow a wide variety of system-level USB capabilities. Many of the OMAP5912 pins can be used for USB-related signals or for signals from other OMAP5912 peripherals. The OMAP5912 top-level pin multiplexing

controls each pin individually to select one of several possible internal pin signal interconnections. When these shared pins are programmed for use as USB signals, the OMAP5912 USB signal multiplexing selects how the signals associated with the three OMAP5912 USB host ports and the OMAP5912 USB function controller can be brought out to OMAP5912 pins.

OMAP5912 Multimedia Processor Multi-channel Buffered Serial Ports (McBSPs) Reference Guide (literature number SPRU762) describes the three multi-channel buffered serial ports (McBSPs) available on the OMAP5912 device. The OMAP5912 device provides multiple high-speed multichannel buffered serial ports (McBSPs) that allow direct interface to codecs and other devices in a system.

OMAP5912 Multimedia Processor Camera Interface Reference Guide (literature number SPRU763) describes two camera inerfaces implemented in the OMAP5912 multimedia processor: compact serial camera port and camera parallel interface.

OMAP5912 Multimedia Processor Display Interface Reference Guide (literature number SPRU764) describes the display interface of the OMAP5912 multimedia processor.

- LCD module
- LCD data conversion module
- LED pulse generator
- Display interface

OMAP5912 Multimedia Processor Multimedia Card (MMC/SD/SDIO) (literature number SPRU765) describes the multimedia card (MMC) interface of the OMAP5912 multimedia processor. The multimedia card/secure data/secure digital IO (MMC/SD/SDIO) host controller provides an interface between a local host, such as a microprocessor unit (MPU) or digital signal processor (DSP), and either an MMC or SD memory card, plus up to four serial flash cards. The host controller handles MMC/SD/SDIO or serial port interface (SPI) transactions with minimal local host intervention.

OMAP5912 Multimedia Processor Keyboard Interface Reference Guide (literature number SPRU766) describes the keyboard interface of the OMAP5912 multimedia processor. The MPUIO module enables direct I/O communication between the MPU (through the public TIPB) and external devices. Two types of I/O can be used: specific I/Os dedicated for 8 x 8 keyboard connection, and general-purpose I/Os.

OMAP5912 Multimedia Processor General-Purpose Interface Reference Guide (literature number SPRU767) describes the general-purpose in-

terface of the OMAP5912 multimedia processor. There are four GPIO modules in the OMAP5912. Each GPIO peripheral controls 16 dedicated pins configurable either as input or output for general purposes. Each pin has an independent control direction set by a programmable register. The two−edge control registers configure events (rising edge, falling edge, or both edges) on an input pin to trigger interrupts or wake−up requests (depending on the system mode). In addition, an interrupt mask register masks out specified pins. Finally, the GPIO peripherals provide the set and clear capabilities on the data output registers and the interrupt mask registers. After detection, all event sources are merged and a single synchronous interrupt (per module) is generated in active mode, whereas a unique wake−up line is issued in idle mode. Eight data output lines of the GPIO3 are ORed together to generate a global output line at the OMAP5912 boundary. This global output line can be used in conjunction with the SSI to provide a CMT−APE interface to the OMAP5912.

OMAP5912 Multimedia Processor VLYNQ Serial Communications Interface Reference Guide (literature number SPRU768) describes the VLYNQ of the OMAP5912 multimedia processor.

VLYNQ is a serial communications interface that enables the extension of an internal bus segment to one or more external physical devices. The external devices are mapped into local, physical address space and appear as if they are on the internal bus of the OMAP 5912. The external devices must also have a VLYNQ interface. The VLYNQ module serializes bus transactions in one device, transfers the serialized data between devices via a VLYNQ port, and de-serializes the transaction in the external device.

OMAP5912 includes one VLYNQ module connected on OCPT2 target port and OCPI initiator port. These connections are configured via a static switch, which selects either SSI or VLYNQ module. This switch, forbids the simultaneous use of GDD/SSI and VLYNQ. The switch is controlled by the VLYNQ_EN bit in the OMAP5912 configuration control register (CONF_5912_CTRL).

OMAP5912 Multimedia Processor Pinout Reference Guide (literature number SPRU769) provides the pinout of the OMAP5912 multimedia processor. After power-up reset, the user can change the configuration of the default interfaces. If another interface is available on top of the default, it is possible to enable a new interface for each ball by setting the corresponding 3-bit field of the associated FUNC_MUX_CTRL register. It is also possible to configure on-chip pullup/pulldown. This document

also describes the various power domains so that the user can apply the different interfaces seamlessly with external components.

OMAP5912 Multimedia Processor Window Tracer (WT) Reference Guide (literature number SPRU770) describes the window tracer module used to capture the memory transactions from four interfaces: EMIFF, EMIFS, OCP-T1, and OCP-T2. This module is located in the OMAP3.2 traffic controller (TC).

OMAP5912 Multimedia Processor Real-Time Clock Reference Guide (literature number SPRUxxx) describes the real-time clock of the OMAP5912 multimedia processor. The real-time clock (RTC) block is an embedded real-time clock module directly accessible from the TIPB bus interface.

Trademarks

Contents

Figures

Tables

Peripheral Interconnects

This document describes various peripheral interconnects of the OMAP5912 multimedia processor.

1 Shared Peripherals

Outside OMAP 3.2, the OMAP5912 includes several peripherals. They can be considered shared by the MPU and the DSP, or can be DSP or MPU private.

Privacy is mainly driven by the performance requested by the peripheral. OMAP provides four peripheral buses:

- 16-bit data bus DSP private
- 16-bit data bus DSP shared
- 32-bit data bus MPU private
- 32-bit data bus MPU shared

OMAP supports one bus protocol: the TIPB protocol.

The TIPB protocol enables unidirectional asynchronous transfer from or to peripherals to or from CPUs/DMAs.

Both shared buses ((MPU + system DMA) and (DSP + DSPDMA)) are multiplexed in the L4 interface either dynamically or semistatically to communicate with all the shared peripherals.

The L4 interface groups a set of wrappers (or protocol converter), a dynamic switch, and a static switch. They enable interface with all the peripherals, regardless of their interface protocol with the MPU and DSP peripheral buses.

The DSP peripherals are also accessible from the MPU via the DSP MPUI port (see Figure 1).

Figure 1. Peripherals Access

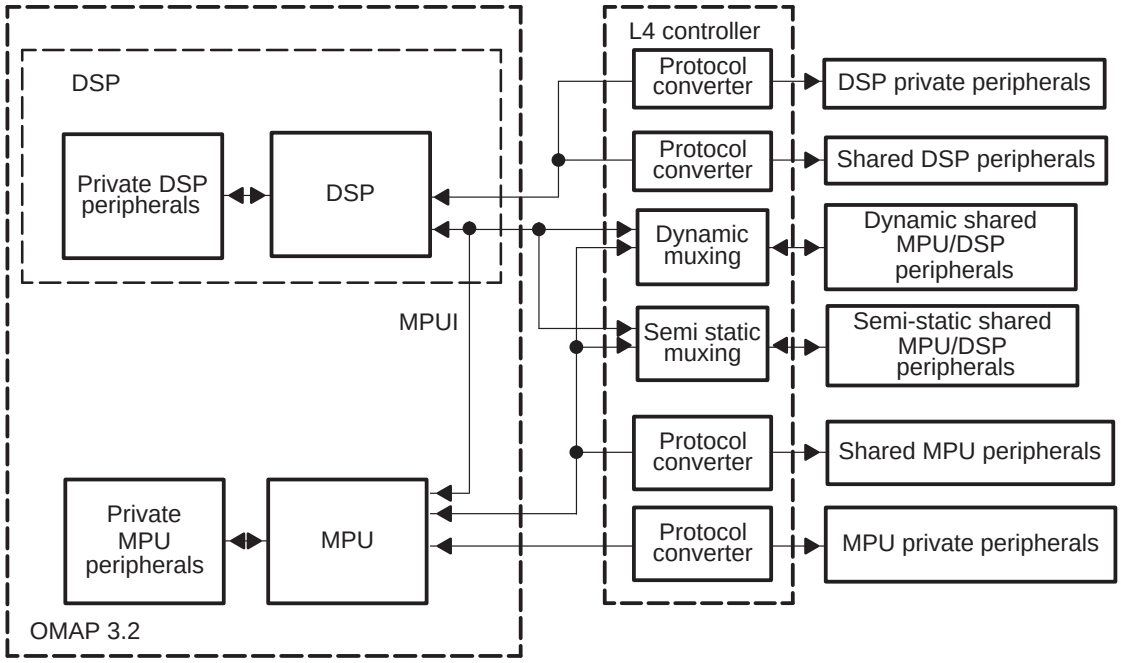


Table 1 lists the following for each module:

- On which bus the module resides
- Whether a DMA can initiate a transaction to the peripheral
- Whether the module can be accessed dynamically by each processor or the corresponding static switch must be configured

Table 1. MPU/DSP Peripheral Access

	MPU Domain			DSP Domain			
Module Name	**MPU Start**	**MPU End**	**MPU TIPB Bus Type**	**L4 Controler Switch**	**DSP Start**	**DSP End**	**DSP TIPB Bus Type**
DPLL1	FFFE CF00	FFFE CFFF	Private				
CLKM	FFFE CE00	FFFE CEFF	Private				
ULPD	FFFE 0800	FFFE 0FFF	Private				

Note: The SSI and the GDD modules are on the L3-OCP2 port and thus are seen as part of memory port interface.

Table 1. MPU/DSP Peripheral Access (Continued)

MPU Domain				DSP Domain			
Module Name	**MPU Start**	**MPU End**	**MPU TIPB Bus Type**	**L4 Controler Switch**	**DSP Start**	**DSP End**	**DSP TIPB Bus Type**
OMAP 5912 configuration	FFFE 1000	FFFE 17FF	Private				
32-kHz synchronization counter	FFFB C400	FFFB C7FF	Shared	Dynamic	E101 C400	E101 C7FF	Shared
Secure watchdog	FFFE A800	FFFE AFFF	Private				
32-kHz watchdog	FFFE B000	FFFE B7FF	Private				
RTC	FFFB 4800	FFFB 4FFF	Shared				
RNG	FFFE 5000	FFFE 57FF	Private				
DES/3DES	FFFE 4000	FFFE 47FF	Private				
SHA-1/MD5	FFFE 4800	FFFE 4FFF	Private				
OMAP MPU timer1	FFFE C500	FFFE C5FF	Private				
OMAP MPU timer2	FFFE C600	FFFE C6FF	Private				
OMAP MPU timer3	FFFE C700	FFFE C7FF	Private				
OMAP watchdog timer	FFFE C800	FFFE C8FF	Private				
OS timer	FFFB 9000	FFFB 93FF	Shared				
L3 OCP initiator	FFFE C300	FFFE C3FF	Private				

Note: The SSI and the GDD modules are on the L3-OCP2 port and thus are seen as part of memory port interface.

Table 1. MPU/DSP Peripheral Access (Continued)

MPU Domain				DSP Domain			
Module Name	**MPU Start**	**MPU End**	**MPU TIPB Bus Type**	**L4 Controler Switch**	**DSP Start**	**DSP End**	**DSP TIPB Bus Type**
L3 OCP T1 interface	FFFE C100	FFFE C1FF	Private				
OMAP5912 TIPB switch	FFFB C800	FFFB CBFF	Shared	Semi-static	E101 C800	E101 CBFF	Shared
TIPB bridge (internal)	FFFE CA00	FFFE CAFF	Private				
TIPB bridge2 (external)	FFFE D300	FFFE D3FF	Private				
MPUI interface	FFFE C900	FFFE C9FF	Private				
DSP MMU	FFFE D200	FFFE D2FF	Private				
Traffic controller	FFFE CC00	FFFE CCFF	Private				
Gigacell mailbox	FFFB F000	FFFB F7FF	Shared	Semi-static	E101 F000	E101 F7FF	Shared
Mailbox	FFFC F000	FFFC F7FF	Shared				
DSP TIPB					E100 0000	E100 07FF	Shared
MGS3 MPUI control register					E102 0000	E102 0003	Shared
System DMA	FFFE D800	FFFE DFFF	Private				
System DMA	FFFE E000	FFFE E7FF	Private				
MPU level 1 interrupt handler	FFFE CB00	FFFE CBFF	Private				

Note: The SSI and the GDD modules are on the L3-OCP2 port and thus are seen as part of memory port interface.

Table 1. MPU/DSP Peripheral Access (Continued)

MPU Domain				DSP Domain			
Module Name	**MPU Start**	**MPU End**	**MPU TIPB Bus Type**	**L4 Controler Switch**	**DSP Start**	**DSP End**	**DSP TIPB Bus Type**
MPU level 2 interrupt handler	FFFE 0000	FFFE 07FF	Private				
MPUIO	FFFB 5000	FFFB 57FF	Shared				
GPIO1	FFFB E400	FFFB E7FF	Shared	Dynamic	E101 E400	E101 E7FF	Shared
GPIO2	FFFB EC00	FFFB EFFF	Shared	Dynamic	E101 EC00	E101 EFFF	Shared
GPIO3	FFFB B400	FFFB B7FF	Shared	Dynamic	E101 B400	E101 B7FF	Shared
GPIO4	FFFB BC00	FFFB BFFF	Shared	Dynamic	E101 BC00	E101 BFFF	Shared
PWL	FFFB 5800	FFFB 5FFF	Shared				
PWT	FFFB 6000	FFFB 67FF	Shared				
GP timer 1	FFFB 1400	FFFB 17FF	Shared	Semi-static	E101 1400	E101 17FF	Shared
GP timer 2	FFFB 1C00	FFFB 1FFF	Shared	Semi-static	E101 1C00	E101 1FFF	Shared
GP timer 3	FFFB 2400	FFFB 27FF	Shared	Semi-static	E101 2400	E101 27FF	Shared
GP timer 4	FFFB 2C00	FFFB 2FFF	Shared	Semi-static	E101 2C00	E101 2FFF	Shared
GP timer 5	FFFB 3400	FFFB 37FF	Shared	Semi-static	E101 3400	E101 37FF	Shared
GP timer 6	FFFB 3C00	FFFB 3FFF	Shared	Semi-static	E101 3C00	E101 3FFF	Shared
GP timer 7	FFFB 7400	FFFB 77FF	Shared	Semi-static	E101 7400	E101 77FF	Shared

Note: The SSI and the GDD modules are on the L3-OCP2 port and thus are seen as part of memory port interface.

Table 1. MPU/DSP Peripheral Access (Continued)

	MPU Domain				DSP Domain		
Module Name	**MPU Start**	**MPU End**	**MPU TIPB Bus Type**	**L4 Controler Switch**	**DSP Start**	**DSP End**	**DSP TIPB Bus Type**
GP timer 8	FFFB D400	FFFB D7FF	Shared	Semi-static	E101 D400	E101 D7FF	Shared
LCDCONV	FFFE 3000	FFFE 37FF	Private				
LCD controller	FFFE C000	FFFE C0FF	Private				
LPG1	FFFB D000	FFFB D3FF	Shared				
LPG2	FFFB D800	FFFB BFFF	Shared				
McBSP1					E101 1800	E101 1BFF	Shared
McBSP2	FFFB 1000	FFFB 13FF	Shared	Semi-static	E101 1000	E101 13FF	Shared
McBSP3					E101 7000	E101 73FF	Shared
MCSI1					E101 2800	E101 2BFF	Shared
MCSI2					E101 2000	E101 23FF	Shared
Memory Stick	FFFB 8000	FFFB 83FF	Shared				
MMC/SDIO 1	FFFB 7800	FFFB 7BFF	Shared				
MMC/SDIO 2	FFFB 7C00	FFFB 7FFF	Shared	Semi-static	E101 7C00	E101 7FFF	Shared
Compact flash controller	FFFE 2800	FFFE 2FFF	Private				
NAND flash controller	FFFB CC00	FFFB CFFF	Shared				

Note: The SSI and the GDD modules are on the L3-OCP2 port and thus are seen as part of memory port interface.

Table 1. MPU/DSP Peripheral Access (Continued)

MPU Domain				DSP Domain			
Module Name	**MPU Start**	**MPU End**	**MPU TIPB Bus Type**	**L4 Controler Switch**	**DSP Start**	**DSP End**	**DSP TIPB Bus Type**
SoSSI	FFFB AC00	FFFB AFFF	Shared				
SPI	FFFB 0C00	FFFB 0FFF	Shared	Semi-static	E101 0C00	E101 0FFF	Shared
UART1	FFFB 0000	FFFB 03FF	Shared	Semi-static	E101 0000	E101 07FF	Shared
UART2	FFFB 0800	FFFB 0BFF	Shared	Semi-static	E101 0800	E101 0BFF	Shared
UART3	FFFB 9800	FFFB 9BFF	Shared	Semi-static	E101 9800	E101 9BFF	Shared
FAC	FFFB A800	FFFB ABFF	Shared				
USB client	FFFB 4000	FFFB 43FF	Shared				
USB host	FFFB A000	FFFB A3FF	Shared				
USB OTG	FFFB 0400	FFFB 07FF	Shared				
I^2C multi-master	FFFB 3800	FFFB 3BFF	Shared	Semi-static	E101 3800	E101 3BFF	Shared
µWire	FFFB 3000	FFFB 33FF	Shared				
HDQ 1-Wire	FFFB C000	FFFB C3FF	Shared				
STI (reserved)	FFFE A000	FFFE A7FF	Private	Dynamic	E101 A400	E101 A7FF	Shared
OMAP5912 JTAG (test)	FFFE 5800	FFFE 5FFF	Private				
BCM (test)	FFFE 3800	FFFE 3FFF	Private				
Production ID (test)	FFFE 2000	FFFE 27FF	Private				

Note: The SSI and the GDD modules are on the L3-OCP2 port and thus are seen as part of memory port interface.

Table 1. MPU/DSP Peripheral Access (Continued)

MPU Domain					DSP Domain		
Module Name	**MPU Start**	**MPU End**	**MPU TIPB Bus Type**	**L4 Controler Switch**	**DSP Start**	**DSP End**	**DSP TIPB Bus Type**
Die ID (test)	FFFE 1800	FFFE 1FFF	Private				
Test block (PSA test)	FFFE D400	FFFE D4FF	Private				
DSP trace					E100 4000	E100 47FF	Private

Note: The SSI and the GDD modules are on the L3-OCP2 port and thus are seen as part of memory port interface.

2 Layer 4 Interconnect

The layer 4 interface manages accesses to OMAP5912 peripherals through the MPU shared TIPB bridge, DSP shared TIPB bridge, MPU private TIPB bridge, and DSP private TIPB bridge. DSP TIPB bridges are inside the C55x DSP. It is composed of a set of dynamic and static switches and wrappers dedicated to the access protocol of each peripheral. The MPU TIPB bridge is already a merge of MPU and system DMA signals for peripheral accesses. The DSP TIPB bridge is already a merge of DSP and system DMA signals for peripheral accesses (see Figure 2).

Two different switches are implemented:

- ❏ Static TIPB/OCP: OCP/TIPB switch (configuration register included)
- ❏ Dynamic TIPB/OCP: OCP/TIPB switch with resynchronization on MPU and DSP accesses

Two different wrappers are provided:

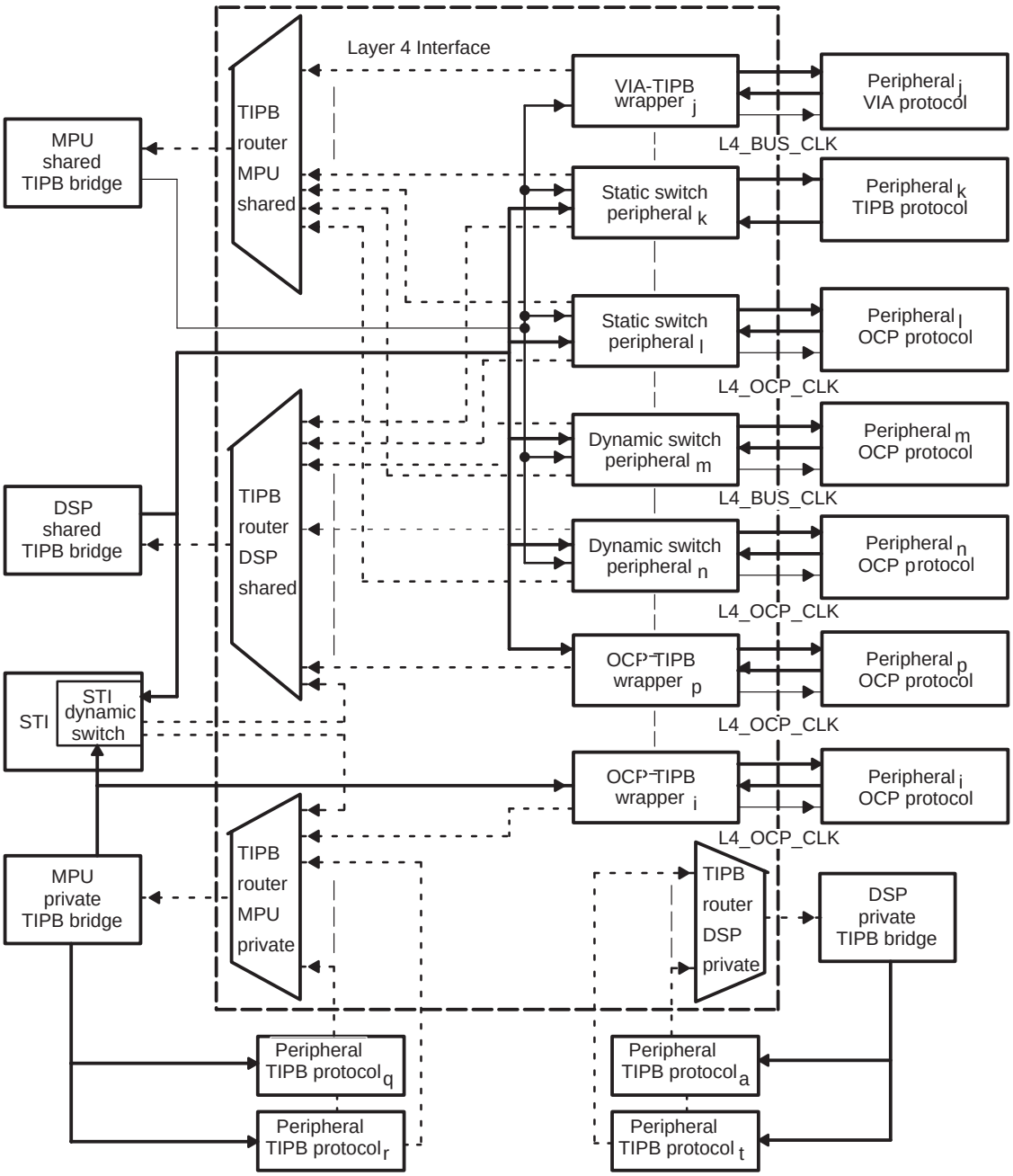
- ❏ TIPB/VIA: VIA/TIPB wrapper with resynchronization
- ❏ MPU (or DSP) TIPB/OCP: OCP/TIPB wrapper

The common protocol on the MPU/DSP side is the TIPB bus protocol. For each TIPB bridge, a dedicated router is implemented to allow the multiplexing of all the return paths from the peripherals.

- ❏ MPU shared TIPB router to MPU shared TIPB bridge
- ❏ DSP shared TIPB router to DSP shared TIPB bridge

- ❏ MPU private TIPB router to MPU private TIPB bridge
- ❏ DSP private TIPB router to DSP private TIPB bridge

Figure 2. Layer 4 Interface



2.1 TIPB Router Connections

Table 2. Private Peripherals

OMAP 5912 Peripheral	Interface	TIPB Router
DSP level 2 interrupt handler	Wrapper OCP	Private DSP
ULPD (clock and reset)	TIPB	Private MPU
OMAP5912 configuration	TIPB	Private MPU
Secure watchdog	Wrapper OCP	Private MPU
32-kHz watchdog	Wrapper OCP	Private MPU
MPU level 2 interrupt handler	Wrapper OCP	Private MPU
SHA_1 accelerator	Wrapper OCP	Private MPU
RNG random generator	Wrapper OCP	Private MPU
DES/3DES	Wrapper OCP	Private MPU
System DMA handler	Wrapper OCP	Private MPU

Table 3. MPU Peripherals Connected to MPU Shared TIPB Bridge

OMAP 5912 Peripheral	Interface	TIPB Router
HDQ/1-Wire	TIPB	Shared MPU
µWIRE	TIPB	Shared MPU
MMCSD/IO1	Wrapper OCP	Shared MPU
MPUIO	TIPB	Shared MPU
2 x LPG	TIPB	Shared MPU
SoSSI	Wrapper VIA	Shared MPU
RTC	TIPB	Shared MPU
Memory Stick	TIPB	Shared MPU
PWL	TIPB	Shared MPU
PWT	TIPB	Shared MPU
FAC	TIPB	Shared MPU
OS timer	TIPB	Shared MPU
USBOTG	Wrapper OCP	Shared MPU

Table 4. DSP Peripherals Connected to DSP Shared TIPB Bridge

OMAP 5912 Peripheral	Interface	TIPB Router
2 x MCSI	TIPB	Shared DSP
2 x McBSP	Wrapper OCP	Shared DSP

Table 5. DSP and MPU Shared Peripherals

OMAP 5912 Peripheral	Interface	TIPB Router	
		MPU	DSP
MMCSD/IO2	Static OCP	Shared	Shared
STI (reserved)	Dynamic OCP	Private	Shared
32-kHz synchronization timer	Dynamic OCP	Shared	Shared
4 x GPIO	Dynamic OCP	Shared	Shared
GPIO	Dynamic OCP	Shared	Shared
3 x UART	Static OCP	Shared	Shared
McBSP	Static OCP	Shared	Shared
I^2C	Static OCP	Shared	Shared
SPI	Static OCP	Shared	Shared
8 x GP timers	Static OCP	Shared	Shared
NAND flash control	Static OCP	Shared	Shared
Static switch configuration	TIPB	Shared	Shared

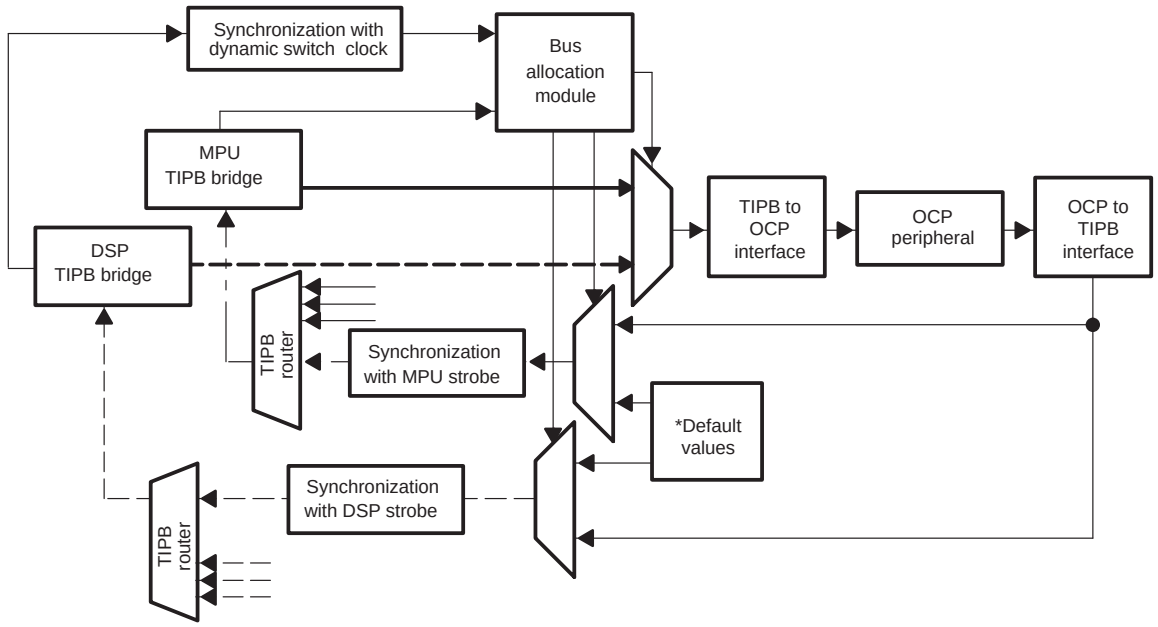
2.2 Protocols

The layer 4 interface is connected to several peripherals, which use different protocols:

- ❏ TI peripheral bus or TIPB bus
- ❏ Versatile interconnection architecture designated also as VIA bus or VIA protocol
- ❏ Open core protocol designated as OCP

2.3 Dynamic Switch

Figure 3. Dynamic Switch for OCP Peripheral



Note: Default value: To simplify the TIPB router implementation and reduce the toggling on buses, a default value is returned to the host, which does not access the peripheral.

2.4 Functional Description

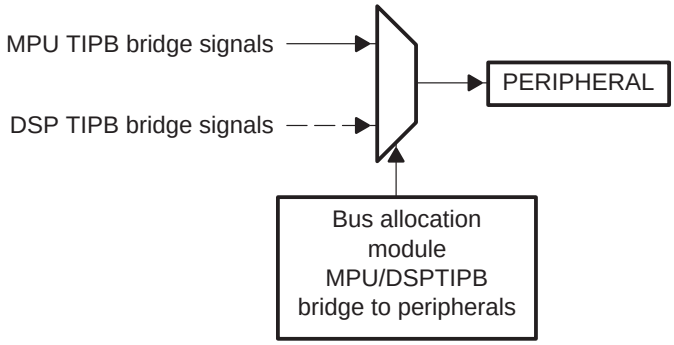
2.4.1 Bus Allocation

MPU/DSP TIPB Bridge to Peripherals

The peripheral TIPB buses are shared between the MPU TIPB bridge and the DSP TIPB bridge. Both bridges can access the same peripheral. Therefore, a bus allocation module is required to decide which one can access to the peripheral bus in case of conflict. This arbitration occurs after a synchronization of the DSP enable signal with the dynamic switch clock (ARMPER_CLK).

If no more than one of the two masters requests the bus, the bus is allocated to the requesting master. Otherwise, the MPU TIPB bridge takes control of the bus. DSP TIPB bridge accesses the peripheral after the data has been returned to the MPU TIPB bridge. During the transfer, the access size can be either 16-bit for the DSP or 8-, 16-, or 32-bit for the MPU.

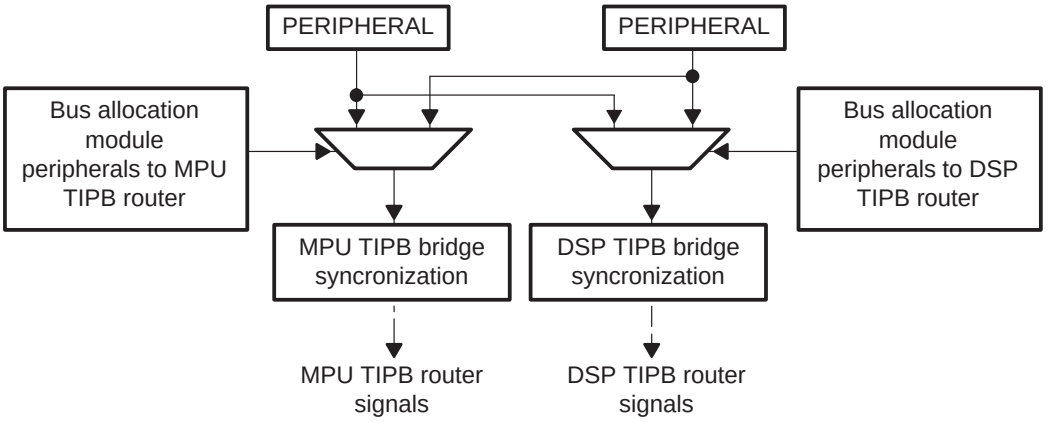
Figure 4. Transfer from Host to Peripheral



Peripherals to MPU/DSP TIPB Bridge

All peripherals can send data back on a dedicated bus (one bus per peripheral—8-, 16-, or 32-bit according to each peripheral requirement), either to the MPU or to the DSP.

Figure 5. Transfer from Peripheral to Host



Note: Default value: To simplify the TIPB router implementation and reduce the toggling on buses, a default value is returned to the host that does not access the peripheral.

2.4.2 16-bit Accesses on 32-Bit Atomic Registers

There is no protection inside the dynamic switch to ensure the atomic reading or writing of one 32-bit register with two 16-bit accesses. The nonoverlapping requests from the two hosts must be controlled in the software.

2.4.3 Conflicting Transaction

When one host requires an access to the peripheral, an enable host signal is set (synchronously with the host strobe). With the DSP TIPB bridge strobe, a DSP TIPB bridge enable is generated, and with the MPU TIPB bridge strobe an MPU TIPB bridge enable is generated. DSP TIPB bridge enable and MPU TIPB bridge enable are then resynchronized with the dynamic switch clock to perform arbitration. This arbitration takes place during one dynamic switch clock cycle. In case of conflict, the MPU TIPB bridge has the priority and the DSP TIPB bridge waits until the ready has been sent to the MPU TIPB bridge. Then the DSP TIPB bridge performs its access.

2.4.4 Abort Transaction

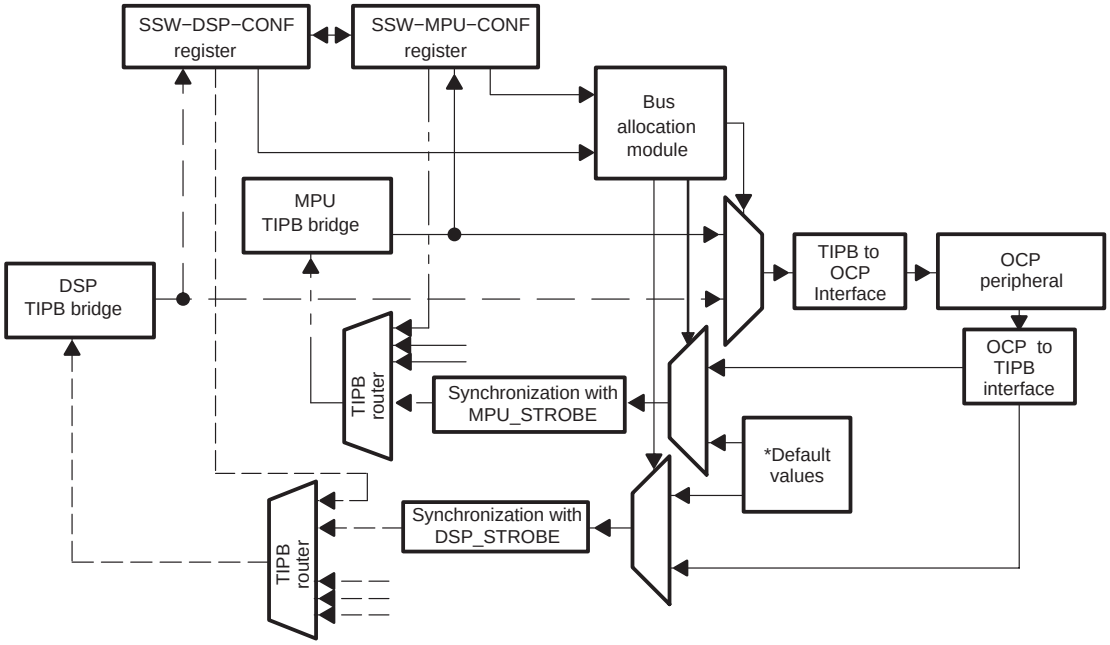
After a predetermined number of strobe cycles (DSP_nStrobe or MPU_nStrobe), if a peripheral does not activate its ready signal, the TIPB bridge generates an abort. The dynamic switch uses this abort to kill the current access. The maximum number of strobe cycles is programmable within the DSP TIPB bridge for DSP_nStrobe, and within the MPU TIPB bridge for MPU_nStrobe.

2.4.5 Reset Methodology

An MPU initiator reset is always used in the dynamic switch to reset an eventual MPU ongoing transaction. A DSP initiator reset is also used into the dynamic switch to reset an eventual DSP ongoing transaction.

2.5 Static Switch

Figure 6. Static Switch for OCP Peripheral



Note: Default value: To simplify the TIPB router implementation and reduce the toggling on buses, a default value is returned to the host which does not access the peripheral.

The static switch module allows the switching of a given peripheral between two TIPB host buses. This module enables static sharing between the DSP and the MPU processors. The switch between the two TIPB buses is implemented through a basic protocol. The software must control carefully the accessibility of each shared peripheral by the desired host. The programmer must ensure that the ongoing DMA transfers from/to the peripheral are completed before updating the peripheral ownership. There is no protection from the static switch hardware.

If the static switch is not well programmed before an access to a shared peripheral register, the corresponding ready signal is not sent back to the host. The result is a time-out error generated by the host TIPB bridge, depending on the TIPB bridge programming.

Two registers control the accessibility from the host to each statically sharedperipheral:

- ❏ SSW_MPU_CONF in the MPU peripheral address space
- ❏ SSW_DSP_CONF in the DSP peripheral address space

To perform an MPU access, the MPU software must first set (write 1) the MPU_SWITCH bit in the SSW_MPU_CONF register. Then all the MPU TIPB bus signals are connected either to the TIPB bus peripheral signals (for a TIPB peripheral), or to the dedicated wrapper TIPB bus interface (for OCP peripheral).

To perform a DSP access, the MPU software must first authorize this access by resetting the MPU_SWITCH bit in the SSW_MPU_CONF register (write 0); then the DSP software must set the DSP_SWITCH in the SSW_DSP_CONF register.

If the MPU wants to access the SSW_MPU_CONF register when the DSP_SWITCH bit is set, the access is completed but a write access has no effect on the register.

Symmetrically, if the DSP wants to access the SSW_DSP_CONF register when the MPU_SWITCH bit is set, the access is completed but a write access has no effect on the register.

The following programming sequence enables DSP access to the peripheral:

1) Check that the MPU_SWITCH bit is reset (no MPU access is ongoing on the same peripheral).
2) Set the DSP_SWITCH bit in the SSW_DSP_CONF register (DSP software: write 1)
3) Check that the DSP_SWITCH is set (in case of priority conflict with the MPU software).
4) Perform the shared peripheral access through the DSP TIPB bus.
5) After the completion of this access, reset the DSP_SWITCH bit in the SSW_DSP_CONF register (write 0).

The following programming sequence enables MPU access to the peripheral:

6) Check that DSP_SWITCH bit is reset (no DSP access is on going on the same peripheral).
7) Set the MPU_SWITCH bit in the SSW_MPU_CONF register (MPU software: write 1).
8) Check that the MPU_SWITCH is set (in case of priority conflict with the DSP software).
9) Perform the shared peripheral access through the MPU TIPB bus.
10) After the completion of this access, reset the MPU_SWITCH bit in the SSW_MPU_CONF register (write 0)

Depending on the host interface protocol for the peripheral, a dedicated wrapper is connected between the host and the peripheral bus. Two different peripheral protocols are supported:

- TIPB
- OCP

2.5.1 Simultaneous Access

Setting both the DSP_SWITCH bit and MPU_SWITCH bits is impossible, by design. An access control and arbitration phase in the ARMPER_CLK clock domain ensures that this programming conflict never occurs.

2.5.2 TIPB to OCP Interface

A full resynchronization process between the MPU TIPB interface, the DSP TIPB interface, and the peripheral interface OCP clock is implemented. The TIPB bus is stalled until the decoding of the peripheral acknowledge by the wrapper. The decoded acknowledge is then synchronized with the TIPB strobe to generate the ready (end of TIPB transaction).

2.6 Abort Transaction

After a predetermined number of strobe cycles (DSP_nStrobe or MPU_nStrobe), if a peripheral did not complete the transaction, the TIPB bridge generates an abort. The static switch uses this abort to kill the current access. The maximum number of strobe cycles is programmable, within the DSP TIPB bridge for DSP_nStrobe, and within the MPU TIPB bridge for MPU_nStrobe configuration registers.

2.7 Peripheral Static Switch Configuration Registers

The registers listed in Table 6 are accessible through the DSP shared TIPB bridge (16-bit access) or through the MPU shared TIPB bridge (16- or 32-bit accesses).

Table 6. Static Switch Configuration Registers Offset Addresses

Register Name	# Bits	Offset	Bus
UART1_SSW_MPU_CONF	16/32	0x000	MPU
UART1_SSW_DSP_CONF	16		DSP
UART2_SSW_MPU_CONF	16/32	0x020	MPU

Table 6. Static Switch Configuration Registers Offset Addresses (Continued)

Register Name	# Bits	Offset	Bus
UART2_SSW_DSP_CONF	16		DSP
UART3_SSW_MPU_CONF	16/32	0x040	MPU
UART3_SSW_DSP_CONF	16		DSP
McBSP_SSW_MPU_CONF	16/32	0x090	MPU
McBSP_SSW_DSP_CONF	16		DSP
I2C_SSW_MPU_CONF	16/32	0x0A0	MPU
I2C_SSW_DSP_CONF	16		DSP
SPI_SSW_MPU_CONF	16/32	0x0B0	MPU
SPI_SSW_DSP_CONF	16		DSP
TIMER1_SSW_MPU_CONF	16/32	0x0C0	MPU
TIMER1_SSW_DSP_CONF	16		DSP
TIMER2_SSW_MPU_CONF	16/32	0x0D0	MPU
TIMER2_SSW_DSP_CONF	16		DSP
TIMER3_SSW_MPU_CONF	16/32	0x0E0	MPU
TIMER3_SSW_DSP_CONF	16		DSP
TIMER4_SSW_MPU_CONF	16/32	0x0F0	MPU
TIMER4_SSW_DSP_CONF	16		DSP
TIMER5_SSW_MPU_CONF	16/32	0x100	MPU
TIMER5_SSW_DSP_CONF	16		DSP
TIMER6_SSW_MPU_CONF	16/32	0x110	MPU
TIMER6_SSW_DSP_CONF	16		DSP
TIMER7_SSW_MPU_CONF	16/32	0x130	MPU
TIMER7_SSW_DSP_CONF	16		DSP
TIMER8_SSW_MPU_CONF	16/32	0x140	MPU
TIMER8_SSW_DSP_CONF	16		DSP
NFCtrl_SSW_MPU_CONF	16/32	0x150	MPU

Table 6. Static Switch Configuration Registers Offset Addresses (Continued)

Register Name	# Bits	Offset	Bus
NFCtrl_SSW_DSP_CONF	16		DSP
MMCSD2_SSW_MPU_CONF	16/32	0x160	MPU
MMCSD2_SSW_DSP_CONF	16		DSP

For each static-shared peripheral, two registers are defined (Table 7 and Table 8): one is dedicated to MPU accesses, and the other to DSP accesses.

Table 7. Peripheral MPU Static Switch Configuration Register (xxx_SSW_MPU_CONF)

Base Address = FFFB C800, Offset Address = see Table 6				
Bit	**Name**	**Description**	**Access**	**Reset**
31:2	–	Reserved	R	0x00000000
1	DSP_SWITCH	0: No DSP TIPB bridge access required 1: DSP TIPB bridge access required	R	0
0	MPU_SWITCH	0: No MPU TIPB bridge access required 1: MPU TIPB bridge access required	RW	1

This register is only accessible through the MPU shared TIPB bridge.

It is used to perform MPU access on the peripheral.

Table 8. Peripheral DSP Static Switch Configuration Register (xxx_SSW_DSP_CONF)

Base Address = FFFB C800, Offset Address = see Table 6				
Bit	**Name**	**Description**	**Access**	**Reset**
15:2	–	Reserved	R	0x00000000
1	DSP_SWITCH	0: No DSP TIPB bridge access required 1: DSP TIPB bridge access required	RW	0
0	MPU_SWITCH	0: No MPU TIPB bridge access required 1: MPU TIPB bridge access required	R	1

This register is accessible only through the DSP shared TIPB bridge.

2.7.1 Reset Methodology

An MPU or a DSP initiator reset is always used internally by the static switch.

A reset on the static switch configuration register resets the DSP_SWITCH bit and sets the MPU_SWITCH bit, which turns the static switch in the MPU position.

2.8 TIPB1-VIA/VIA-TIPB Wrapper

The synchronous VIA protocol requires a free-running clock and synchronous signals relative to this clock.

In addition to the TIPB bus signals, DMA requests need to be adapted.

Two successive DMA requests must be separated by at least two system DMA clock cycles.

2.8.1 Reset Methodology

The internal state machine of the wrapper is reset by the MPU peripheral reset.

For the SoSSI peripheral, two resets must be provided to the peripheral:

- L4_VIA_Reset: synchronous with the CK_DPLL1OUT
- L4_DMA_Reset: synchronous with the DMA_LCDFREE_CLK

In both cases, the reset source is the MPU peripheral reset.

For the CCP peripheral, one reset is generated to the peripheral; L4_VIA_RESET is synchronous with the ARMPER_CLK.

2.9 TIPB-OCP/OCP-TIPB Wrapper

The synchronous OCP protocol requires a free-running clock and synchronous signals realtive to this clock.

The free-running clock is ARMPER_CLK.

2.9.1 Reset Methodology

The NO_RESET signal comes from the wrapper to the connected peripheral.

2.9.2 DSP TIPB to OCP

This wrapper is sensitive to either DSP watchdog reset or DSP peripheral reset in order to clear the interface state machine.

2.9.3 MPU TIPB to OCP

This wrapper is sensitive to MPU watchdog reset or MPU peripheral reset in order to clear the internal state machine.

2.10 TIPB-OCP/OCP-TIPB Wrapper for USBOTG

The synchronous OCP protocol requires a free-running clock and synchronous signals relative to this clock.

The free-running clock is ARMPER_CLK.

This wrapper is used for transmissions with USB client, USB host, and OTG.

Depending on the access, an increment is applied to the address:

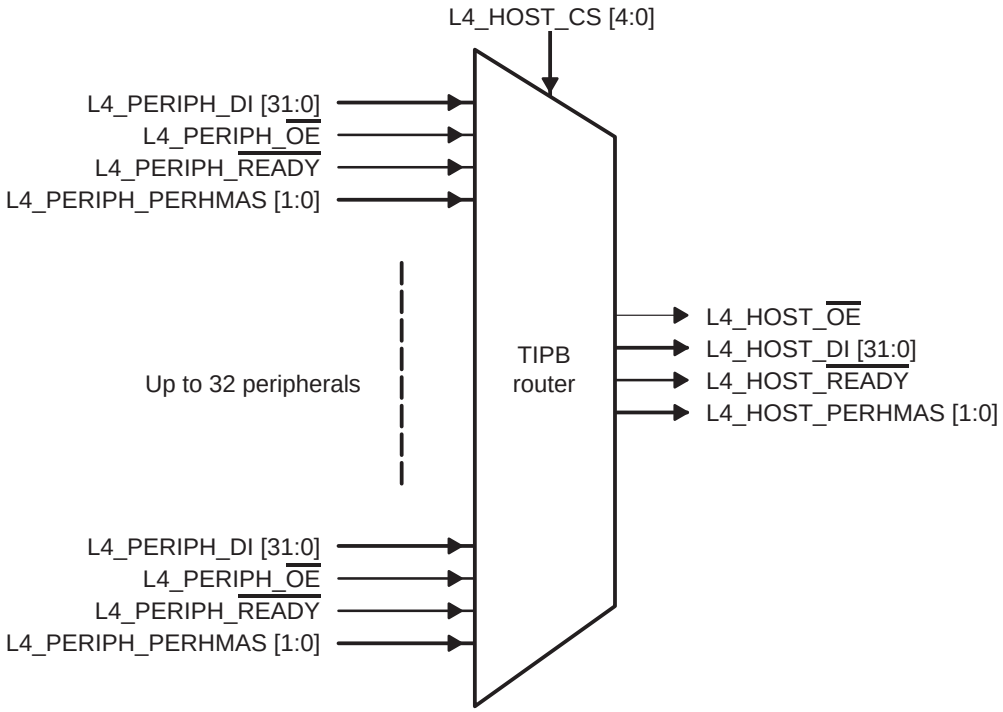
- If USB host is accessed, the address remains unchanged.
- If USB client is accessed, the address is incremented by 0x200h.
- If OTG is accessed, the address is incremented by 0x300h.

2.10.1 Reset Methodology

This wrapper is sensitive to MPU peripheral reset in order to reset the internal state machine.

2.11 TIPB Router

Figure 7. TIPB Router Block Diagram



The TIPB router is a simple multiplexer that returns information coming from peripherals to the host (DSP or MPU). The information concerns a current READ or WRITE access on a peripheral. This TIPB router supports up to 32 peripheral connections.

2.12 Peripheral Instantiation

Table 9. Private Peripherals Instantiation

OMAP 5912 Peripheral	Interface	Instantiation					TIPB Router
				Access Size			
		Address Bus Alignment†	Data	8b	16b	32b	
3 DSP level 2 interrupt handler	TIPB	Byte	16	–	R	–	Private DSP
ULPD (clock and reset)	TIPB	Byte	16	–	R	–	Private MPU
OMAP5912 configuration	TIPB	Byte	32	–	–	R	Private MPU
Secure watchdog	Wrapper OCP	Byte	32	–	R	R	Private MPU
32-kHz watchdog	Wrapper OCP	Byte	32	–	R	R	Private MPU
2 MPU level 2 interrupt handlers	TIPB	Byte	32	–	–	R	Private MPU
SHA_1 accelerator	Wrapper OCP	Byte	32	–	–	R	Private MPU
RNG random generator	Wrapper OCP	Byte	32	–	–	R	Private MPU
DES/3DES	Wrapper OCP	Byte	32	–	–	R	Private MPU
System DMA handler	Wrapper OCP	Byte	32	–	–	R	Private MPU

† Address bus alignment:
Byte: The least significant bit of the peripheral address bus corresponds to a byte address in the peripheral.
16-bit: The least significant bit of the peripheral address bus corresponds to a 16-bit address in the peripheral.
32-bit: The least significant bit of the peripheral address bus corresponds to a 32-bit address in the peripheral.

Table 10. MPU Peripherals Connected to the MPU Shared TIPB Bridge Instantiation

OMAP 5912 Peripheral	Interface	Instantiation					TIPB Router
		Address Bus Alignment†	Data	Access Size 8b	16b	32b	
HDQ/1-Wire	TIPB	Byte	8	R	–	–	Shared MPU
µWIRE	TIPB	Byte	16	–	R	–	Shared MPU
MMCSD/IO1	Wrapper OCP	16b	16	–	R	–	Shared MPU
MPUIO	TIPB	16b	16	–	R	–	Shared MPU
2 x LPG	TIPB	Byte	8	R	–	–	Shared MPU
SoSSI	Wrapper VIA	32b	32	–	–	R	Shared MPU
RTC	TIPB	Byte	8	R	–	–	Shared MPU
Memory Stick	TIPB	32b	32	–	–	R	Shared MPU
PWL	TIPB	Byte	8	R	–	–	Shared MPU
PWT	TIPB	Byte	8	R	–	–	Shared MPU
FAC	TIPB	Byte	16	–	R	–	Shared MPU
OS timer	TIPB	Byte	32	–	R	R	Shared MPU
USBOTG	Wrapper OCP	Byte	16	–	R	–	Shared MPU

† Address bus alignment:
Byte: The least significant bit of the peripheral address bus corresponds to a byte address in the peripheral.
16-bit: The least significant bit of the peripheral address bus corresponds to a 16-bit address in the peripheral.
32-bit: The least significant bit of the peripheral address bus corresponds to a 32-bit address in the peripheral.

Table 11. DSP Peripherals Connected to the DSP Shared TIPB Bridge Instantiation

OMAP 5912 Peripheral	Interface	Instantiation					TIPB Router
		Address Bus alignment	Data	Access Size 8b	16b	32b	
2 x MCSI	TIPB	Byte	16	–	R	–	Shared DSP
2 x McBSP	Wrapper OCP	Byte	16	–	R	–	Shared DSP

Table 12. DSP and MPU Shared Peripherals Instantiation

OMAP 5912 Peripheral	Interface	Instantiation					TIPB Router	
		Address Bus Alignment	Data	Access Size 8b	16b	32b	MPU	DSP
MMCSD/IO2	Static OCP	Byte	16	–	R	–	Shared	Shared
STI (reserved)	Dynamic OCP	Byte	32	–	R	R	Private	Shared
32-kHz synchronization timer	Dynamic OCP	Byte	32	–	R	R	Shared	Shared
4 x GPIO	Dynamic OCP	Byte	16	–	R	R	Shared	Shared
GPIO pin control	Dynamic OCP	Byte	16	–	R	R	Shared	Shared
3 x UART	Static TIPB	Byte	8	R	–	–	Shared	Shared
McBSP	Static OCP	Byte	16	–	R	–	Shared	Shared
I^2C	Static OCP	Byte	16	R	R	–	Shared	Shared
SPI	Static OCP	Byte	32	–	R	R	Shared	Shared
8 x GP timers	Static OCP	Byte	32	–	R	R	Shared	Shared
NAND flash control	Static OCP	Byte	32	R	R	R	Shared	Shared
Static switch configuration	TIPB	Byte	32	R	R	R	Shared	Shared

3 OCP Interconnect

The OCP, SSI interconnects, and static switches manage data transfers between OMAP3.2 and the SSI or VLYNQ and USB peripherals.

The SSI is a full-duplex interface, using a synchronous serial interconnect protocol (SSI). This protocol consists of a transmitter called (SST) in SSI, which is in charge of transmitting information, and a receiver called (SSR), which is in charge of receiving information

SSI DMA transfers are handled by the generic distributed DMA (GDD).

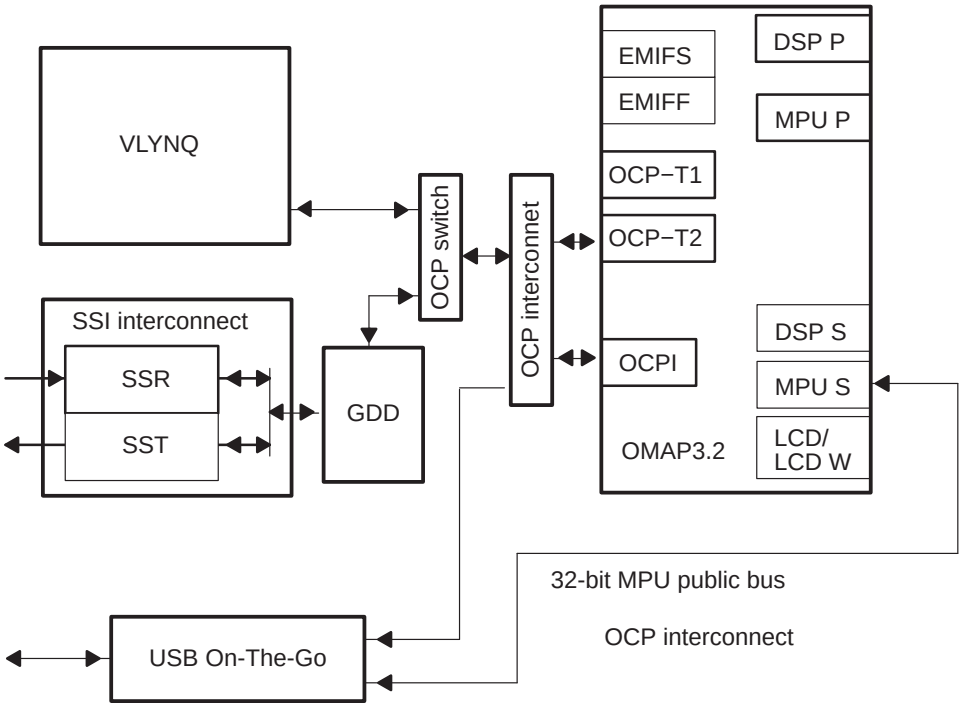
Because the SSI uses a proprietary bus interface (versatile interconnection architecture (VIA)), bridges are implemented in the SSI interconnect to perform data transfers between the SSI VIA bus and the OMAP3.2 OCP bus.

VLYNQ is also a full duplex interface, using a VBUS 1.0 protocol. A wrapper around the VLYNQ core performs the protocol conversion VBUS to OCP and OCP to VBUS.

In addition, data transfer between the USB host and OMAP3.2 are done through the OCP interconnect and the OCPI port.

The GDD has priority over the USB.

Figure 8. OCP and SSI Interconnects



3.1 OCP Introduction

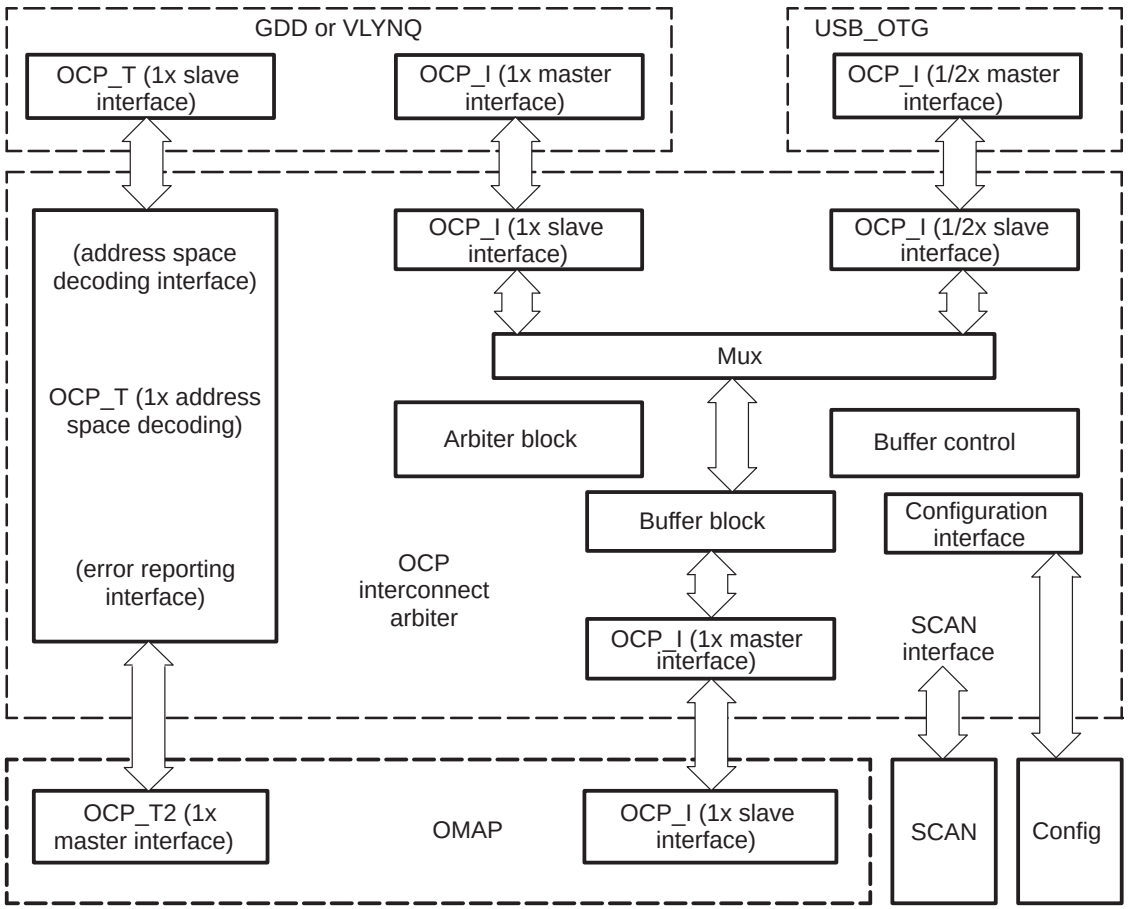
The OCP arbiter must interconnect two OCP master devices (peripherals), such as GDD or VLYNQ and USB_OTG peripheral, to a single OMAP OCP_I slave port. GDD or VLYNQ OCP_I port has priority over USB_OTG OCP_I port.

All interfaces are 32-bit data width and support burst and nonburst operations.

Because the OMAP OCP_I port does not support split bursts, the buffer block translates split burst (master can insert wait states between valid commands during burst) into a simple burst action (the master cannot insert wait states between valid commands during burst).

The OCP arbiter interconnect also provides address space decoding and standard OCP error signaling if the address is out of GDD/SSI or VLYNQ address space.

Figure 9. OCP Interconnect



3.2 Module Features

The OCP interconnect arbiter provides the following features:

- One OCP master port 32-bit width, 1x balanced clock, 4x32 (FOUR-TWO-TWO-LAST incremented burst only) nonsplit burst and nonburst accesses (OMAP OCP_I interconnect)
- One OCP slave port 32-bit width, 1x balanced clock, 4x32 split burst and nonburst accesses (GDD OCP_I interconnect)
- One OCP slave port 32-bit width, 1/2x balanced clock, 4x32 split burst and nonburst accesses (USB_OTG OCP_I interconnect)
- Translate split burst to nonsplit burst accesses for OMAP OCP_I slave interface (buffer for burst transaction)
- Offer high priority to OCP_I 1x master (GDD OCP_I has the priority)
- Address space decoding for GDD/SSI and standard error report to the OCP_T2 OMAP port when address is out of GDD/SSI address space.

3.3 1x OCP Slave Port

The OCP interconnect arbiter 1x OCP slave port has the following features:

- OCP interface (slave)
- Synchronous
- Balanced 100-MHz clock
- 32-bit data width only
- Support 4x32 burst and nonburst accesses

This port is directly connected to the GDD memory port.

3.4 1/2x OCP Slave Port

The OCP interconnect arbiter 1/2x OCP slave port has the following features:

- OCP interface (slave)
- Synchronous
- Balanced 100-MHz clock using a 50-MHz phase indication
- 32-bit data width only
- Support 4x32 burst and nonburst accesses

This port is directly connected to the USB_OTG OCP master interface.

3.5 1x OCP Master Port

The OCP interconnect arbiter 1x OCP master port has the following features:

- OCP interface (master)
- Synchronous
- Balanced 100-MHz clock
- 32-bit data width only
- Support 4x32 burst (4−2−2−LAST only) and non-burst accesses
- Non-split burst only

This port is directly connected to the OMAP OCP_I slave interface.

3.6 Address Space

The address space for the GDD/SSI and VLYNQ is shown in Table 13 (and is fixed; no configuration registers are needed).

Table 13. GDD/SSI Address Space

Block Name	Start Address	End Address	Size
SSI mapping	0x3000 0000	0x3000 0FFF	4K bytes
GDD mapping	0x3000 1000	0x3000 1FFF	4K bytes
VLYNQ conf mapping	0x3000 2000	0x3000 21FF	512 bytes
VLYNQ window mapping	0x3100 0000	0x34FF FFFF	64 Mbytes

3.7 Arbiter Block

The arbiter block controls the priority between GDD or VLYNQ and USB conflicting requests to the OMAP OCP-I port. The arbitration scheme is simple. If both 1x OCP and 1/2x OCP ports request at the same time, priority is given to the 1x port. In other words, the GDD or VLYNQ always has the highest priority.

3.8 Buffer Block

The buffer block acts as a buffer between the input and output registers. This block provides the buffering required to have all outputs come from the register (no combinatorial output). Also, because the OMAP OCP_I interface does not have the capability to handle split bursts, this buffer is used to provide burst transaction without wait states (nonsplit burst) when bursts are initiated by the USB OCP_I interface. The buffer block contains four registers to handle the 4 x 32 burst.

The following configurations are possible for the buffer block in case of a burst request transaction**:**

- GDD or VLYNQ side, send as soon as possible
- GDD or VLYNQ side, wait for buffer mode
- USB side, send as soon as possible mode
- USB side, wait for buffer mode

For all USB, GDD, and VLYNQ simple read or write transactions, the request is sent directly to the OMAP OCP_I interface.

4 SSI Interconnect

4.1 Introduction

The SSI interconnect manages accesses to OMAP5912 peripherals through GDD, acting as a bridge between external serial peripherals (using VIA buses) and the internal parallel bus (OCP). On one side it is connected to SSI, using two VIA buses, and on the other to GDD, using an OCP bus (see Figure 9). The SSI module is composed of two blocks, SSR and SST, each using a different VIA bus.

SSI interconnect is composed of five blocks:

- Decode

  The decode module dispatches incoming transactions either to OCP to the VIA asynchronous bridge or to the synchronous bridge of OCP to VIA.

- OCP to the VIA synchronous bridge

  OCP to the VIA synchronous bridge connects SST to GDD via the OCP bus. Because the SST module is synchronous with the OCP bus, resynchronization is not required and a faster synchronous bridge can be used.

- OCP to the VIA asynchronous bridge

  OCP to the VIA asynchronous bridge connects SSR to the OCP bus. Because the SSR module works on a clock that is asynchronous and faster than the OCP bus, resynchronization is required.

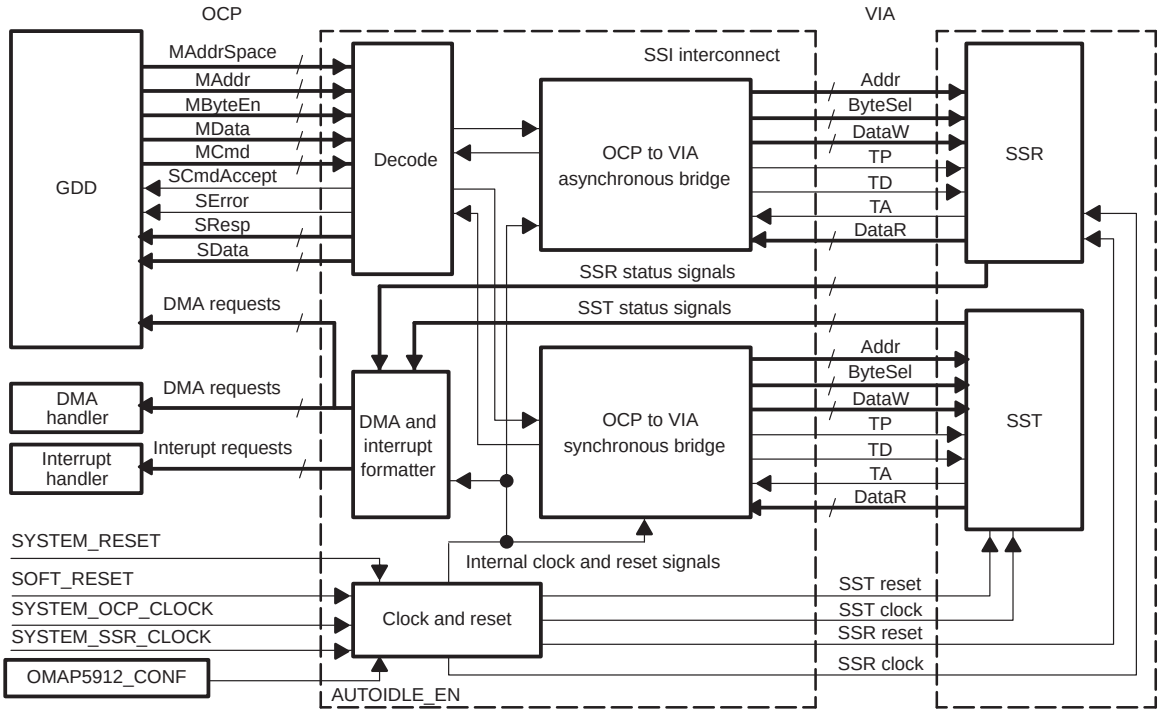
- DMA and interrupt formatter

  DMA and interrupt formatter is responsible for generating DMA and interrupt requests from the SSR and SST status signals.

- Clock and reset module

  The clock and reset module manages clock and reset signals for the SSI interconnect module and the SSI (SSR and SST) module. This module re-synchronizes reset signals and implements the AUTOIDLE mechanism for the SSI interconnect module.

Figure 10. SSI Interconnect



4.2 OCP to VIA Asynchronous Bridge

OCP to the VIA asynchronous bridge is a connection bridge between an OCP bus and a VIA bus. This submodule is a target on the OCP bus and a master on the VIA bus. On the OCP, this bridge provides only basic functionality signals plus SError (slave error). The slave error signal is needed because only basic read and basic write transactions are supported.

Burst transactions are treated as a sequence of basic read and write accesses and do not cause SError to be asserted. When unsupported commands occur, SError is asserted.

4.3 OCP to VIA Synchronous Bridge

OCP to the VIA synchronous bridge is a connection bridge between an OCP bus and a VIA bus. This submodule is a target on the OCP bus and a master on the VIA bus.

On the OCP part, this bridge provides only basic functionality signals plus slave error.

4.4 Decode

This submodule decodes the address received from the OCP initiator. If the current OCP command is for SSR or SST, the decode block forwards this command to that block while keeping the other one inactive.

For SSI, the allocated address space begins at 0x30000000 and ends at 0x30000FFF, in case the address is generated by GDD directly. When the address is provided by the GDD configuration port, the allocated address space begins at 0x000 and ends at 0xFFF. If the address does not belong to one of these two address spaces, SError is asserted. There are 4K bytes of memory allocated for SSI, divided in the following mode: 2K bytes for SST and 2K bytes for SSR. For basic compatibility with the previous version of the chip, the SST address space is assigned the lower address range, and the SSR address space uses the higher addresses.

Table 14. SSI Address Spaces

	Start Address	End Address	MAddrSpace	Address Width
SSI address space by GDD	0x3000_0000	0x3000_0FFF	0000	32 bits
SST address space	0x3000_0000	0x3000_07FF	0000	32 bits
SSR address space	0x3000_0800	0x3000_0FFF	0000	32 bits
SSI address space by OMAP	0x000	0xFFF	0001	12 bits
SST address space	0x000	0x7FF	0001	12 bits
SSR address space	0x800	0xFFF	0001	12 bits

4.5 DMA and Interrupt Formatter

The DMA and interrupt block formats status signals provided by SSR and SST and generates interrupts and DMA requests.

When programming SSR or SST in interrupt mode, take care to avoid reading the buffer before the interrupt signal is asserted.

4.6 Clock and Reset Module

The clock and reset module manages the clock and reset signals for the SSI interconnect module and the SSI (SSR and SST) module. This module resynchronizes reset signals and implements the AUTOIDLE mechanism for the SSI interconnect module.

4.7 Programming Model

4.7.1 OCPI REGISTERS (MPU Address: FFFE:C320)

The OCPI interface allocates memory spaces to the initiator in secure mode and logs unauthorized data transfers and aborts.

See the *OMAP3.2 Technical Reference Manual* (SWPU019E) for additional information.

4.7.2 OCPT REGISTERS (MPU Address: FFFE:CC00)

The OCP-T2 interface manages priority between concurrent data transfers and logs unauthorized data transfers and aborts.

See the *OMAP3.2 Technical Reference Manual* (SWPU019E) for additional information.

4.7.3 Power Saving Modes

To avoid free-running clocks in the OCP and the SSI interconnects, it is possible to shut off the clocks when no data transfer is in progress. This feature is called autoidle.

Bit [25] of MOD_CONF_CTRL_1 in OMAP5912 configuration enables the OCP interconnect to use its autoidle features.

Reset value is 0x1.

Bit [26] of MOD_CONF_CTRL_1 in OMAP5912 configuration enables the SSI interconnect to use its autoidle features.

Reset value is 0x1.

5 On-Chip/Off-Chip Memory and Peripheral Access Latencies

Table 15 and Table 16 list latencies for accesses either to off-chip memories or on-chip memories and peripherals.

All accesses are calculated assuming a 200-MHz MPU and DSP clocks.

Table 15 lists the OMAP3.2 timing parameters.

Table 15. OMAP3.2 Timing Parameters

OMAP3.2			
MPU clock cycle (ns)	5		
DSP clock cycle (ns)	5		
TC clock cycle	10		
MCU 50-MHz TIPB internal peripheral bus cycle (ns)	20		
100-MHz dynamic and static switch cycle (ns)	10		
DSP 33-MHz TIPB internal peripheral bus (ns)	30		
Memories			
Flash initial latency (TC cycle)	10	FLASH_INIT (ns)	100
Flash access clock cycle (TC cycle) @50-MHz device	2	FLASH_BURST (ns)	20
SDRAM RAS latency (TC cycle)	3	RAS_T (ns()	30
SDRAM CAS latency (TC cycle) func(memory type)	3	CAS_T (ns)	30†/20‡
SDRAM access clock cycle (TC cycle) @100-MHz device	1	BURSTCYCLE_T (ns)	5†/10‡
Secure RAM write latency (TC cycle) @ 100 MHz	3		
Secure ROM and RAM initial read latency (TC cycles) @100 MHz	3		
Secure ROM and RAM access clock cycles in (TC cycles) @ 100 MHz	1		
Datum per cycle func (memory type)	2†/1‡		
OMAP5912 Peripherals			
GDD functional clock (ns)	10		
SSR functional/interface clock (ns)	5		
SST functional/interface clock (ns)	10		
VLYNQ clock (ns)	10		
Computing			
Memory type = MDDR/SDR	Mddr†/Sdr‡		
Assumed address offset for I fetch miss (1−4)	1		
Flash width	16		
SDRAM width	16		

† Timing parameters are calculated for a mobile DDR.
‡ Timing parameters are calculated for a standard SDRAM.

Table 15. OMAP3.2 Timing Parameters (Continued)

DMA		
System DMA clock (ns)	10	
DSP DMA clock (ns)	5	
CCP		
CCP clock (ns)	20	CCP cycle

† Timing parameters are calculated for a mobile DDR.
‡ Timing parameters are calculated for a standard SDRAM.

Table 16 lists access time calculations for different class of peripherals.

Table 16. Peripheral Access Time Calculations

		Burst Read				**Single Read**		**Single Write**	
		First Data		**Line Fill**					
		ns	Initiator cycles	ns	Initiator cycles	ns	Initiator cycles	ns	Initiator cycles
MPU to boot ROM*	Burst read/write, 32b read, 16/8b read	70	14	185	37	70	14		
MPU to secure ROM*	Burst read/write, 32b read, 16/8b read	70	14	185	37	70	14		
MPU to secure RAM*	Burst read/write, 32b read/write, 16/8b read/write	70	14	185	37	70	14	70	14

† This latency value includes the following:
− Initial latency: from enable to 1st transaction
− Pipeline latency: from source to destination
− Close latency: from last acknowledge to release of the channel
− For the normal and dedicated P-channel case
‡ Page open: external SDRAM is a mobile DDR
§ 51 equivalent TC cycles = 19 TC cycles plus 16 CCP interface cycles

Table 16. Peripheral Access Time Calculations (Continued)

		Burst Read				Single Read		Single Write	
		First Data		Line Fill					
		ns	initiator cycles	ns	initiator cycles	ns	initiator cycles	ns	initiator cycles
OCP-T2 Peripherals									
MPU to SSR	Read					155	31		
MPU to SST	Write							110	22
OCP-I Interconnect									
MPU to GDD	Read/write					70	14	70	14
MPU Public Peripheral									
MPU to UART	Read/write					135	27	155	31
GDD Accesses									
GDD access to SSR	DMA request to read/write					125	12.5		
GDD access to SST	DMA request to read/write							50	5
GDD access to SDRAM	Request to acknowledge read/write	175†/200‡	17.5†/ 20‡	145†/ 170‡	14.5†/ 17‡	140†/ 130‡	14†/ 13‡	110†/ 100‡	11†/10‡
MPU Private Peripherals									
MPU access to level1 int handler register	Read/write					80	16	100	20

† This latency value includes the following:
− Initial latency: from enable to 1st transaction
− Pipeline latency: from source to destination
− Close latency: from last acknowledge to release of the channel
− For the normal and dedicated P-channel case
‡ Page open: external SDRAM is a mobile DDR
§ 51 equivalent TC cycles = 19 TC cycles plus 16 CCP interface cycles

Table 16. Peripheral Access Time Calculations (Continued)

		Burst Read				Single Read		Single Write	
		First Data		Line Fill					
GDMA Accesses									
GDMA pipeline latency (1)	GDMA pipeline latency(initial, intermediate, close)					230	23	200	20
GDMA peripheral latency	From request to 4*32-bit read from CCP FIFO§	510	51						
GDMA EMIFF latency	DMA read/write from/to EMIFF	145†/170‡	14.5†/17‡	135†/160‡	13.5†/16‡	110†/100‡	11†/10‡	100†/90‡	10†/9‡
DSP Public Peripherals									
DSP to McBSP access	16 bits read/write					400	80	400	80
DSP Private Peripherals									
DSP to level 2.1 interrupt handler access	16 bits read/write					75	15	90	18
DSP_DMA Accesses									
DSP_DMA to EMIFF access	16 bits read/write (2)					130†/120‡	26†/24‡	120†/110‡	24†/22‡

† This latency value includes the following:
− Initial latency: from enable to 1st transaction
− Pipeline latency: from source to destination
− Close latency: from last acknowledge to release of the channel
− For the normal and dedicated P-channel case
‡ Page open: external SDRAM is a mobile DDR
§ 51 equivalent TC cycles = 19 TC cycles plus 16 CCP interface cycles

Table 16. Peripheral Access Time Calculations (Continued)

		Burst Read		Single Read		Single Write	
		First Data	Line Fill				
Other							
DSP to GDD access	OCPT2 :16 bits read/write from/to GDD			80	16	80	16
DSP to SSI access	OCPT2: 16 bits read/write from/to SSI			90	18	90	18
GDMA peripheral latency, NAND	From request to 4*32-bit read from NAND flash controller	710	71				
MPU Public Peripheral (OCP)							
MPU to NAND controller	MPU 32-bit read from NAND flash controller	50	10				

† This latency value includes the following:
− Initial latency: from enable to 1st transaction
− Pipeline latency: from source to destination
− Close latency: from last acknowledge to release of the channel
− For the normal and dedicated P-channel case
‡ Page open: external SDRAM is a mobile DDR
§ 51 equivalent TC cycles = 19 TC cycles plus 16 CCP interface cycles

Index

P

S

T