OMAP5910 Dual-Core Processor Functional and Peripheral Overview

Reference Guide

Literature Number: SPRU602C
January 2003
Revised October 2003

OMAP™
TEXAS INSTRUMENTS TECHNOLOGY

TEXAS INSTRUMENTS

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 provides a functional and peripheral overview of the OMAP5910 device.

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.

OMAP5910 Dual-Core Processor MPU Subsystem Reference Guide (literature number SPRU671)

OMAP5910 Dual-Core Processor DSP Subsystem Reference Guide (literature number SPRU672)

OMAP5910 Dual-Core Processor Memory Interface Traffic Controller Reference Guide (literature number SPRU673)

OMAP5910 Dual-Core Processor System DMA Controller Reference Guide (literature number SPRU674)

OMAP5910 Dual-Core Processor LCD Controller Reference Guide (literature number SPRU675)

OMAP5910 Dual-Core Processor Universal Asynchronous Receiver/Transmitter (UART) Devices Reference Guide (literature number SPRU676)

OMAP5910 Dual-Core Processor Universal Serial Bus (USB) and Frame Adjustment Counter (FAC) Reference Guide (literature number SPRU677)

OMAP5910 Dual-Core Processor Clock Generation and System Reset Management Reference Guide (literature number SPRU678)

OMAP5910 Dual-Core Processor General-Purpose Input/Output (GPIO) Reference Guide (literature number SPRU679)

OMAP5910 Dual-Core Processor MMC/SD Reference Guide (literature number SPRU680)

OMAP5910 Dual-Core Processor Inter-Integrated Circuit (I2C) Controller Reference Guide (literature number SPRU681)

OMAP5910 Dual-Core Processor Timer Reference Guide (literature number SPRU682)

OMAP5910 Dual-Core Processor Inter-Processor Communication Reference Guide (literature number SPRU683)

OMAP5910 Dual-Core Processor Camera Interface Reference Guide (literature number SPRU684)

OMAP5905 Dual-Core Processor Multichannel Serial Interface (MCSI) Reference Guide (literature number SPRU685)

OMAP5910 Dual-Core Processor Micro-Wire Interface Reference Guide (literature number SPRU686)

OMAP5910 Dual-Core Processor Real-Time Clock (RTC) Reference Guide (literature number SPRU687)

OMAP5910 Dual-Core Processor HDQ/1-Wire Interface Reference Guide (literature number SPRU688)

OMAP5910 Dual-Core Processor PWL, PWT, and LED Peripheral Reference Guide (literature number SPRU689)

OMAP5910 Dual-Core Processor Multichannel Buffered Serial Port (McBSP) Reference Guide (literature number SPRU708)

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

OMAP5910 Dual-Core Processor Functional and Peripherals Overview

1 Functional Overview

The OMAP5910™ is a highly integrated hardware and software platform designed to meet the application processing needs of next-generation embedded devices.

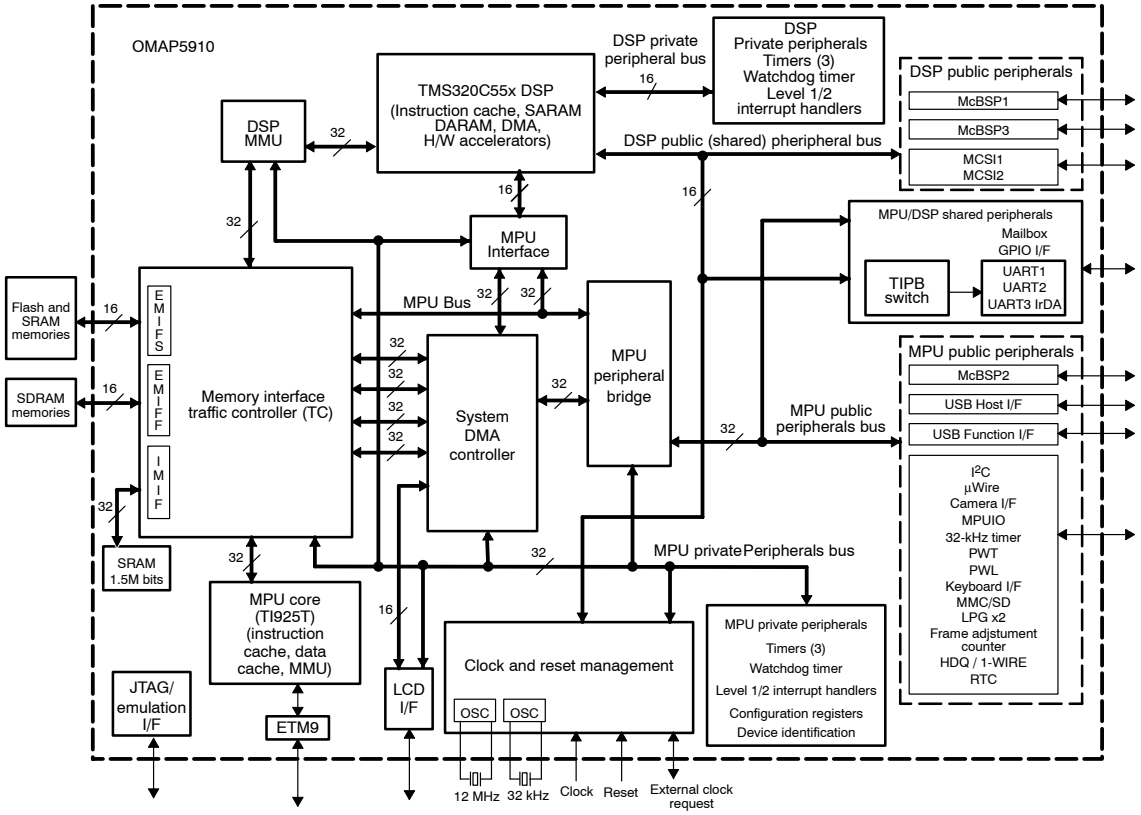
The OMAP5910 processor features a unique dual-core architecture that combines the command and control capabilities of the TI-enhanced ARM™ 925 processor (TI925T) with the high-performance and low-power capabilities of the TMS320C55x™ DSP core. These two key components of the OMAP5910 processor are:

- A TI reduced instruction set computer (RISC) microprocessor unit (MPU) subsystem. The MPU subsystem is based on the TI925T control processor, peripherals, and other components. The TI925T processor is based on the Advanced RISC Machines ARM9TDMI technology.
- A TI digital signal processor (DSP) subsystem. The DSP subsystem incorporates a TI TMS320C55x DSP, peripherals, and other components.

The OMAP5910 processor is available in the small 289-pin MicroStar™ BGA package (12x12 mm).

Figure 1 is a master block diagram of the 289-pin OMAP5910 processor. Figure 2 shows the OMAP5910 in more detail.

Figure 1. OMAP5910 Master Block Diagram



2 Description

The OMAP5910 processor features the first generation of the Texas Instruments Incorporated OMAP™ architecture.

The OMAP platform enables OEMs and ODMs to quickly bring to market devices featuring rich user interfaces, high processing performance, and long battery life through the maximum flexibility of a fully integrated mixed processor solution.

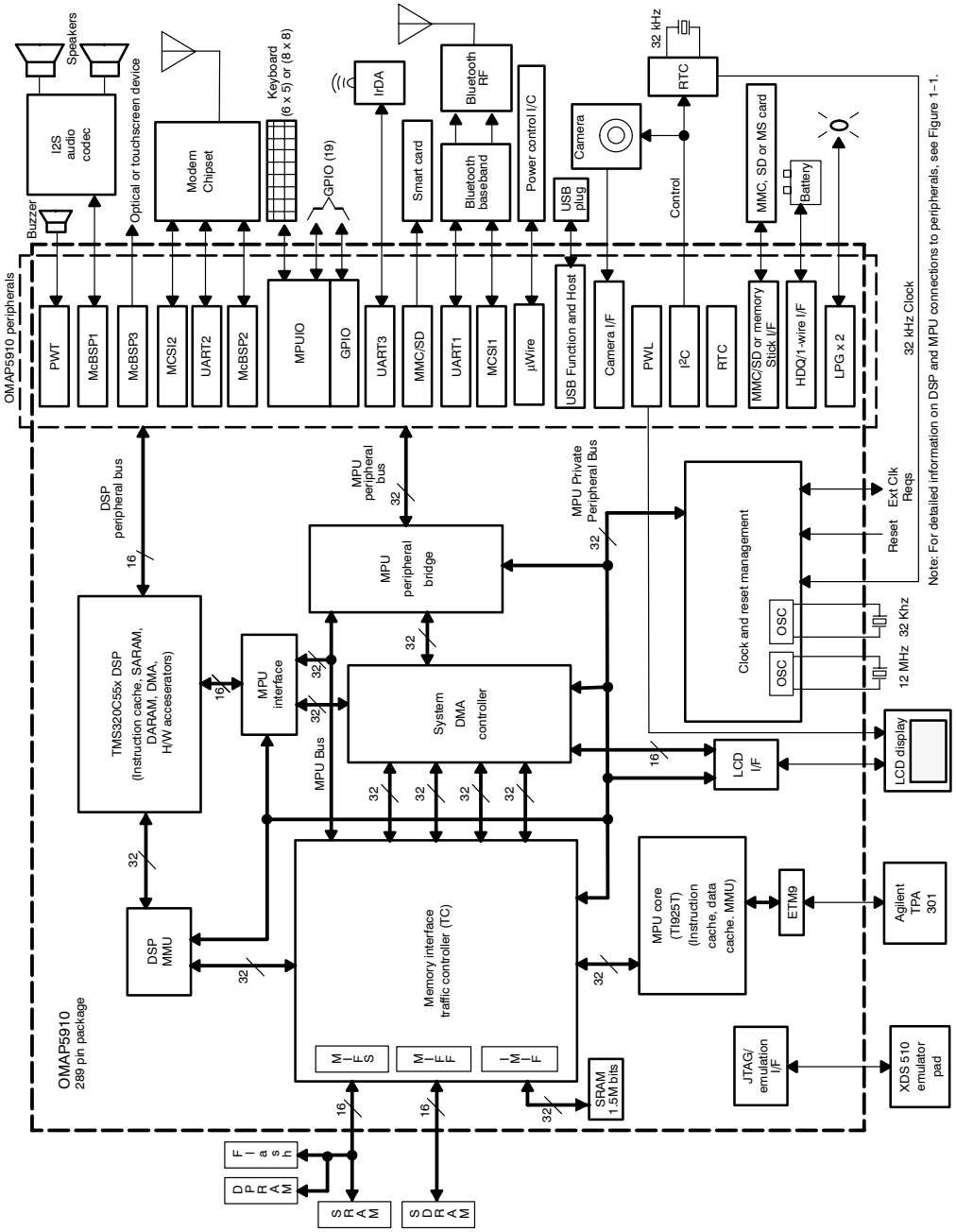
The dual-core architecture provides benefits of both DSP and RISC technologies, incorporating a TMS320C55x DSP core and a high-performance TI925T core.

The OMAP5910 device is designed to run leading open and embedded RISC-based operating systems, as well as the Texas Instruments (TI) DSP/BIOS™ software kernel foundation, and is available in a 289-ball MicroStar BGA package.

The OMAP5910 is targeted for the following applications:

- Applications processing devices
- Mobile communications
  - 802.11
  - Bluetooth™ wireless technology
  - Proprietary government and other
- Video and image processing (MPEG4, JPEG, Windows Media Video, etc.)
- Advanced speech applications (text-to-speech, speech recognition)
- Audio processing (MPEG-1 Audio Layer3 [MP3], AMR, WMA, AAC, and other GSM speech codecs)
- Graphics and video acceleration
- Generalized web access
- Data processing (fax, encryption/decryption, authentication, signature verification and watermarking)

Figure 2. OMAP5910 Diagram



Note: As a system real-time time clock, there are two possible solutions:

- The OMAP5910 internal solution (internal RTC and 32-kHz oscillator pads) is not a low-power solution, because the OMAP5910 RTC power supply is not separate from the OMAP5910 core power supply.
- The OMAP5910 external solution (external RTC and 32-kHz oscillators) is the recommended solution for low IDDQ.

3 Features

The OMAP5910 device has the following features:

- Ability to support reduced instruction set computer (RISC) and DSP operating systems
- TI925T MPU subsystem with:
  - Instruction cache (16K bytes) and data cache (8K bytes)
  - Memory management unit (MMU)
  - A 17-word write buffer (WB)
- DSP subsystem (C55x™ DSP core and subsystems) with:
  - Internal 32K-word dual-access RAM (DARAM), 48K-word single access RAM (SARAM), 16K-word ROM
  - Software-configurable instruction cache (12K words, 128-bit line size, 2-way set-associative + RAM set)
  - Hardware accelerators for video processing, pixel interpolation, and motion estimation
  - Six-channel DMA controller for high-speed data movement without DSP intervention
- DSP MMU for address translation and access permission checks
- System DMA controller with:
  - Six ports and nine independently programmable generic channels
  - An additional dedicated DMA channel tied to the liquid crystal display (LCD) controller
  - Ability to transfer 8-,16-, or 32-bit data between the external memory, the MPU, and peripherals with byte alignment and packing capability
  - Ability to perform simultaneous transfers (single or multiple burst) if no resources conflict
  - Low-power design (no clocking when idle)
- Two external memory interfaces that allow glueless hookup to:
  - A 16-bit bus interface to external memory interface slow (EMIFS), such as flash/SRAM/ROM/page-mode ROM/SB flash/DPRAM), with 128M bytes of memory space
  - A 16-bit bus interface to external memory interface fast (EMIFF), such as memory SDRAM, with 64M bytes of memory space

- 192K bytes of 32-bit-wide internal SRAM memory that allows local storage of operating system (OS) critical routines and that provides a direct path from the SRAM to the LCD controller
- An external memory traffic controller (TC) that allows asynchronous operation among the external memory interface, the MPU, and the DSP
- Mailboxes (two for MPU-to-DSP and two for DSP-to-MPU) for interprocessor communication
- Endianism conversion (default bypass, selectable, and configurable) between the DSP and the traffic controller and between the DSP and the MPU interface (MPUI) port boundaries
- Elastic buffering between the traffic controller and the MPU/DSP controllers to facilitate fully synchronous and synchronous scalable mode clock operations
- JTAG port for test, debug, and emulation
- Clock management:
  - One digital phase-locked loop (DPLL) and three clock management units for MPU, DSP, and traffic controller clock generation and management
  - System power management for idle mode and power-down functions
- Peripherals available for the OS, general-purpose housekeeping, and application-specific functions:
  - For the MPU:
    - Three 32-bit timers
    - A 16-bit watchdog timer
    - An interrupt handler
    - An LCD controller
    - Configuration registers
    - McBSP2 (multichannel buffered serial port)
    - Inter-integrated circuit (I^2C) interface
    - MicroWire interface
    - Keyboard interface
    - Universal serial bus (USB) function and host interface

- Camera interface
- Five MPUIO general-purpose input/output signals in default multiplexing mode; five more available through alternative pin multiplexing modes
- 32-kHz timer
- Pulse-width tone (PWT) module
- Pulse-width light (PWL) module
- Real-time clock (RTC) module
- Multimedia card (MMC) or serial data (SD) card interface
- HDQ and 1-Wire serial interface
- Two light emitting diode (LED) pulse generator modules
- Frame adjustment counter

- For the DSP:
  - Three 32-bit timers
  - A 16-bit watchdog timer
  - An interrupt handler
  - McBSP1: Multichannel buffered serial port
  - McBSP3: Multichannel buffered serial port
  - MCSI1: Multichannel serial voice interface
  - MCSI2: Multichannel serial voice interface
- Shared peripherals:
  - UART1: UART modem with autobaud (16C750 compatible)
  - UART2: UART modem with autobaud (16C750 compatible)
  - UART3: UART modem with IrDA (16C750 compatible)
  - Fourteen general-purpose input/output (GPIO)
  - Mailbox

4 Architecture

The OMAP5910 device includes the MPU subsystem, the DSP subsystem, a memory interface traffic controller, general-purpose peripherals, dedicated multimedia application (MMA) peripherals, and multiple interfaces. The MPU is the master of the platform, and it has access to the entire 16M bytes of memory space and to the 128K bytes of I/O space of the DSP subsystem. Additionally, the MPU and DSP share access to the internal SRAM and external memory interfaces.

5 Memory Maps

Figure 3 shows the MPU memory map. Figure 4 shows the DSP memory map.

Figure 3. MPU Memory Map

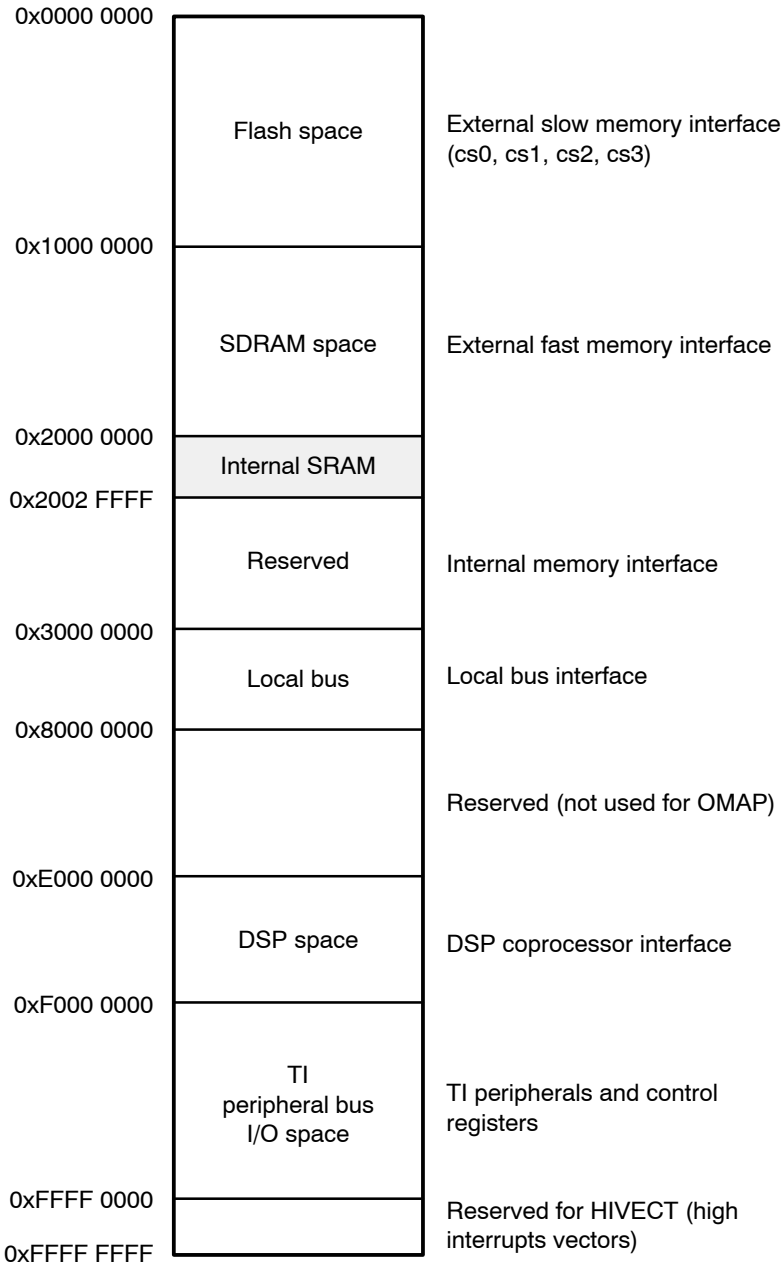
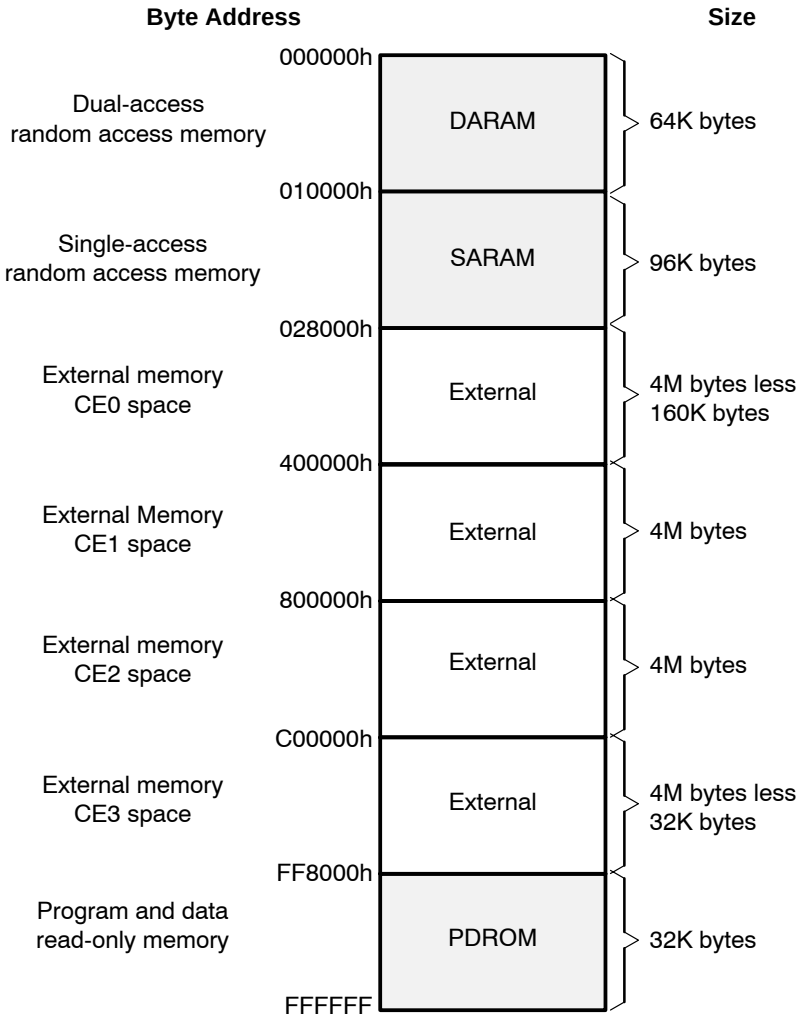


Figure 4. DSP Memory Map



The programmable DSP MMU configures how the DSP external address range is physically mapped to the MPU address range. For more information, see the *DSP Memory Management Unit* section.

6 Software Compatibility

Code compatibility with future OMAP59x devices is only possible if driver writers adhere to the conventions detailed in Section 6.1.

6.1 OMAP Driver Compatibility Conventions

All locations marked as reserved or unused in the documentation are written as zero, and, in general, values read from reserved locations are not used.

In practice, read-mask-update operations can be applied to registers that include reserved bits, provided the register is initialized by writing 0 to all reserved bits when the register is first used.

These conventions allow use of reserved bits to enable new features in future implementations. Initialization of the complete register, including reserved bits, is required to avoid problems in these future devices when a new driver uses (sets) some of the previously reserved bits. In this case, a following legacy driver must clear the bits.

All software that comes into contact with hardware registers in OMAP devices must follow these conventions.

7 Input/Output Descriptions

This section describes OMAP5910 inputs and outputs (I/O) and OMAP5910 functional multiplexing, which include:

- I/O signals ordered by their functions (Table 1)
- Functional multiplexing control bits for each ball (Table 2)

Consult the OMAP5910 Data Manual (literature number SPRS197) for additional information, including I/O pad reset status, buffer types and boundary scan, pullup/pulldown and gating/inhibiting information.

7.1 I/O Signals

Table 1 identifies the input and output signals for the OMAP5910 device.

Some signals are available on multiple pins via pin multiplexing configuration settings.

Table 1. Input and Output Signals for the OMAP5910 Device

Signals	Description	Ballout
FLASH		
$\overline{\text{FLASH.WP}}$	FLASH write protect	V4
$\overline{\text{FLASH.WE}}$	FLASH write enable	W2
$\overline{\text{FLASH.RP}}$	FLASH power down for TI/reset for Intel	W1
$\overline{\text{FLASH.OE}}$	FLASH output enable	U4
FLASH.D[15]	FLASH data bit 15	V3
FLASH.D[14]	FLASH data bit 14	T4
FLASH.D[13]	FLASH data bit 13	U3
FLASH.D[12]	FLASH data bit 12	U1
FLASH.D[11]	FLASH data bit 11	P8
FLASH.D[10]	FLASH data bit 10	T3
FLASH.D[9]	FLASH data bit 9	T2
FLASH.D[8]	FLASH data bit 8	R4
FLASH.D[7]	FLASH data bit 7	R3
FLASH.D[6]	FLASH data bit 6	R2
FLASH.D[5]	FLASH data bit 5	P7

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
FLASH (continued)		
FLASH.D[4]	FLASH data bit 4	P4
FLASH.D[3]	FLASH data bit 3	P2
FLASH.D[2]	FLASH data bit 2	N7
FLASH.D[1]	FLASH data bit 1	N2
FLASH.D[0]	FLASH data bit 0	N4
FLASH.CLK	FLASH clock	N3
$\overline{\text{FLASH.CS3}}$	FLASH chip select bit 3	N8
$\overline{\text{FLASH.CS2}}$	FLASH chip select bit 2	M4
$\overline{\text{FLASH.CS1}}$	FLASH chip select bit 1	M3
$\overline{\text{FLASH.CS0}}$	FLASH chip select bit 0	M7
$\overline{\text{FLASH.BE[1]}}$	FLASH byte enable bit 1	M8
$\overline{\text{FLASH.BE[0]}}$	FLASH byte enable bit 0	L3
$\overline{\text{FLASH.ADV}}$	FLASH address valid	L4
FLASH.A[24]	FLASH address bit 24	L7
FLASH.A[23]	FLASH address bit 23	K3
FLASH.A[22]	FLASH address bit 22	K4
FLASH.A[21]	FLASH address bit 21	L8
FLASH.A[20]	FLASH address bit 20	J1
FLASH.A[19]	FLASH address bit 19	J3
FLASH.A[18]	FLASH address bit 18	J4
FLASH.A[17]	FLASH address bit 17	J2
FLASH.A[16]	FLASH address bit 16	K7
FLASH.A[15]	FLASH address bit 15	H3
FLASH.A[14]	FLASH address bit 14	H4
FLASH.A[13]	FLASH address bit 13	K8
FLASH.A[12]	FLASH address bit 12	G2

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
FLASH (continued)		
FLASH.A[11]	FLASH address bit 11	G3
FLASH.A[10]	FLASH address bit 10	G4
FLASH.A[9]	FLASH address bit 9	F3
FLASH.A[8]	FLASH address bit 8	J7
FLASH.A[7]	FLASH address bit 7	E3
FLASH.A[6]	FLASH address bit 6	F4
FLASH.A[5]	FLASH address bit 5	D2
FLASH.A[4]	FLASH address bit 4	E4
FLASH.A[3]	FLASH address bit 3	C1
FLASH.A[2]	FLASH address bit 2	D3
FLASH.A[1]	FLASH address bit 1	J8
FLASH.RDY	FLASH ready for TI/wait for Intel	H7
$\overline{\text{FLASH.BAA}}$	FLASH burst advance acknowledge	M4
SDRAM		
$\overline{\text{SDRAM.WE}}$	SDRAM write enable	C3
$\overline{\text{SDRAM.RAS}}$	SDRAM row address srtobe	A2
$\overline{\text{SDRAM.DQMU}}$	SDRAM upper byte mask	D4
$\overline{\text{SDRAM.DQML}}$	SDRAM lower byte mask	B3
SDRAM.D[15]	SDRAM data bit 15	D5
SDRAM.D[14]	SDRAM data bit 14	C4
SDRAM.D[13]	SDRAM data bit 13	B4
SDRAM.D[12]	SDRAM data bit 12	D6
SDRAM.D[11]	SDRAM data bit 11	C5
SDRAM.D[10]	SDRAM data bit 10	H8
SDRAM.D[9]	SDRAM data bit 9	C6
SDRAM.D[8]	SDRAM data bit 8	B6

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
SDRAM (continued)		
SDRAM.D[7]	SDRAM data bit 7	D7
SDRAM.D[6]	SDRAM data bit 6	C7
SDRAM.D[5]	SDRAM data bit 5	D8
SDRAM.D[4]	SDRAM data bit 4	B8
SDRAM.D[3]	SDRAM data bit 3	G8
SDRAM.D[2]	SDRAM data bit 2	C8
SDRAM.D[1]	SDRAM data bit 1	G9
SDRAM.D[0]	SDRAM data bit 0	B9
SDRAM.CKE	SDRAM power down control signal	D9
SDRAM.CLK	SDRAM clock	C9
$\overline{\text{SDRAM.CAS}}$	SDRAM column address srtobe	H9
SDRAM.BA[1]	SDRAM bank select 1	D10
SDRAM.BA[0]	SDRAM bank select 0	C10
SDRAM.A[12]	SDRAM address bit 12	G10
SDRAM.A[11]	SDRAM address bit 11	H10
SDRAM.A[10]	SDRAM address bit 10	C11
SDRAM.A[9]	SDRAM address bit 9	D11
SDRAM.A[8]	SDRAM address bit 8	G11
SDRAM.A[7]	SDRAM address bit 7	C12
SDRAM.A[6]	SDRAM address bit 6	D12
SDRAM.A[5]	SDRAM address bit 5	H11
SDRAM.A[4]	SDRAM address bit 4	C13
SDRAM.A[3]	SDRAM address bit 3	D13
SDRAM.A[2]	SDRAM address bit 2	G12
SDRAM.A[1]	SDRAM address bit 1	C14
SDRAM.A[0]	SDRAM address bit 0	B14

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
LCD Interface		
LCD.VS	LCD vertical synchronization	D14
LCD.HS	LCD horizental synchronization	H12
LCD.AC	LCD ac-bias or output enable for connection to LCD	B15
LCD.PCLK	LCD pixel clock	C15
LCD.P[15]	LCD pixel data bit 15	D15
LCD.P[14]	LCD pixel data bit 14	C16
LCD.P[13]	LCD pixel data bit 13	A17
LCD.P[12]	LCD pixel data bit 12	G13
LCD.P[11]	LCD pixel data bit 11	B17
LCD.P[10]	LCD pixel data bit 10	C17
LCD.P[9]	LCD pixel data bit 9	D16
LCD.P[8]	LCD pixel data bit 8	D17
LCD.P[7]	LCD pixel data bit 7	C18
LCD.P[6]	LCD pixel data bit 6	B19
LCD.P[5]	LCD pixel data bit 5	A20
LCD.P[4]	LCD pixel data bit 4	H13
LCD.P[3]	LCD pixel data bit 3	G14
LCD.P[2]	LCD pixel data bit 2	C19
LCD.P[1]	LCD pixel data bit 1	B21
LCD.P[0]	LCD pixel data bit 0	D18
McBSP1 Interface		
MCBSP1.CLKS	McBSP1 clock input	G20
MBSP1.CLKX	McBSP1 bit clock	G21
MCBSP1.FSX	McBSP1 frame synchronization	H15, H18
MCBSP1.DX	McBSP1 data ouput	H18, H15
MCBSP1.DR	McBSP1 data input	H20

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
MCSI1 Interface		
MCSI1.DOUT	MCSI1 data output	W14
MCSI1.SYNC	MCSI1 frame synchronization	V13
MCSI1.CLK	MCSI1 bit clock	AA13
MCSI1.DIN	MCSI1 data input	W13
McBSP2 Interface		
MCBSP2.DR	McBSP2 data input	P10, AA5
MCBSP2.FSX	McBSP2 transmit frame synchronization	W7
MCBSP2.CLKR	McBSP2 receive clock	V7
MCBSP2.CLKX	McBSP2 transmit clock	Y6
MCBSP2.FSR	McBSP2 receive frame synchronization	W6
MCBSP2.DX	McBSP2 data output	AA5, P10
MCSI2 interface		
MCSI2.CLK	MCSI2 clock	Y10
MCSI2.DIN	MCSI2 data input	AA9
MCSI2.DOUT	MCSI2 data output	W9
MCSI2.SYNC	MCSI2 frame synchronization	V9
UART1 Interface		
UART1.RX	UART1 receive data	V14
UART1.TX	UART1 transmit data	Y14
UART1.RTS	UART1 request to send	AA15
UART1.CTS	UART1 clear to send	R14
UART1.DSR	UART1 data set ready	U18, R13
UART1.DTR	UART1 data terminal ready	W21, Y13
UART3 Interface		
UART3.RX	UART3 receive data	L14, K19
UART3.TX	UART3 transmit data	M18, K18

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
UART3 Interface (continued)		
UART3.RTS	UART3 request to send in UART mode SD_MODE in IRDA mode.	Y13, R19, K14
UART3.CTS	UART3 clear to send	R13, K15
UART3.DSR	UART3 data set ready	U18
UART3.DTR	UART3 data terminal ready	W21
UART2 Interface		
UART2.RX	UART2 receive data	R9, L14
UART2.TX	UART2 transmit data	V6, M18
UART2.RTS	UART2 request to send	W5, M19
UART2.CTS	UART2 clear to send	Y5,L15
UART2.BCLK	UART2 baud clock	Y4
Clock COM Interface		
MCLK	M-CLK (master clock) output (12 MHz)	Y9
MCLKREQ	Request for the M-CLK	R10
BCLK	BCLK general-purpose clock output	Y13
BCLKREQ	BCLK request input	R13
GPIO		
GPIO15	General purpose I/O 15	M20
GPIO14	General purpose I/O 14	N21
GPIO13	General purpose I/O 13	N19
GPIO12	General purpose I/O 12	N18, W6
GPIO11	General purpose I/O 11	N20, V7
GPIO9	General purpose I/O 9	W8
GPIO8	General purpose I/O 8	Y8
GPIO7	General purpose I/O 7	M15, Y5, V9
GPIO6	General purpose I/O 6	P19
GPIO4	General purpose I/O 4	P20

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
GPIO (continued)		
GPIO3	General purpose I/O 3	P18
GPIO2	General purpose I/O 2	M14
GPIO1	General purpose I/O 1	R19
GPIO0	General purpose I/O 0	R18
MPU I/O		
MPUIO12	MPU input/output 12	L19
MPUIO11	MPU input/output 11	W10
MPUIO7	MPU input/output 7	V10
MPUIO6	MPU input/output 6	W11
MPUIO5	MPU input/output 5	T20, W5
MPUIO4	MPU input/output 4	T19
MPUIO3	MPU input/output 3	V8
MPUIO2	MPU input/output 2	N15
MPUIO1	MPU input/output 1	U19
MPUIO0	MPU input/output 0	Y12
I^2C Interface		
SCL	I^2C master serial clock	T18
SDA	I^2C serial bidirectional data	V20
UWIRE Interface		
UWIRE.SDI	UWIRE serial data input	U18, J14
UWIRE.SDO	UWIRE serial data output	W21, H19
UWIRE.SCLK	UWIRE serial clock	V19, J15
$\overline{\text{UWIRE.CS0}}$	UWIRE serial chip select 0	N14, J18
$\overline{\text{UWIRE.CS3}}$	UWIRE serial chip select 3	P15, J19

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
McBSP3 Interface		
MCBSP3.DR	McBSP3 data input	AA17, U18
MCBSP3.DX	McBSP3 data output	P14, W21
MCBSP3.FSX	McBSP3 frame synchronization	N18, P18, P19, P20
MCBSP3.CLKX	McBSP3 clock	W16, N14
Miscelleanous		
MPU_BOOT	MPU boot mode	AA17
RST_HOST_OUT	Reset signal	P14
CLK32K_IN	32 kHz clock input	P13
CLK32K_OUT	32 kHz clock output	Y12
CLK32K_CTRL	32 kHz clock Selection	AA20
$\overline{\text{RST\_OUT}}$	Global reset output of MPU subsystem	W15
$\overline{\text{PWRON\_RESET}}$	Power on reset	G19
$\overline{\text{MPU\_RST}}$	Warm boot reset to TI925T only	V15
$\overline{\text{EXT\_DMA\_REQ1}}$	External DMA request 1	T19
$\overline{\text{EXT\_DMA\_REQ0}}$	External DMA request 0	N15
LOW_PWR	Low power request	T20
BFAIL/$\overline{\text{EXT\_FIQ}}$	Battery voltage failure detection and/or external FIQ input	W19
IRQ_OBS	Interrupt observability output	M18
DMA_REQ_OBS	DMA request observability output	L14
USB Integrated Transceiver Pins		
USB.DP	USB differential (+) line	P9
USB.DM	USB differential (–) line	R8
USB.PUEN	USB pullup enable	W4
USB.CLKO	USB clock output	W4
USB.VBUS	USB VBUS detect input	R18

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
Camera Interface		
CAM.EXCLK	Camera clock output	H19
CAM.LCLK	Camera image data latch clock	J15
CAM.D[7]	Camera digital image data bit 7	J18
CAM.D[6]	Camera digital image data bit 6	J19
CAM.D[5]	Camera digital image data bit 5	J14
CAM.D[4]	Camera digital image data bit 4	K18
CAM.D[3]	Camera digital image data bit 3	K19
CAM.D[2]	Camera digital image data bit 2	K15
CAM.D[1]	Camera digital image data bit 1	K14
CAM.D[0]	Camera digital image data bit 0	L19
CAM.VS	Camera vertical synchronization	L18
CAM.HS	Camera horizontal synchronization	L15
CAM.RSTZ	Camera module reset	M19
MMC/SD Interface		
MMC.DAT3	SD card data bit 3	W11
MMC.DAT2	SD card data bit 2	W10, M15
MMC.DAT1	SD card data bit 1	V10
MMC.DAT0_SPI.DI	MMC or SD card data bit 0/SPI serial input	R11
MMC.CLK	MMC/SD clock	V11
MMC.CMD_SPI.DO	MMC/SD command/SPI serial output	P11
SPI.CS3	SPI chip select 3	P18
SPI.CS2	SPI chip select 2	P20
SPI.CS1	SPI chip select 1	P19
SPI.RDY	SPI ready	R18
SPI.CLK	SPI clock	M14

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
Oscillators		
OSC1_IN	12 MHz quartz connection (XI)	Y2
OSC1_OUT	12 MHz quartz connection (XO)	W3
OSC32K_IN	32 kHz quartz connection (XI)	W12
OSC32K_OUT	32 kHz quartz connection (XO)	R12
Configuration Interface		
CONF	OMAP5910 configuration Input	V18
STAT_VAL/$\overline{\text{WKUP}}$	MMC select/wakeup input	Y17
Keyboard Interface		
KB.C[7]	Keyboard matrix column 7	V19
KB.C[6]	Keyboard matrix column 6	P15
KB.C[5]	Keyboard matrix column 5	C20
KB.C[4]	Keyboard matrix column 4	C21
KB.C[3]	Keyboard matrix column 3	E18
KB.C[2]	Keyboard matrix column 2	D19
KB.C[1]	Keyboard matrix column 1	D20
KB.C[0]	Keyboard matrix column 0	F18
KB.R[7]	Keyboard matrix row 7	M20
KB.R[6]	Keyboard matrix row 6	N21
KB.R[5]	Keyboard matrix row 5	N19
KB.R[4]	Keyboard matrix row 4	E19
KB.R[3]	Keyboard matrix row 3	E20
KB.R[2]	Keyboard matrix row 2	H14
KB.R[1]	Keyboard matrix row 1	F19
KB.R[0]	Keyboard matrix row 0	G18

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
JTAG Interface		
TDI	Test data input	Y19
TDO	Test data output	AA19
TMS	Test mode select	V17
TCK	Test clock	W18
$\overline{\text{TRST}}$	Test reset	Y18
EMU0	Test emulation 0	V16
EMU1	Test emulation 1	W17
Trace Interface		
ETM.SYNC	ETM9 trace synchronization	H19
ETM.CLK	ETM9 trace clock	J15
ETM.D[7]	ETM9 trace packet bit 7	J18
ETM.D[6]	ETM9 trace packet bit 6	J19
ETM.D[5]	ETM9 trace packet bit 5	J14
ETM.D[4]	ETM9 trace packet bit 4	K18
ETM.D[3]	ETM9 trace packet bit 3	K19
ETM.D[2]	ETM9 trace packet bit 2	K15
ETM.D[1]	ETM9 trace packet bit 1	K14
ETM.D[0]	ETM9 trace packet bit 0	L19
ETM.PSTAT[2]	ETM9 trace pipe state bit 2	L18
ETM.PSTAT[1]	ETM9 trace pipe state bit 1	L15
ETM.PSTAT[0]	ETM9 trace pipe state bit 0	M19
HDQ/1-Wire Interface		
HDQ	HDQ/1-Wire interface pin	N20
LED Pulse Generators		
LED1	LED pulse generator 1 output	P18
LED2	LED pulse generator 2 output	T19

Table 1. Input and Output Signals for the OMAP5910 Device (Continued)

Signals	Description	Ballout
PWM Interface		
PWT	Pulse width tone	M18
PWL	Pulse width light	L14
USB Pin Group #1		
USB1.SUSP	USB 1 bus segment suspend control	AA17
USB1.VM	USB 1 Vminus receive data	AA13
USB1.SE0	USB 1 single ended zero	P14
USB1.TXEN	USB 1 transmit enable	W16
USB1.SPEED	USB 1 bus segment speed control	Y12
USB1.VP	USB 1 Vplus receive data	V13
USB1.TXD	USB 1 transmit data	W14
USB1.RCV	USB 1 receive data	W13
USB Pin Group #2		
USB2.VP	USB 2 Vplus receive data	AA9
USB2.VM	USB 2 Vminus receive data	R9
USB2.SE0	USB 2 single ended zero	W5
USB2.TXEN	USB 2 transmit enable	W9
USB2.SUSP	USB 2 bus segment suspend control	Y10
USB2.RCV	USB 2 receive data	Y5
USB2.TXD	USB 2 transmit data	V6

7.2 I/O Functional Multiplexing

Table 2 provides the input/output configuration programming for signal multiplexing on each pin and for enabling/disabling internal pullup and pulldown resistors on each pin.

Table 2. Configuration Programming

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
A1								
B2								
C3								
A2								
D4								
B3								
A3								
D5								
C4								
B4								
D6								
C5								
B5								
A5								
H8								
C6								
B6								
D7								
C7								
B7								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
A7								
G8								
D8								
B8								
C8								
G9								
B9								
D9								
A9								
C9								
B10								
H9								
D10								
C10								
G10								
H10								
A11								
C11								
D11								
G11								
C12								
D12								
H11								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
B12								
A13								
C13								
D13								
B13								
G12								
C14								
B14								
A15								
D14								
H12								
B15								
C15								
D15								
B16								
C16								
A17								
G13								
B17								
C17								
D16								
B18								
D17								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
A19								
C18								
B19								
A20								
H13								
G14								
A21								
B20								
C19								
B21								
D18								
C20								
C21								
E18								
D19								
D20								
F18								
E19								
E20								
E21								
H14								
F19								
F20								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
G18								
G19								
G20								
G21								
H15	FUNC_MUX_CTRL_4	0x14	14:12	000	MCBSP1.FSX			
				001	MCBSP1.DX			
H18	FUNC_MUX_CTRL_4	0x14	17:15	000	MCBSP1.DX			
				001	MCBSP1.FSX			
H20						PULL_DOWN_CTRL_0	0x40	16
H19	FUNC_MUX_CTRL_4	0x14	23:21	000	CAM.EXCLK			
				001	ETM.SYNC			
				010	UWIRE.SDO			
J15	FUNC_MUX_CTRL_4	0x14	26:24	000	CAM.LCLK			
				001	ETM.CLK			
				010	UWIRE.SCLK			
J20								
J18	FUNC_MUX_CTRL_4	0x14	29:27	000	CAM.D[7]			
				001	ETM.D[7]			
				010	$\overline{\text{UWIRE.CS0}}$			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
J21								
J19	FUNC_MUX_ CTRL_5	0x18	2:0	000	CAM.D[6]			
				001	ETM.D[6]			
				010	$\overline{\text{UWIRE.CS3}}$			
K20								
J14	FUNC_MUX_ CTRL_5	0x18	5:3	000	CAM.D[5]	PULL_DOWN_ CTRL_0	0x40	21
				001	ETM.D[5]			
				010	UWIRE.SDI			
K18	FUNC_MUX_ CTRL_5	0x18	8:6	000	CAM.D[4]			
				001	ETM.D[4]			
				010	UART3.TX			
K19	FUNC_MUX_ CTRL_5	0x18	11:9	000	CAM.D[3]	PULL_DOWN_ CTRL_0	0x40	23
				001	ETM.D[3]			
				010	UART3.RX			
K15	FUNC_MUX_ CTRL_5	0x18	14:12	000	CAM.D[2]	PULL_DOWN_ CTRL_0	0x40	24
				001	ETM.D[2]			
				010	UART3.CTS			
K14	FUNC_MUX_ CTRL_5	0x18	17:15	000	CAM.D[1]			
				001	ETM.D[1]			
				010	UART3.RTS			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
L21								
L19	FUNC_MUX_CTRL_5	0x18	20:18	000	CAM.D[0]			
				001	ETM.D[0]			
				010	MPUIO12			
L18	FUNC_MUX_CTRL_5	0x18	23:21	000	CAM.VS			
				001	ETM.PSTAT[2]			
L15	FUNC_MUX_CTRL_5	0x18	26:24	000	CAM.HS	PULL_DOWN_CTRL_0	0x40	28
				001	ETM.PSTAT[1]			
				010	UART2.CTS			
M19	FUNC_MUX_CTRL_5	0x18	29:27	000	CAM.RSTZ			
				001	ETM.PSTAT[0]			
				010	UART2.RTS			
M18	FUNC_MUX_CTRL_6	0x1C	2:0	000	UART3.TX			
				001	UART3.TX			
				010	PWT			
				011	Reserved			
				100	UART2.TX			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
L14	FUNC_MUX_CTRL_6	0x1C	5:3	000	UART3.RX			
				001	PWL			
				010	Reserved			
				011	UART2.RX			
M20	FUNC_MUX_CTRL_6	0x1C	8:6	000	GPIO15	PULL_DOWN_CTRL_1	0x44	0
				001	KB.R[7]			
N21	FUNC_MUX_CTRL_6	0x1C	11:9	000	GPIO14	PULL_DOWN_CTRL_1	0x44	1
				001	KB.R[6]			
N19	FUNC_MUX_CTRL_6	0x1C	14:12	000	GPIO13	PULL_DOWN_CTRL_1	0x44	2
				001	KB.R[5]			
N18	FUNC_MUX_CTRL_6	0x1C	17:15	000	GPIO12	PULL_DOWN_CTRL_1	0x44	3
				001	MCBSP3.FSX			
N20	FUNC_MUX_CTRL_6	0x1C	20:18	000	GPIO11	PULL_DOWN_CTRL_1	0x44	4
				001	HDQ			
M15	FUNC_MUX_CTRL_6	0x1C	23:21	000	GPIO7	PULL_DOWN_CTRL_1	0x44	5
				001	MMC.DAT2			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
P19	FUNC_MUX_CTRL_6	0x1C	26:24	000	GPIO6	PULL_DOWN_CTRL_1	0x44	6
				001	SPI.CS1			
				010	MCBSP3.FSX			
P20	FUNC_MUX_CTRL_6	0x1C	29:27	000	GPIO4	PULL_DOWN_CTRL_1	0x44	7
				001	SPI.CS2			
				010	MCBSP3.FSX			
R21								
P18	FUNC_MUX_CTRL_7	0x20	2:0	000	GPIO3	PULL_DOWN_CTRL_1	0x44	8
				001	SPI.CS3			
				010	MCBSP3.FSX			
				011	LED1			
M14	FUNC_MUX_CTRL_7	0x20	5:3	000	GPIO2	PULL_DOWN_CTRL_1	0x44	9
				001	SPI.CLK			
R20								
R19	FUNC_MUX_CTRL_7	0x20	8:6	000	GPIO1	PULL_DOWN_CTRL_1	0x44	10
				001	UART3.RTS			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
R18	FUNC_MUX_CTRL_7	0x20	11:9	000	GPIO0	PULL_DOWN_CTRL_1	0x44	11
				001	SPI.RDY			
				010	USB.VBUS			
T20	FUNC_MUX_CTRL_7	0x20	14:12	000	MPUIO5	PULL_DOWN_CTRL_1	0x44	12
				001	LOW_PWR			
T19	FUNC_MUX_CTRL_7	0x20	17:15	000	MPUIO4	PULL_DOWN_CTRL_1	0x44	13
				001	$\overline{\text{EXT\_DMA\_REQ0}}$			
				010	LED2			
U21								
N15	FUNC_MUX_CTRL_7	0x20	20:18	000	MPUIO2	PULL_DOWN_CTRL_1	0x44	14
				001	$\overline{\text{EXT\_DMA\_REQ0}}$			
U20								
U19								
T18								
V20								
U18	FUNC_MUX_CTRL_8	0x24	2:0	000	UWIRE.SDI	PULL_DOWN_CTRL_1	0x44	18
				001	UART3.DSR			
				010	UART1.DSR			
				011	MCBSP3.DR			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
W21	FUNC_MUX_CTRL_8	0x24	5:3	000	UWIRE.SDO			
				001	UART3.DTR			
				010	UART1.DTR			
				011	MCBSP3.DX			
V19	FUNC_MUX_CTRL_8	0x24	8:6	000	UWIRE.SCLK			
				001	KB.C[7]			
W20								
Y21								
N14	FUNC_MUX_CTRL_8	0x24	11:9	000	$\overline{\text{UWIRE.CS0}}$			
				001	$\overline{\text{UWIRE.CS0}}$			
				010	MCBSP3.CLKX			
P15	FUNC_MUX_CTRL_8	0x24	14:12	000	$\overline{\text{UWIRE.CS3}}$			
				001	$\overline{\text{UWIRE.CS3}}$			
				010	KB.C[6]			
AA21								
Y20								
W19								
AA20								
V18						PULL_DOWN_CTRL_3	0x4C	9

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
Y19						PULL_DOWN_CTRL_3	0x4C	10
AA19								
V17						PULL_DOWN_CTRL_3	0x4C	11
W18						PULL_DOWN_CTRL_3	0x4C	12
Y18						PULL_DOWN_CTRL_3	0x4C	13
V16						PULL_DOWN_CTRL_1	0x44	25
W17						PULL_DOWN_CTRL_1	0x44	26
Y17								
AA17	FUNC_MUX_CTRL_8	0x24	29:27	000	MPU_BOOT	PULL_DOWN_CTRL_1	0x44	27
				001	MCBSP3.DR			
				010	USB1_SUSP			
P14	FUNC_MUX_CTRL_9	0x28	2:0	000	RST_HOST_OUT			
				001	MCBSP3.DX			
				010	USB1_SE0			
W16	FUNC_MUX_CTRL_9	0x28	5:3	000	MCBSP3.CLKX	PULL_DOWN_CTRL_1	0x44	29
				001	MCBSP3.CLKX			
				010	USB1_TXEN			
Y16								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
V15								
W15								
Y15								
AA15	FUNC_MUX_ CTRL_9	0x28	14:12	000	UART1.RTS			
				001	UART1.RTS			
R14						PULL_DOWN_ CTRL_2	0x48	1
V14						PULL_DOWN_ CTRL_2	0x48	2
Y14	FUNC_MUX_ CTRL_9	0x28	23:21	000	UART1.TX			
				001	UART1.TX			
W14	FUNC_MUX_ CTRL_9	0x28	26:24	000	MCSI1. DOUT			
				001	USB1.TXD			
R13	FUNC_MUX_ CTRL_9	0x28	29:27	000	UART3. CLKREQ	PULL_DOWN_ CTRL_2	0x48	5
				001	UART3.CTS			
				010	UART1.DSR			
Y13	FUNC_MUX_ CTRL_A	0x2C	2:0	000	UART3. BCLK			
				001	UART3.RTS			
				010	UART1. DTR			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
V13	FUNC_MUX_CTRL_A	0x2C	5:3	000	MCSI1.SYNC	PULL_DOWN_CTRL_2	0x48	7
				001	USB1.VP			
AA13	FUNC_MUX_CTRL_A	0x2C	8:6	000	MCSI1.CLK	PULL_DOWN_CTRL_2	0x48	8
				001	USB1.VM			
W13	FUNC_MUX_CTRL_A	0x2C	11:9	000	MCSI1.DIN	PULL_DOWN_CTRL_2	0x48	9
				001	USB1.RCV			
Y12	FUNC_MUX_CTRL_A	0x2C	14:12	000	CLK32K_OUT			
				001	MPUIO0			
				010	USB1.SPEED			
P13								
V12								
W12								
R12								
P12								
AA11								
W11	FUNC_MUX_CTRL_D	0x38	14:12	000	MMC.DAT3	PULL_DOWN_CTRL_3	0x4C	8
				001	ms_remove			
				010	MPUIO6			
V11								
R11								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
W10	FUNC_MUX_CTRL_A	0x2C	20:18	000	MMC.DAT2	PULL_DOWN_CTRL_2	0x48	12
				001	Reserved			
				010	MPUIO11			
V10	FUNC_MUX_CTRL_A	0x2C	26:24	000	MMC.DAT1	PULL_DOWN_CTRL_2	0x48	14
				001	ms_ins			
				010	MPUIO7			
P11						PULL_DOWN_CTRL_2	0x48	15
Y10	FUNC_MUX_CTRL_B	0x30	5:3	000	MCSI2.CLK	PULL_DOWN_CTRL_2	0x48	17
				001	USB2.SUSP			
AA9	FUNC_MUX_CTRL_B	0x30	8:6	000	MCSI2.DIN	PULL_DOWN_CTRL_2	0x48	18
				001	USB2.VP			
W9	FUNC_MUX_CTRL_B	0x30	11:9	000	MCSI2.DOUT			
				001	USB2.TXEN			
V9	FUNC_MUX_CTRL_B	0x30	14:12	000	MCSI2.SYNC	PULL_DOWN_CTRL_2	0x48	20
				001	GPIO7			
Y9								
R10	FUNC_MUX_CTRL_B	0x30	20:18	000	UART2.CLKREQ	PULL_DOWN_CTRL_2	0x48	22
				001	EXT_MASTER_REQ			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
W8						PULL_DOWN_CTRL_2	0x48	23
Y8						PULL_DOWN_CTRL_2	0x48	24
AA7								
V8						PULL_DOWN_CTRL_2	0x48	25
P10	FUNC_MUX_CTRL_C	0x34	2:0	000	MCBSP2.DR	PULL_DOWN_CTRL_2	0x48	26
				001	MCBSP2.DX			
Y7								
W7						PULL_DOWN_CTRL_2	0x48	27
V7	FUNC_MUX_CTRL_C	0x34	8:6	000	MCBSP2.CLKR	PULL_DOWN_CTRL_2	0x48	28
				001	GPIO11			
Y6						PULL_DOWN_CTRL_2	0x48	29
W6	FUNC_MUX_CTRL_C	0x34	14:12	000	MCBSP2.FSR	PULL_DOWN_CTRL_2	0x48	30
				001	GPIO12			
AA5	FUNC_MUX_CTRL_C	0x34	17:15	000	MCBSP2.DX	PULL_DOWN_CTRL_2	0x48	31
				001	MCBSP2.DR			
R9	FUNC_MUX_CTRL_C	0x34	20:18	000	UART2.RX	PULL_DOWN_CTRL_3	0x4C	0
				001	USB2.VM			

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
Y5	FUNC_MUX_CTRL_C	0x34	23:21	000	UART2.CTS	PULL_DOWN_CTRL_3	0x4C	1
				001	USB2.RCV			
				010	GPIO7			
W5	FUNC_MUX_CTRL_C	0x34	26:24	000	UART2.RTS			
				001	UART2.RTS			
				010	USB2.SE0			
				011	MPUIO5			
V6	FUNC_MUX_CTRL_C	0x34	29:27	000	UART2.TX			
				001	UART2.TX			
				010	USB2.TXD			
Y4								
V5								
AA3								
W4	FUNC_MUX_CTRL_D	0x38	5:3	000	USB.PUEN			
				001	USB.CLKO			
Y3								
AA2								
P9								
R8								
AA1								
Y2								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
W3								
Y1								
V4								
W2								
W1								
U4								
V3								
V2								
T4								
U3								
U2								
U1								
P8								
T3								
T2								
R4								
R3								
R2								
R1								
P7								
P4								
P2								
P3								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
N7								
N2								
N4								
N1								
N3								
M2								
N8								
M4	FUNC_MUX_CTRL_D	0x38	8:6	000	$\overline{\text{FLASH.CS2}}$			
				001	$\overline{\text{FLASH.BAA}}$			
M3								
M7								
M8								
L1								
L3								
L4								
L7								
K3								
K4								
L8								
K2								
J1								
J3								
J4								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
J2								
K7								
H3								
H2								
G1								
H4								
K8								
G2								
G3								
G4								
F2								
F3								
E1								
J7								
E2								
E3								
F4								
D2								
E4								
C1								
D3								
C2								
B1								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

Table 2. Configuration Programming (Continued)

Ballout	Pin-by-Pin Multiplexing OMAP5910 Configuration Register	Register Offset	Register Field	Value	Signal	Pin-by-Pin Pullup/Down OMAP5910 Configuration Register	Register Offset	Register Field
J8								
H7								
E5								

Note: When a row is empty, it means that there is no:
- Functional multiplexing on this pin
- Software configurable pullup/pulldown on this pin

8 Switching Clock Modes

This section describes the programming guidelines for switching clock modes in the OMAP5910 device.

8.1 Switching Procedure

Perform the following procedure to switch OMAP5910 clock modes:

1) Make sure the DSP clock is enabled.
2) Make sure there are no active transfers on interfaces (EMIFS, EMIFF, TIPB, IMIF, MPUI, etc.).
3) Make sure there are no active DSP transactions being performed.
4) Disable the MPU D-cache/MMU.
5) Make sure the MPU clock control register (ARM_CKCTL), clock dividers, and the DPLL_REG have correct contents for the clock mode the system is being switched to.
    a) Switch from SYNC mode to SYNCSCALE mode:
        i) Make sure that the frequency of the traffic controller is always less than the maximum frequency of the traffic controller.
        ii) Change the clock mode.
        iii) Program the clock dividers.
        iv) Program DPLL to frequency desired.
    b) Switch from SYNCSCALE to SYNC mode:
        i) Program the DPLL to the desired frequency in synchronous mode.
        ii) Program all clock dividers to be equal.
        iii) Change the clock mode to SYNC mode.
6) After the MPU write to the MPU system status register (ARM_SYSST) (0x18) to switch modes, there must be no requests from the MPU to the traffic controller for the next 100 MPU cycles (see Section 8.2, *Main Code*, and Section 8.3, *Delay Procedure*).
7) Make sure all read and write accesses to the clock reset registers are 16-bit accesses.

8.2 Main Code

The following is the main code for switching modes:

```
main()
{
   …
…..
// Enable Icache
   INT_SetSupervisor();
   ARM_WRITE_REG1(I_bit);
   INT_SetUser();
// Enable DSP Clock
   MCU_CKCTL = 0x2000;
   switch_mode(CLOCK_MODE_SYNC_SCALE);
// Passing in 0x1000
….
….
}
```

8.3 Delay Procedure

Use the following software routine to create a delay of 100 clock cycles after a switch mode write.

Ensure the I-cache is enabled during switching modes.

```
      state16               ; thumb mode
      .ref  edata           ; defined by armas
      .global $switch_mode
$switch_mode:
       push    {lr}
       push    {r1-r7}
       adr     r4, into_32_bis
       bx      r4
   nop
   nop
   nop
       .state32                ; arm mode
into_32_bis:
       ;
      LDR   R1,ARM_SYSST
      MOV   R3,#0
      MOV   R2,#0
; This is the loop that will wait for at least 100 cycles
; before issuing next request from MPU. On the first run
of the loop only Icache
; gets loaded with the loop and the next 2 instructions
but write to SYSST does not occur
; In the 2nd run of the loop only write to SYSST happens
and after that MPU runs the loop from
;Icache so no request goes out
```

```
LOOP      CMP   R2,#1
          STREQ R0,[R1]
          ADD   R2,R2,#1
          CMP   R2,#16
          BNE   LOOP
the_end:
          adr   r2, into_16_bis + 1
          bx    r2
   .state16
into_16_bis:
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       pop   {r1-r7}
       pop   {pc}
;*********************************************************
;* CONSTANT TABLE                                        *
;*********************************************************
ARM_SYSST       .long 0xFFFECE18
```

9 Peripheral Overview

The user-accessible peripherals available on the OMAP5910 devices are configured using a set of memory-mapped control registers. The peripheral bus controller performs the arbitration for accesses of on-chip peripherals.

Peripherals available on the OMAP5910 devices, the page number in this document on which a short description of each peripheral can be found, and the peripheral's associated literature number are listed in Table 3.

Table 3. OMAP5910 Dual-Core Processor DSP Peripherals Documentation

Peripheral	Lit #	Page #
MPU Subsystem	SPRU671	57
DSP Subsystem	SPRU672	58
Memory Interface Traffic Controller	SPRU673	58
System DMA Controller	SPRU674	58
LCD Controller	SPRU675	58
Universal Asynchronous Receiver/Transmitter (UART) Devices	SPRU676	59
Universal Serial Bus (USB) and Frame Adjustment Counter (FAC)	SPRU677	60
Clock Generation and System Reset Management	SPRU678	61
General-Purpose Input/Output (GPIO)	SPRU679	61
MMC/SD	SPRU680	62
Inter-Intergrated Circuit (I2C) Controller	SPRU681	62
Timer Reference	SPRU682	63
Inter-Processor Communication	SPRU683	64
Camera Interface	SPRU684	64
Multichannel Serial Interface (MCSI)	SPRU685	64
Micro-Wire Interface	SPRU686	65
Real-Time Clock (RTC)	SPRU687	65
HDQ/1-Wire Interface	SPRU688	65
PWL, PWT, and LED Peripheral	SPRU689	66
Multichannel Buffered Serial Port (McBSP)	SPRU708	67

10 MPU Subsystems

The MPU of the OMAP5910 device controls the memory management units (MMUs), the system direct memory access (DMA) controller, the MPU TI peripheral bus (TIPB) bridge, and peripherals.

- ❏ MPU core (see the MPU Core section)
- ❏ Traffic controller (see SPRU673, *OMAP5910 Dual-Core Processor Memory Interface Traffic Controller Reference Guide*)
- ❏ MPU MMU (see SPRU671, *OMAP5910 Dual-Core Processor MPU Subsystems Reference Guide*)
- ❏ DSP MMU (see SPRU671, *OMAP5910 Dual-Core Processor MPU Subsystems Reference Guide*)
- ❏ System DMA controller (see SPRU674, *OMAP5910 Dual-Core Processor System DMA Controller Reference Guide*)
- ❏ LCD controller (see SPRU675, *OMAP5910 Dual-Core Processor LCD Controller Reference Guide*)
- ❏ MPU TIPB bridge (see SPRU671, *OMAP5910 Dual-Core Processor MPU Subsystems Reference Guide*)
- ❏ Clock manager (see SPRU678, *OMAP5910 Dual-Core Processor Clock Generation and System Reset Management Reference Guide*)
- ❏ Interrupt handler (see SPRU671, *OMAP5910 Dual-Core Processor MPU Subsystems Reference Guide*)
- ❏ Timers (see SPRU682, *Timer Reference Guide*)
- ❏ Watchdog timer (see SPRU682, *OMAP5910 Dual-Core Processor Timer Reference Guide*)
- ❏ Interprocessor communication
  (see SPRU683, *OMAP5910 Dual-Core Processor Interprocessor Communication Reference Guide*)
- ❏ 1.5M-bit SRAM internal memory

11 DSP Subsystem

The digital signal processor (DSP) subsystem is built around a core processor and peripherals that interface with:

- The TI925T via the microprocessor unit interface (MPUI)
- Various standard memories via the external memory interface (EMIF)
- Various system peripherals via the TI peripheral bus (TIPB) bridge

12 Memory Interface Traffic Controller

The memory interface traffic controller (TC) manages all accesses by the MPU, the TMS320C55x DSP, the system DMA, and the local bus to the OMAP5910 system memory resources (SRAM, SDRAM, flash, ROM, etc.). The TC also manages accesses by the MPU or the USB host. The USB host is an internal OMAP5910 peripheral connected on the local bus, so the TC contributes to managing USB host accesses.

13 System DMA Controller

The system direct memory access (DMA) controller transfers data between points in the memory space without intervention by the MPU. The DMA allows movements of data to and from internal memory, external memory, and peripherals to occur in the background of MPU operation. It is designed to off-load the block data transfer function from the MPU processor.

14 LCD Controller

The OMAP5910 device includes an LCD controller that interfaces with most industry-standard LCD displays. The LCD controller operates only in single-panel mode (dual-panel mode is not supported). The module is designed to work with a separate RAM block to provide data to the FIFO at the front end of the LCD controller data path at a rate sufficient to support the chosen display mode and resolution.

The panel size is programmable, and can be any width (line length) from 16 to 1024 pixels in 16 pixel increments. The number of lines is set by programming the total number of pixels in the LCD. The total frame size is programmable up to 1024×1024.

The screen is intended to be mapped to the frame buffer as one contiguous block where each horizontal line of pixels is mapped to a set of consecutive bytes of words in the frame memory.

Frame sizes and frame rates supported in specific applications depend upon the available memory bandwidth allowed by the application.

The principal features of the LCD controller are:

- A dedicated 64-entry x 16-bit FIFO
- A dedicated LCD DMA channel for LCD display
- A programmable display including support for 2-, 4-, 8-, 12-, and 16-bit graphics modes.
- A programmable display resolutions up to 1024 pixels by 1024 lines
- Support for passive monochrome (STN) displays
- Support for passive color (STN) displays
- Support for active color (TFT) displays
- A patented dithering algorithm, providing:
  - 15 grayscale levels for monochrome passive displays
  - 3375 colors for color passive displays
- 65536 colors for active color displays
- A 256-entry x 12-bit palette
- A programmable pixel rate
- A pixel clock plus horizontal and vertical synchronization signals
- An ac-bias drive signal
- An active display enable signal

15 Universal Asynchronous Receiver/Transmitter (UART) Devices

The OMAP5910 multimedia processor contains three universal asynchronous receiver/transmitter (UART) peripherals. UART1 and UART2 are UART modems with autobaud capability. UART3 is a modem with IrDA. Either the MPU (default) or the DSP controls the three UARTs via three TIPB switches (one for each UART).

16 Universal Serial Bus (USB) and Frame Adjustment Counter (FAC)

The OMAP5910 USB host controller (HC) is a three-port controller that communicates with USB devices at the USB low-speed (1.5M bit/s maximum) and full-speed (12M bit/s maximum) data rates. It is compatible with the *Universal Serial Bus Specification Revision 1.1* and the *Open HCI—Open Host Controller Interface Specification for USB,* Release 1.0a, available through the Compaq Computer Corporation web site, and hereafter called the *OHCI Specification for USB.* It is assumed that users of the OMAP5910 USB host controller are already familiar with the *USB Specification* and *OHCI Specification for USB*.

The OMAP5910 USB host controller implements the register set and makes use of the memory data structures defined in the *OHCI Specification for USB*. These registers and data structures are the mechanism by which a USB host controller driver software package can control the OMAP5910 USB host controller. The *OHCI Specification for USB* also defines how the USB host controller implementation must interact with those registers and data structures in system memory.

To reduce processor software and interrupt overhead, the USB host controller generates USB traffic based on data structures and data buffers stored in system memory. The OMAP5910 USB host controller accesses these data structures without direct intervention by the processor using the OMAP5910 local bus. These data structures and data buffers can be located in internal or external system RAM. The local bus MMU allows the USB host controller to access the full address range of internal and external memories.

The OMAP5910 USB host controller is connected to the OMAP5910 MPU public peripheral bus to enable MPU access to registers. The USB host controller gains access to the data structures in the OMAP5910 system memory via the internal OMAP5910 local bus (LB) interface. The USB host controller provides an interrupt to the MPU level 2 interrupt handler to signal certain hardware events to the host controller driver software.

Flexible multiplexing of signals from the OMAP5910 USB host controller, the OMAP5910 USB function controller, and other OMAP5910 peripherals allows a wide variety of system-level USB functions. The OMAP5910 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 OMAP5910 USB signal multiplexing selects how the signals associated with the three OMAP5910 USB host ports and the OMAP5910 USB function controller can be brought out to OMAP5910 pins.

17 Clock Generation and System Reset

The clock generator and reset management module supplies clocks and resets to the entire OMAP5910 device.

In the OMAP5910 device, clock generation and system reset are controlled by several modules.

There are three major components of this circuitry: the ultralow-power device (ULPD), the reset-management module, and the clock-generation and management module.

18 General-Purpose Input/Output (GPIO)

The GPIOs are programmable inputs or outputs. They generate a level interrupt, the sources for which can be masked from within the GPIO module. Under software control, the GPIOs can be individually dedicated to the DSP or the MPU.

GPIOs are accessible on general-purpose input and output external pins available to the user for system-level control and general-purpose functions. The signals are user-defined as either input or output. The output state can be controlled and inputs can be configured to provide an interrupt.

The MPU and the DSP have separate instances of the GPIO registers, but share the same device pins. The determination of whether MPU or DSP has control of the device pin is controlled by a single shared pin control register (at offset 0x18). This register is shown as the configuration and control register. This register is read/write from/to the MPU, but read-only from the DSP. The MPU is responsible for writing to this register to assign any necessary GPIO signals to the DSP. By default, all GPIOs are assigned to the MPU.

GPIO interrupts are routed to both the MPU and DSP interrupt handlers, but a GPIO can only signal an interrupt to the processor to which it is assigned in the pin control register. By default, GPIO interrupts are disabled at both of the interrupt handlers.

There are no I/O signals associated with GPIO.5 and GPIO.10.

19 MMC/SD

The MMC/SD host controller provides an interface between the MPU and either the MMC or SD memory card plus up to three serial flash cards and handles the MMC/SD or SPI transactions with minimal MPU intervention.

The following combination of external devices is supported:

- One or more MMC memory cards sharing the same bus plus up to three devices with 8-bit SPI protocol interface (flash serial memory).
- One single-SD memory card plus up to three devices with 8-bit SPI protocol interface.

Other combinations , e.g., two SD cards, one MMC card + one SD card, are not supported.

The application interface is responsible for managing transaction semantics. The MMC/SD host controller handles the MMC/SD protocol at the transmission level, packing data, adding cyclic redundancy checking (CRC), start/end bit, and checking for syntactical correctness. The SD mode wide-bus width is also supported.

The application interface can send every MMC/SD command and either poll for the status of the adapter or wait for an interrupt request, which is sent back in case of exceptions or to warn for the end of operations. The application interface can read card responses or flag registers. It can also individually mask interrupt sources. All these operations can be performed by reading and writing the control registers.

20 Inter-Integrated Circuit (I2C) Controller

The I^2C controller function supports the multimaster mode using the multimaster bus, to which devices capable of controlling the bus can be connected. Each I^2C device (including the OMAP5910) has a unique address and can operate as either transmitter or receiver. In addition to being a transmitter or receiver, a device connected to the I^2C bus can also be considered a master or slave when performing data transfers. A master device initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this master is considered a slave.

21 Timer

This document describes the 32-bit timers, the watchdog timers, and the 32kHz clock-based timer present in the OMAP5910 device.

32-bit timers:

The MPU has three 32-bit timers for the operating system that provide general-purpose housekeeping functions. These timers are configured either in autoreload or one-shot mode with on-the-fly read capability. The timers generate an interrupt to the MPU when equal to zero.

The timers are 32-bit counters that receive a dedicated clock from the clock generator module 1 (either CLKIN or DPLL1 output). This clock can then be prescaled, which divides it down further. Prescaling is controlled by the PTV field of the control-timer register (CNTL_TIMER).

Watchdog timers:

The MPU-watchdog timer is power-up enabled and defaults to a watchdog timer for the OMAP5910 device. A watchdog timer requires that the user program or that the OS periodically write to the count register before the counter underflows. If the counter underflows, the watchdog timer generates a reset to the MPU and the DSP. The watchdog timer detects user programs stuck in an infinite loop, loss of program control, or a runaway condition. When used as a general-purpose timer, the watchdog timer is a 16-bit timer configurable either in autoreload or one-shot mode with on-the-fly read capability. The timer generates an interrupt to the MPU when the count passes through zero.

32kHz timer:

The MPU subsystem operating system (OS) requires interrupts at regular time intervals for OS scheduling purposes. The OS time intervals can be from 1 ms to 30 ms. These time intervals can be generated using the three MPU 32-bit OS/general-purpose timers, which use CLKIN or DPLL1; however, they can not be used when the system clock is not operating. Therefore, a 32-kHz clock-based timer is needed to provide the required OS timing interval. The clock period of 32 kHz is 30.60 μs. (Note: 32 kHz refers to 32678, not 32000.)

22 Inter-Processor Communication

The MPU and DSP processors communicate via a mailbox-interrupt mechanism. This mechanism provides a very flexible software protocol between the MPU and DSP processors. The mailboxes are located in the shared memory space (byte address 0xFFFC:E000 for MPU; word address 0x0F800 for DSP).

23 Camera Interface

The camera interface receives 8-bit parallel data with vertical and horizontal synchronization signals from an external camera module. The interface is connected to the 32-bit MPU-public-peripheral bus. Data can be retrieved either by the MPU directly or through the System DMA. Optionally, the clock and reset signals can be provided from the OMAP5910 device.

24 Multichannel Serial Interface (MCSI)

Multichannel serial interfaces (MCSIs) have multichannel transmission capability and expand the parallel interface of a DSP to connect to external devices such as codecs and GSM system simulators.

The two public MCSIs provide full duplex transmission and master/slave clock control. All transmission parameters are configurable to cover the maximum number of operating conditions:

- Master/slave clock control (transmission clock and frame synchronization pulse)
- Programmable transmission clock frequency
- Single-channel or multichannel (x16) frame structure
- Programmable word length: 3 to 16 bits
- Full-duplex transmission
- Programmable frame configuration
  - Continuous or burst transmission
  - Normal or alternate framing
  - Normal or inverted frame polarity
  - Short or long frame pulse
  - Programmable oversize frame length
  - Programmable frame length
- Programmable interrupt occurrence time (TX and RX)
- Error detection with interrupt generation on wrong frame length
- DMA support for both TX and RX data transfers

25 Micro-Wire Interface

This serial synchronous interface can drive two serial external components. For the external devices, this interface is compatible with the MicroWire standard and is seen as the master.

A transmit DMA mode is available.

26 Real-Time Clock (RTC)

The real time clock (RTC) is an embedded module. Its basic features include:

- Time information (seconds/minutes/hours) directly in BCD code
- Calendar information (day/month/year/day of the week) directly in BCD code up to the year 2099
- Interrupt generation, at 1second, 1minute, 1hour, or 1day periods or at a precise, predetermined time of the day (alarm function)
- 30 second time correction
- Oscillator drift compensation

27 HDQ/1-Wire Interface

The HDQ and 1-Wire battery monitoring serial interface module in the OMAP5910 device implements the hardware protocol of the master function of the TI/Benchmarq HDQ and the Dallas Semiconductor 1-Wire protocols. The module utilizes a command structure that is programmed into transmit-command registers. The received data is in the received-data register. The firmware is responsible for doing the correct sequencing in the command registers. The module only implements the hardware-interface layer of the protocols. (See section 2 below for a description of the HDQ and 1-Wire Protocols.)

The HDQ and the 1-Wire modes are selectable in software, which must be done before any transmit and receive from the module is performed. The mode is assumed static during operation of the device.

28 PWL, PWT, and LED

The OMAP5910 Pseudonoise pulse-width light modulator (PWL), pulse-width tone (PWT), and light-emitting diode (LED) modules are described in this chapter.

This pulse-width light (PWL) module provides control of LCD backlighting and keypad voltage by employing a 4096-bit random sequence generator. This voltage-level control technique decreases the spectral power at the modulator's harmonic frequencies. The module uses a 32-kHz clock from the ultra−low power device (ULPD) module on the Clock and Reset Management module. For more information on the ULPD module see SPRU678.

The pulse-width tone (PWT) module generates a modulated frequency signal for driving an external buzzer. The frequency is programmable with 12 half-tone frequencies per octave. The volume level is also programmable. All frequencies are generated from the PWT_CLK, which is a 12 or 13 MHz clock derived in the Clock and Reset Management module.

The LED pulse generator (LPG) module controls indicator LEDs. The blinking period is programmable between 152 ms and 4s, and the LED can be switched on permanently. The OMAP5910 device has two LED modules. Each LED module drives a single output pin (LED1, LED2) on the OMAP5910 device which can be used to switch an LED driver.

29 Multichannel Buffered Serial Port (McBSP)

Multichannel buffered serial ports (McBSPs) are configurable, high-speed, full-duplex serial ports that allow direct interface to external communication devices. The OMAP5910 device has three McBSPs; McBSP1 and McBSP3 are on the DSP public peripheral bus, and McBSP2 is on the MPU public peripheral bus.

The key features of the McBSPs include:

- ❏ Full-duplex communication
- ❏ DMA support for both RX and TX transfers
- ❏ Double-buffered data registers, which allow a continuous data stream
- ❏ Independent framing and clocking for receives and transmits
- ❏ External shift clock generation or an internal programmable frequency shift clock
- ❏ Multichannel transmits and receives 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
- ❏ Data transfers with the LSB or MSB first
- ❏ Programmable polarity for both frame synchronization and data clocks
- ❏ Highly programmable internal clock and frame generation
- ❏ Supports bit rates up to 25M bits/second
- ❏ RX and TX interrupts as well as RX data overrun interrupt

For a detailed description of the functionality of all three McBSPs, see SPRU592, *TMS320VC5501/5502/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide*. The operation of the OMAP5910 McBSPs is consistent with the description in SPRU592, with the following exceptions and clarifications:

- ❏ Only the DXENA = 0 setting is supported.
- ❏ The transmit output (DX) pins do not go to high impedance when the transmitter is not actively sending data. In other words, the OMAP5910 device always actively drives the DX pins.
- ❏ The CLKS input is only available on McBSP1.
- ❏ The receiver can only operate in the slave mode on McBSP1 and McBSP3.

30 Setup, Components, and Features

This section introduces the setup, components, and features of the OMAP5910 processor and provides a high-level view of the device architecture.

Index