TMS320VC5501 Fixed-Point Digital Signal Processor

Data Manual

Literature Number: SPRS206
December 2002

PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.

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.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

REVISION HISTORY

REVISION	DATE	PRODUCT STATUS	HIGHLIGHTS
*	December 2002	Product Preview	Original

Contents

Section Page

Section Page

List of Figures

Figure *Page*

List of Tables

Table Page

1 TMS320VC5501 Features

- **High-Performance, Low-Power, Fixed-Point TMS320C55x™ Digital Signal Processor (DSP)**
  - **3.33-ns Instruction Cycle Time for 300-MHz Clock Rate**
  - **16K-Byte Instruction Cache (I-Cache)**
  - **One/Two Instructions Executed per Cycle**
  - **Dual Multipliers [Up to 600 Million Multiply-Accumulates Per Second (MMACS)]**
  - **Two Arithmetic/Logic Units (ALUs)**
  - **One Program Bus, Three Internal Data/Operand Read Buses, and Two Internal Data/Operand Write Buses**
- **16K x 16-Bit On-Chip RAM That is Composed of Four Blocks of 4K × 16-Bit Dual-Access RAM (DARAM) (32K Bytes)**
- **16K × 16-Bit One-Wait-State On-Chip ROM (32K Bytes)**
- **4M × 16-Bit Maximum Addressable External Memory Space**
- **16-Bit External Parallel Bus Memory Supporting External Memory Interface (EMIF) With General-Purpose Input/Output (GPIO) Capabilities and Glueless Interface to:**
  - **Asynchronous Static RAM (SRAM)**
  - **Asynchronous EPROM**
  - **Synchronous DRAM (SDRAM)**
  - **Synchronous Burst RAM (SBRAM)**
- **Emulation/Debug Trace Capability Saves Last 16 Program Counter (PC) Discontinuities and Last 32 PC Values**
- **Programmable Low-Power Control of Six Device Functional Domains**
- **On-Chip Peripherals**
  - **Six-Channel Direct Memory Access (DMA) Controller**
  - **Two Multichannel Buffered Serial Ports (McBSPs)**
  - **Programmable Analog Phase-Locked Loop (APLL) Clock Generator**
  - **General-Purpose I/O (GPIO) Pins and a Dedicated Output Pin (XF)**
  - **8-Bit Parallel Host-Port Interface (HPI)**
  - **Four Timers**
    - **Two 64-Bit General-Purpose Timers**
    - **64-Bit Programmable Watchdog Timer**
    - **64-Bit DSP/BIOS™ Counter**
  - **Inter-Integrated Circuit (I^2C) Interface**
  - **Universal Asynchronous Receiver/ Transmitter (UART)**
- **On-Chip Scan-Based Emulation Logic**
- **IEEE Std 1149.1† (JTAG) Boundary Scan Logic**
- **Packages:**
  - **176-Terminal LQFP (Low-Profile Quad Flatpack) (PGF Suffix)**
  - **176-Terminal MicroStar BGA™ (Ball Grid Array) (GGW Suffix)**
- **3.3-V I/O Supply Voltage**
- **1.26-V Core Supply Voltage**

PRODUCT PREVIEW

TMS320C55x, DSP/BIOS, and MicroStar BGA are trademarks of Texas Instruments.
Other trademarks are the property of their respective owners.
† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

2 Introduction

This section describes the main features of the TMS320VC5501 and gives a brief description of the device.

NOTE: This document is designed to be used in conjunction with the *TMS320C55x™ DSP Functional Overview* (literature number SPRU312) and the *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371).

2.1 Description

The TMS320VC5501 (5501) fixed-point digital signal processor (DSP) is based on the TMS320C55x™ DSP generation CPU processor core. The C55x™ DSP architecture achieves high performance and low power through increased parallelism and total focus on reduction in power dissipation. The CPU supports an internal bus structure that is composed of one program bus, three data read buses, two data write buses, and additional buses dedicated to peripheral and DMA activity. These buses provide the ability to perform up to three data reads and two data writes in a single cycle. In parallel, the DMA controller can perform data transfers independent of the CPU activity.

The C55x™ CPU provides two multiply-accumulate (MAC) units, each capable of 17-bit x 17-bit multiplication in a single cycle. A central 40-bit arithmetic/logic unit (ALU) is supported by an additional 16-bit ALU. Use of the ALUs is under instruction set control, providing the ability to optimize parallel activity and power consumption. These resources are managed in the Address Unit (AU) and Data Unit (DU) of the C55x CPU.

The C55x DSP generation supports a variable byte width instruction set for improved code density. The Instruction Unit (IU) performs 32-bit program fetches from internal or external memory and queues instructions for the Program Unit (PU). The Program Unit decodes the instructions, directs tasks to AU and DU resources, and manages the fully protected pipeline. Predictive branching capability avoids pipeline flushes on execution of conditional instructions.

The 5501 peripheral set includes an external memory interface (EMIF) that provides glueless access to asynchronous memories like EPROM and SRAM, as well as to high-speed, high-density memories such as synchronous DRAM and synchronous burst RAM. Additional peripherals include UART, watchdog timer, and an I-Cache. Two full-duplex multichannel buffered serial ports (McBSPs) provide glueless interface to a variety of industry-standard serial devices, and multichannel communication with up to 128 separately enabled channels. The host-port interface (HPI) is an 8-bit parallel interface used to provide host processor access to 16K words of internal memory on the 5501. The HPI operates in multiplexed mode to provide glueless interface to a wide variety of host processors. The DMA controller provides data movement for six independent channel contexts without CPU intervention. Two general-purpose timers, eight dedicated general-purpose I/O (GPIO) pins, and analog phase-locked loop (APLL) clock generation are also included.

The 5501 is supported by the industry's award-winning eXpressDSP™, Code Composer Studio™ Integrated Development Environment (IDE), DSP/BIOS™, Texas Instruments' algorithm standard, and the industry's largest third-party network. The Code Composer Studio™ IDE features code generation tools that include a C Compiler, Visual Linker, simulator, RTDX™, XDS510™ emulation device drivers, and evaluation modules. The 5501 is also supported by the C55x™ DSP Library, which features more than 50 foundational software kernels (FIR filters, IIR filters, FFTs, and various math functions) as well as chip and board support libraries.

2.2 Pin Assignments

2.2.1 Ball Grid Array (GGW)

Figure 2–1 illustrates the ball locations for the 176-pin ball grid array (BGA) package and Table 2–1 lists the signal names and terminal numbers.

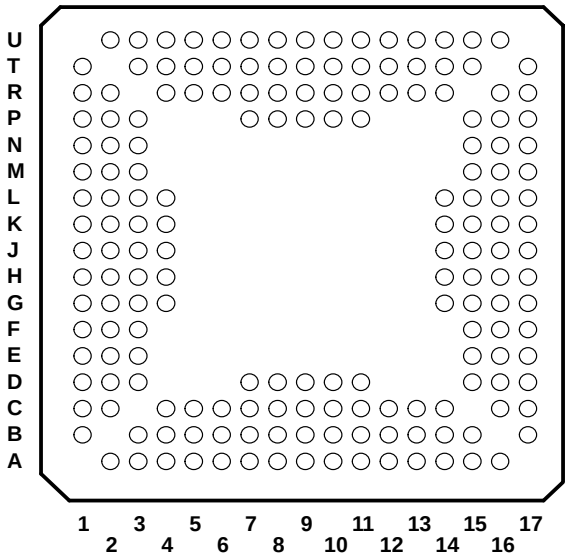


Figure 2–1. 176-Terminal GGW Ball Grid Array (Bottom View)

Table 2–1. 176-Terminal GGW Ball Grid Array Ball Assignments†‡

BALL NO.	SIGNAL NAME	BALL NO.	SIGNAL NAME	BALL NO.	SIGNAL NAME	BALL NO.	SIGNAL NAME
B1	GPIO6	U2	HCNTL1	T17	A19	A16	PGPIO4
C2	GPIO4	T3	HCNTL0	R16	A18	B15	D15
C1	GPIO2	U3	V_{SS}	R17	V_{SS}	A15	D14
D3	GPIO1	R4	HR/$\overline{W}$	P15	A17	C14	D13
D2	GPIO0	T4	$\overline{HDS2}$	P16	A16	B14	D12
D1	TIM1	U4	CV_{DD}	P17	DV_{DD}	A14	D11
E3	TIM0	R5	$\overline{HDS1}$	N15	A15	C13	D10
E2	$\overline{INT0}$	T5	HRDY	N16	A14	B13	D9
E1	CV_{DD}	U5	DV_{DD}	N17	V_{SS}	A13	DV_{DD}
F3	$\overline{INT1}$	R6	CLKOUT	M15	A13	C12	D8
F2	$\overline{INT2}$	T6	XF	M16	A12	B12	D7
F1	DV_{DD}	U6	V_{SS}	M17	CV_{DD}	A12	V_{SS}
G4	$\overline{INT3}$	P7	C15	L14	A11	D11	D6
G3	$\overline{NMI}$/WDTOUT	R7	C14	L15	A10	C11	D5
G2	$\overline{IACK}$	T7	$\overline{HINT}$	L16	A9	B11	D4
G1	V_{SS}	U7	PV_{DD}	L17	A8	A11	CV_{DD}
H1	CLKR0	U8	PSENSE	K17	DV_{DD}	A10	D3
H4	DR0	P8	X1	K14	A7	D10	D2
H3	FSR0	R8	X2/CLKIN	K15	A6	C10	D1
H2	CLKX0	T8	EMIFCLKS	K16	A5	B10	D0
J1	CV_{DD}	U9	V_{SS}	J17	V_{SS}	A9	V_{SS}
J4	DX0	P9	C13	J14	A4	D9	EMU1/$\overline{OFF}$
J3	FSX0	R9	C12	J15	A3	C9	EMU0
J2	CLKR1	T9	C11	J16	A2	B9	TDO
K1	DR1	U10	C10	H17	CV_{DD}	A8	V_{SS}
K2	FSR1	T10	C9	H16	PGPIO19	B8	TDI
K4	DX1	P10	C8	H14	PGPIO18	D8	$\overline{TRST}$
K3	CLKX1	R10	C7	H15	PGPIO17	C8	TCK
L1	V_{SS}	U11	V_{SS}	G17	V_{SS}	A7	TMS
L2	FSX1	T11	ECLKIN	G16	PGPIO16	B7	$\overline{RESET}$
L3	TEST	R11	ECLKOUT2	G15	PGPIO15	C7	HPIENA
L4	NC	P11	ECLKOUT1	G14	PGPIO14	D7	HD7
M1	CV_{DD}	U12	CV_{DD}	F17	CV_{DD}	A6	CV_{DD}
M2	RX	T12	C6	F16	PGPIO13	B6	HD6
M3	GPIO5	R12	C5	F15	PGPIO12	C6	HD5
N1	DV_{DD}	U13	DV_{DD}	E17	DV_{DD}	A5	DV_{DD}
N2	TX	T13	C4	E16	PGPIO11	B5	HD4
N3	GPIO3	R13	C3	E15	PGPIO10	C5	HD3
P1	V_{SS}	U14	V_{SS}	D17	PGPIO9	A4	CV_{DD}
P2	SCL	T14	C2	D16	PGPIO8	B4	HD2
P3	SDA	R14	C1	D15	PGPIO7	C4	HD1
R1	HC1	U15	C0	C17	V_{SS}	A3	V_{SS}
R2	HC0	T15	A21	C16	PGPIO6	B3	HD0
T1	$\overline{HCS}$	U16	A20	B17	PGPIO5	A2	GPIO7

† CV_{DD} is core V_{DD}, DV_{DD} is I/O V_{DD}, and PV_{DD} is PLL V_{DD}. PSENSE must be left unconnected.
‡ NC indicates “no connect”.

2.2.2 Low-Profile Quad Flatpack (PGF)

Figure 2–2 illustrates the pin locations for the 176-pin low-profile quad flatpack (LQFP) and Table 2–2 provides a numerical list (by pin number) of the pin assignments.

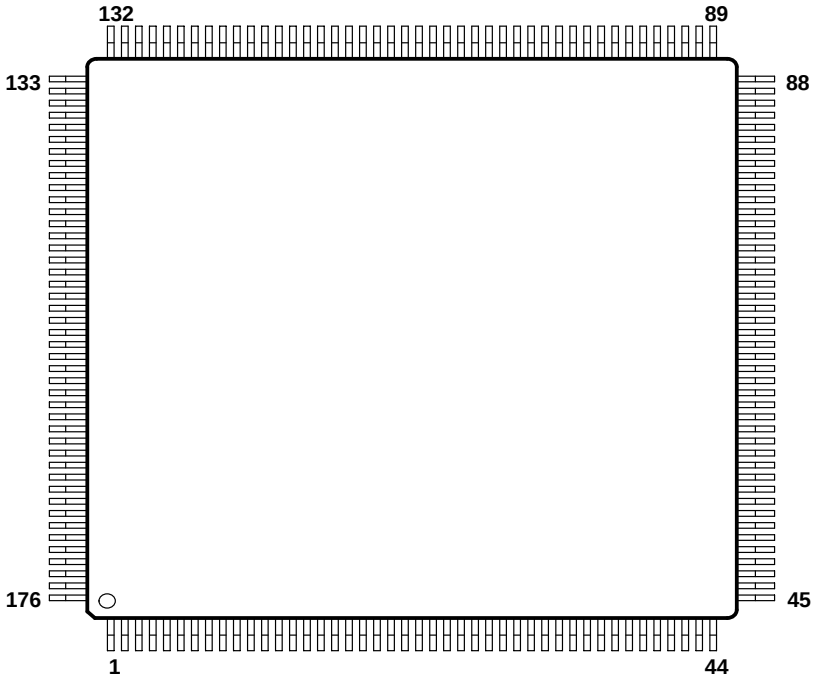


Figure 2–2. 176-Pin PGF Low-Profile Quad Flatpack (Top View)

Table 2–2. 176-Pin PGF Low-Profile Quad Flatpack Pin Assignments†‡

PIN NO.	SIGNAL NAME	PIN NO.	SIGNAL NAME	PIN NO.	SIGNAL NAME	PIN NO.	SIGNAL NAME
1	GPIO6	45	HCNTL1	89	A19	133	PGPIO4
2	GPIO4	46	HCNTL0	90	A18	134	D15
3	GPIO2	47	V_{SS}	91	V_{SS}	135	D14
4	GPIO1	48	HR/$\overline{W}$	92	A17	136	D13
5	GPIO0	49	$\overline{HDS2}$	93	A16	137	D12
6	TIM1	50	CV_{DD}	94	DV_{DD}	138	D11
7	TIM0	51	$\overline{HDS1}$	95	A15	139	D10
8	$\overline{INT0}$	52	HRDY	96	A14	140	D9
9	CV_{DD}	53	DV_{DD}	97	V_{SS}	141	DV_{DD}
10	$\overline{INT1}$	54	CLKOUT	98	A13	142	D8
11	$\overline{INT2}$	55	XF	99	A12	143	D7
12	DV_{DD}	56	V_{SS}	100	CV_{DD}	144	V_{SS}
13	$\overline{INT3}$	57	C15	101	A11	145	D6
14	$\overline{NMI}$/WDTOUT	58	C14	102	A10	146	D5
15	$\overline{IACK}$	59	$\overline{HINT}$	103	A9	147	D4
16	V_{SS}	60	PV_{DD}	104	A8	148	CV_{DD}
17	CLKR0	61	PSENSE	105	DV_{DD}	149	D3
18	DR0	62	X1	106	A7	150	D2
19	FSR0	63	X2/CLKIN	107	A6	151	D1
20	CLKX0	64	EMIFCLKS	108	A5	152	D0
21	CV_{DD}	65	V_{SS}	109	V_{SS}	153	V_{SS}
22	DX0	66	C13	110	A4	154	EMU1/$\overline{OFF}$
23	FSX0	67	C12	111	A3	155	EMU0
24	CLKR1	68	C11	112	A2	156	TDO
25	DR1	69	C10	113	CV_{DD}	157	V_{SS}
26	FSR1	70	C9	114	PGPIO19	158	TDI
27	DX1	71	C8	115	PGPIO18	159	$\overline{TRST}$
28	CLKX1	72	C7	116	PGPIO17	160	TCK
29	V_{SS}	73	V_{SS}	117	V_{SS}	161	TMS
30	FSX1	74	ECLKIN	118	PGPIO16	162	$\overline{RESET}$
31	TEST	75	ECLKOUT2	119	PGPIO15	163	HPIENA
32	NC	76	ECLKOUT1	120	PGPIO14	164	HD7
33	CV_{DD}	77	CV_{DD}	121	CV_{DD}	165	CV_{DD}
34	RX	78	C6	122	PGPIO13	166	HD6
35	GPIO5	79	C5	123	PGPIO12	167	HD5
36	DV_{DD}	80	DV_{DD}	124	DV_{DD}	168	DV_{DD}
37	TX	81	C4	125	PGPIO11	169	HD4
38	GPIO3	82	C3	126	PGPIO10	170	HD3
39	V_{SS}	83	V_{SS}	127	PGPIO9	171	CV_{DD}
40	SCL	84	C2	128	PGPIO8	172	HD2
41	SDA	85	C1	129	PGPIO7	173	HD1
42	HC1	86	C0	130	V_{SS}	174	V_{SS}
43	HC0	87	A21	131	PGPIO6	175	HD0
44	$\overline{HCS}$	88	A20	132	PGPIO5	176	GPIO7

† CV_{DD} is core V_{DD}, DV_{DD} is I/O V_{DD}, and PV_{DD} is PLL V_{DD}. PSENSE must be left unconnected.
‡ NC indicates “no connect”.

2.3 Signal Descriptions

Table 2–3 lists each signal, function, and operating mode(s) grouped by function. See Section 2.2, Pin Assignments, for exact pin locations based on package type.

Table 2–3. Signal Descriptions

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Parallel Bus				
A[21:18]		I/O/Z	C, D, F, G, H, M	These pins serve one of two functions: EMIF address bus EMIF.A[21:18] or PGPIO[3:0]. At reset, GPIO6 is sampled. If GPIO6 is low, the address bus A[21:18] is set to GPIO with all pins set as inputs. If GPIO6 is high, the EMIF mode is enabled and all address pins are set as outputs. The address bus includes a bus holder feature that eliminates passive components and the power dissipation associated with them. The bus holder keeps the address bus at the previous logic level when the bus went into a high-impedance state. These pins also include Schmitt-trigger inputs.
	PGPIO[3:0]	I/O/Z		**Parallel general-purpose I/O.** PGPIO[3:0] is selected when GPIO6 is low at reset.
	EMIF.A[21:18]	O/Z		**EMIF address bus.** EMIF.A[21:18] is selected when GPIO6 is high at reset, enabling EMIF mode and the EMIF address bus drives the Parallel Port Mux address bus.
A[17:2]		I/O/Z	C, D, F, M	These pins serve one of two functions: EMIF address bus EMIF.A[17:2] or reserved pins. At reset, GPIO6 is sampled. If GPIO6 is low, A[17:2] become reserved pins with no function. If GPIO6 is high, the EMIF mode is enabled and all address pins are set as outputs. The address bus includes a bus holder feature that eliminates passive components and the power dissipation associated with them. The bus holder keeps the address bus at the previous logic level when the bus went into a high-impedance state. These pins also include Schmitt trigger inputs.
	Reserved	I		**Reserved pins.** These pins are reserved when GPIO6 is low at reset.
	EMIF.A[17:2]	O/Z		**EMIF address bus.** EMIF.A[17:2] is selected when GPIO6 is high at reset, enabling EMIF mode and the EMIF address bus drives the Parallel Port Mux address bus.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:
A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to $\overline{\text{HOLD}}$ pin).
F – Pin is high impedance in OFF mode (due to EMU1/$\overline{\text{OFF}}$ pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset ($\overline{\text{RESET}}$ pin is low)

PRODUCT PREVIEW

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Parallel Bus (Continued)				
D[15:0]		I/O/Z	C, D, F, M	The 16 data pins, D[15:0], are used to transfer data between the core CPU and external data/program memory or I/O devices. At reset, GPIO6 is sampled. If GPIO6 is low, then D[15:0] become reserved pins and have no function. If GPIO6 is high, the EMIF mode is enabled and all data pins are set as data inputs. The data bus includes bus holders to reduce the static power dissipation caused by floating, unused pins. The bus holders also eliminate the need for external bias resistors on unused pins. When the data bus is not being driven by the CPU, the bus holders keep the pins at the logic level that were most recently driven. The data bus holders are disabled at reset, and can be enabled/disabled under software control. At reset, the bus is in a high-impedance state.
	Reserved	I/O/Z		**Reserved pins.** These pins are reserved when GPIO6 is low at reset.
	EMIF.D[15:0]	I/O/Z		**EMIF data bus.** EMIF.D[15:0] is selected when GPIO6 is high. This setting enables the EMIF mode and the EMIF data bus drives the Parallel Port Mux data bus.
C0		I/O/Z	C, D, F, G, H, M	EMIF control function [asynchronous memory read-enable ($\overline{ARE}$) multiplexed with synchronous memory address strobe ($\overline{SADS}$), SDRAM column-address strobe ($\overline{SDCAS}$), and synchronous read enable ($\overline{SRE}$) (selected when by RENEN in CE Secondary Control Register 1)] or PGPIO20. At reset, GPIO6 is sampled. If GPIO6 is low, C0 is set to PGPIO20 input. If GPIO6 is high, the full EMIF mode is enabled and C0 is set to EMIF control I/O.
	EMIF.$\overline{ARE}$/$\overline{SADS}$/ $\overline{SDCAS}$/$\overline{SRE}$	O/Z		**EMIF control pin.** Asynchronous memory read-enable (EMIF.$\overline{ARE}$) / synchronous memory address strobe (EMIF.$\overline{SADS}$) / SDRAM column-address strobe (EMIF.$\overline{SDCAS}$) / synchronous read enable (EMIF.$\overline{SRE}$) selected when GPIO6 is high at reset.
	PGPIO20	I/O/Z		**Parallel general-purpose I/O 20.** PGPIO20 is selected when GPIO6 is low at reset.
C1		I/O/Z	C, D, F, G, H, M	EMIF control function [asynchronous memory output-enable ($\overline{AOE}$) multiplexed with synchronous memory output-enable ($\overline{SOE}$) and SDRAM row address strobe ($\overline{SDRAS}$)] or PGPIO21. At reset, GPIO6 is sampled. If GPIO6 is low, C1 is set to PGPIO21 input. If GPIO6 is high, the full EMIF mode is enabled and C1 is set to EMIF control output in high-impedance mode.
	EMIF.$\overline{AOE}$/$\overline{SOE}$/ $\overline{SDRAS}$	O/Z		**EMIF control pin.** Asynchronous memory output-enable (EMIF.$\overline{AOE}$)/ synchronous memory output-enable (EMIF.$\overline{SOE}$)/SDRAM row address strobe (EMIF.$\overline{SDRAS}$) selected when GPIO6 is high at reset.
	PGPIO21	I/O/Z		**Parallel general-purpose I/O21.** PGPIO21 is selected when GPIO6 is low at reset.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:

A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to $\overline{HOLD}$ pin).
F – Pin is high impedance in OFF mode (due to EMU1/OFF pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset ($\overline{RESET}$ pin is low)

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Parallel Bus (Continued)				
C2		I/O/Z	C, D, F, G, H, M	EMIF control function [Asynchronous memory write-enable ($\overline{\text{AWE}}$) multiplexed with synchronous memory write-enable (SWE) and SDRAM write-enable ($\overline{\text{SDWE}}$)] or PGPIO22. At reset, GPIO6 is sampled. If GPIO6 is low, C2 is set to PGPIO22 input. If GPIO6 is high, the full EMIF mode is enabled and C2 is set to EMIF control output in high-impedance mode.
	EMIF.$\overline{\text{AWE}}$/ $\overline{\text{SWE}}$/$\overline{\text{SDWE}}$	O/Z		**EMIF control pin.** Asynchronous memory write-enable (EMIF.$\overline{\text{AWE}}$)/ synchronous memory write-enable (EMIF.$\overline{\text{SWE}}$)/SDRAM write-enable (EMIF.$\overline{\text{SDWE}}$) selected when GPIO6 is high at reset.
	PGPIO22	I/O/Z		**Parallel general-purpose I/O22.** PGPIO22 is selected when GPIO6 is low at reset.
C3		I/O/Z	D, F, G, H, J, M	EMIF data ready input or PGPIO23. At reset, GPIO6 is sampled. If GPIO6 is low, C3 is set to PGPIO23 input. If GPIO6 is high, the full EMIF mode is enabled and C3 is set to EMIF.ARDY input.
	EMIF.ARDY	I		**EMIF data ready.** Input used to insert wait states for slow memories. EMIF.ARDY is selected when GPIO6 is high at reset.
	PGPIO23	I/O/Z		**Parallel general-purpose I/O23.** PGPIO23 is selected when GPIO6 is low at reset.
C4		I/O/Z	C, D, F, G, H, M	EMIF chip-select for memory space CE0 or PGPIO24. At reset, GPIO6 is sampled. If GPIO6 is low, C4 is set to PGPIO24 input. If GPIO6 is high, the full EMIF mode is enabled and C4 is set to EMIF.$\overline{\text{CE0}}$ input.
	EMIF.$\overline{\text{CE0}}$	O/Z		**EMIF chip-select for memory space CE0.** EMIF.$\overline{\text{CE0}}$ is selected when GPIO6 is high at reset.
	PGPIO24	I/O/Z		**Parallel general-purpose I/O24.** PGPIO24 is selected when GPIO6 is low at reset.
C5		I/O/Z	C, D, F, G, H, M	EMIF chip-select for memory space CE1 or PGPIO25. At reset, GPIO6 is sampled. If GPIO6 is low, C5 is set to PGPIO25 input. If GPIO6 is high, the full EMIF mode is enabled and C5 is set to EMIF.$\overline{\text{CE1}}$ input.
	EMIF.$\overline{\text{CE1}}$	O/Z		**EMIF chip-select for memory space CE1.** EMIF.$\overline{\text{CE1}}$ is selected when GPIO6 is high at reset.
	PGPIO25	I/O/Z		**Parallel general-purpose I/O 25.** PGPIO25 is selected when GPIO6 is low at reset.
C6		I/O/Z	C, D, F, G, H, M	EMIF chip-select for memory space CE2 or PGPIO26. At reset, GPIO6 is sampled. If GPIO6 is low, C6 is set to PGPIO26 input. If GPIO6 is high, the full EMIF mode is enabled and C6 is set to EMIF.$\overline{\text{CE2}}$ input.
	EMIF.$\overline{\text{CE2}}$	O/Z		**EMIF chip-select for memory space CE2.** EMIF.$\overline{\text{CE2}}$ is selected when GPIO6 is high at reset.
	PGPIO26	I/O/Z		**Parallel general-purpose I/O 26.** PGPIO26 is selected when GPIO6 is low at reset.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:

A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to $\overline{\text{HOLD}}$ pin).
F – Pin is high impedance in OFF mode (due to EMU1/$\overline{\text{OFF}}$ pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset ($\overline{\text{RESET}}$ pin is low)

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Parallel Bus (Continued)				
C7		I/O/Z	C, D, F, G, H, M	EMIF chip-select for memory space CE3 or PGPIO27. At reset, GPIO6 is sampled. If GPIO6 is low, C7 is set to PGPIO27 input. If GPIO6 is high, the full EMIF mode is enabled and C7 is set to EMIF.$\overline{CE3}$ input.
	EMIF.$\overline{CE3}$	O/Z		**EMIF chip-select for memory space CE3.** EMIF.$\overline{CE3}$ is selected when If GPIO6 is high at reset.
	PGPIO27	I/O/Z		**Parallel general-purpose I/O 27.** PGPIO27 is selected when GPIO6 is low at reset.
C8		I/O/Z	C, D, F, G, H, M	EMIF byte-enable 0 or PGPIO28. At reset, GPIO6 is sampled. If GPIO6 is low, C8 is set to PGPIO28 input. If GPIO6 is high, the full EMIF mode is enabled and C8 is set to EMIF.$\overline{BE0}$ output.
	EMIF.$\overline{BE0}$	O/Z		**EMIF byte-enable 0 control.** EMIF.$\overline{BE0}$ is selected when GPIO6 is high at reset.
	PGPIO28	I/O/Z		**Parallel general-purpose I/O 28.** PGPIO28 is selected when GPIO6 is low at reset.
C9		I/O/Z	C, D, F, G, H, M	EMIF byte-enable 1 or PGPIO29. At reset, GPIO6 is sampled. If GPIO6 is low, C9 is set to PGPIO29 input. If GPIO6 is high, the full EMIF mode is enabled and C9 is set to EMIF.$\overline{BE1}$ output.
	EMIF.$\overline{BE1}$	O/Z		**EMIF byte-enable 1 control.** EMIF.$\overline{BE1}$ is selected when GPIO6 is high at reset.
	PGPIO29	I/O/Z		**Parallel general-purpose I/O 29.** PGPIO29 is selected when GPIO6 is low at reset.
C10		I/O/Z	C, D, F, G, H, M	EMIF byte-enable 2 or PGPIO30. At reset, GPIO6 is sampled. If GPIO6 is low, C10 is set to PGPIO30 input. If GPIO6 is high, the full EMIF mode is enabled and C10 is set to EMIF.$\overline{BE2}$ output.
	NC	O/Z		**No connect.** This pin has no functionality; it should be left unconnected.
	PGPIO30	I/O/Z		**Parallel general-purpose I/O 30.** PGPIO30 is selected when GPIO6 is low at reset.
C11		I/O/Z	C, D, F, G, H, M	EMIF byte-enable 3 or PGPIO31. At reset, GPIO6 is sampled. If GPIO6 is low, C11 is set to PGPIO31 input. If GPIO6 is high, the full EMIF mode is enabled and C11 is set to EMIF.$\overline{BE3}$ output.
	NC	O/Z		**No connect.** This pin has no functionality; it should be left unconnected.
	PGPIO31	I/O/Z		**Parallel general-purpose I/O 31.** PGPIO31 is selected when GPIO6 is low at reset.
C12		I/O/Z	C, D, F, G, H, M	EMIF SDRAM clock-enable or PGPIO32. At reset, GPIO6 is sampled. If GPIO6 is low, C12 is set to PGPIO32 input. If GPIO6 is high, the full EMIF mode is enabled and C12 is set to EMIF.SDCKE output.
	EMIF.SDCKE	O/Z		**EMIF SDRAM clock-enable.** EMIF.SDCKE is selected when GPIO6 is high at reset.
	PGPIO32	I/O/Z		**Parallel general-purpose I/O32.** PGPIO32 is selected when GPIO6 is low at reset.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:

A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to $\overline{HOLD}$ pin).
F – Pin is high impedance in OFF mode (due to EMU1/$\overline{OFF}$ pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset ($\overline{RESET}$ pin is low)

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Parallel Bus (Continued)				
C13		I/O/Z	C, D, F, G, H, M	EMIF synchronous memory output-enable for CE3 or PGPIO33. At reset, GPIO6 is sampled. If GPIO6 is low, C13 is set to PGPIO33 input. If GPIO6 is high, the full EMIF mode is enabled and C13 is set to EMIF.$\overline{\text{SOE3}}$ output.
	EMIF.$\overline{\text{SOE3}}$	O/Z		**EMIF synchronous memory output-enable for CE3** (intended for glueless FIFO interface). EMIF.$\overline{\text{SOE3}}$ is selected when GPIO6 is high at reset.
	PGPIO33	I/O/Z		**Parallel general-purpose I/O 33.** PGPIO33 is selected when GPIO6 is low at reset.
C14		I/O/Z	F, G, H, J, M	EMIF memory interface Hold request from a host or PGPIO34. At reset, GPIO6 is sampled. If GPIO6 is low, C14 is set to PGPIO34 input. If GPIO6 is high, the full EMIF mode is enabled and C14 is set to EMIF.$\overline{\text{HOLD}}$ input.
	EMIF.$\overline{\text{HOLD}}$	O/Z		**Memory interface Hold request.** EMIF.$\overline{\text{HOLD}}$ is selected when GPIO6 is high at reset.
	PGPIO34	I/O/Z		**Parallel general-purpose I/O 34.** PGPIO34 is selected when GPIO6 is low at reset.
C15		I/O/Z	C, D, F, G, H, M	EMIF memory interface Hold Acknowledge from the C55x to a Host or PGPIO35. At reset, GPIO6 is sampled. If GPIO6 is low, C15 is set to PGPIO35 input. If GPIO6 is high, the full EMIF mode is enabled and C15 is set to EMIF.$\overline{\text{HOLDA}}$ output.
	EMIF.$\overline{\text{HOLDA}}$	O/Z		**Memory interface Hold Acknowledge.** EMIF.$\overline{\text{HOLDA}}$ is selected when GPIO6 is high at reset.
	PGPIO35	I/O/Z		**Parallel general-purpose I/O 35.** PGPIO35 is selected when GPIO6 is low at reset.
ECLKIN		I	C, L	**External EMIF input clock.** When EMIFCLKS = 1 during reset, this pin is configured as the input clock to the EMIF.
ECLKOUT1		O/Z	F, M	**EMIF output clock.** Drives EMIF input clock by default. Can be held low or set to high-impedance through the EMIF Global Control Register.
ECLKOUT2		O/Z	F	**EMIF output clock.** Set to high-impedance by default. Can be enabled to drive EMIF input clock divided by a factor of 1, 2, or 4 through the EMIF Global Control Register.
EMIFCLKS		I	C, L	**EMIF input clock source select.** If EMIFCLKS = 0 during reset, the internal clock is used by EMIF. If EMIFCLKS = 1 during reset, ECLKIN is used by EMIF.
HPI Data Pins				
HD[7:0]		I/O/Z	C, D, F, G, H, M	HPI data pins HPI.HD[7:0] or PGPIO[43:36]. At reset, GPIO6 is sampled. If GPIO6 is low, HD[7:0] are set to PGPIO[43:36]. If GPIO6 is high, HD[7:0] are set to HPI.HD[7:0].
	HPI.HD[7:0]	O/Z		**Data pins for HPI.** HPI.HD[7:0] are selected when GPIO6 is high at reset.
	PGPIO[43:36]	I/O/Z		**Parallel general-purpose I/O [43:36].** PGPIO[43:36] are selected when GPIO6 is low at reset.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:
- A – Internal pullup (always enabled)
- B – Internal pulldown (always enabled)
- C – Hysteresis input
- D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
- E – Pin is high impedance in HOLD mode (due to $\overline{\text{HOLD}}$ pin).
- F – Pin is high impedance in OFF mode (due to EMU1/$\overline{\text{OFF}}$ pin).
- G – Pin can be configured as a general-purpose input.
- H – PIn can be configured as a general-purpose output.
- J – Internal pullup enabled through the External Bus Control Register (XBCR)
- K – Internal pulldown enabled through the External Bus Control Register (XBCR)
- L – Fail-safe pin
- M – Pin is in high-impedance during reset ($\overline{\text{RESET}}$ pin is low)

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
HPI Control Pins				
HC0		I/O/Z	C, F, G, H, J, M	Host address strobe or PGPIO44. At reset, GPIO6 is sampled. If GPIO6 is low, HC0 is set to PGPIO44 input. If GPIO6 is high, HC0 is set to HPI.HAS input.
	HPI.HAS	I		**Host address strobe.** HPI.HAS is selected when GPIO6 is high at reset.
	PGPIO44	I/O/Z		**Parallel general-purpose I/O 44.** PGPIO44 is selected when GPIO6 is low at reset.
HC1		I/O/Z	F, G, H, K, M	HPI host byte identification or PGPIO45. At reset, GPIO6 is sampled. If GPIO6 is low, HC1 is set to PGPIO45 input. If GPIO6 is high, HC1 is set to HPI.HBIL input.
	HPI.HBIL	I		**Host byte identification.** HPI.HBIL is selected when GPIO6 is high at reset.
	PGPIO45	I/O/Z		**Parallel general-purpose I/O 45.** PGPIO45 is selected when GPIO6 is low at reset.
HCNTL0		I/O/Z	F, J, M	**HPI host control 0**
HCNTL1		I/O/Z	F, J, M	**HPI host control 1**
HCS		I/O/Z	C, F, J, M	**HPI host chip-select**
HR/W		I/O/Z	F, J, M	**HPI host read- or write-select**
HDS1		I	C, J	**Host data strobe 1**
HDS2		I	C, J	**Host data strobe 2**
HRDY		O/Z	F, J, M	**Host ready**
HINT		O/Z	F, J, M	**Host interrupt**
HPIENA		I	C, K	**HPI enable.** HPIENA must be pulled high and kept high in systems that use the HPI. If the HPI is not needed, HPIENA can be pulled low. An internal pulldown resistor is active on this pin.
Initialization, Interrupt, and Reset Pins				
INT[3:0]		I	C, L	**External user interrupt**. INT0–INT3 are maskable interrupts. They are prioritized by the Interrupt Enable Register (IER) and the interrupt mode bit. INT[3:0] can be polled and reset by way of the interrupt flag register.
NMI/WDTOUT		I/O/Z	C, F, J, M	**Non-maskable interrupt input and/or Watchdog Timer output.** The function of this pin is controlled by the Timer Signal Selection Register.
IACK		O/Z	F, M	**Interrupt acknowledge**
RESET		I	C, L	**Reset.** RESET causes the digital signal processor (DSP) to terminate current program execution. When RESET is brought to a high level, program execution begins by fetching the reset interrupt service routine vector at the reset vector address FFFF00h (IVPD:FFFFh). RESET affects various registers and status bits.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:
A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to HOLD pin).
F – Pin is high impedance in OFF mode (due to EMU1/OFF pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset (RESET pin is low)

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Bit I/O Signals				
GPIO[7:0]		I/O/Z	F, G, H, M	**General-purpose I/O.** Eight input/output lines that can be individually configured as inputs or outputs, and can also be individually set or reset when configured as outputs. At reset, the eight pins (GPIO[7:0]) are configured as inputs. These pins are latched with the rising edge of RESET and their state is used to determine the mode of the 5501 (EMIF/HPI mode or PGPIO mode) and boot mode of the bootloader. GPIO[0–2] are used for boot-mode selection of the bootloader. The GPIO[2:0] pins are sampled at reset and are stored in the BOOT_MOD (0x000F) register. GPIO4 is used as a function of the clock mode pin. At reset, if GPIO4 is high, the PLL is bypassed with clock running at input clock rate and the oscillator disabled. If GPIO4 is low, the PLL is bypassed with clock rate running at input clock rate and the oscillator enabled. GPIO4 is also used as an output for handshaking purposes on some of the boot modes. Although this pin is not involved in boot mode selection, users should be aware that this pin will become active as an output during the bootload process and should design accordingly. After the bootload is complete, the loaded application may change the function of the GPIO4 pin. GPIO6 is used to set the modes for the 5501. At reset, if GPIO6 is low, the 5501 is configured in PGPIO mode (the EMIF and HPI are disabled). If GPIO6 is high, the 5501 is configured in EMIF/HPI mode (PGPIO is not available in this mode).
PGPIO[19:4]		I/O/Z	C, D, F, G, H, M	PGPIO[19:4] or reserved. At reset, if GPIO6 is low at reset, these pins are used as PGPIO[19:4]. If GPIO6 is high, these pins are reserved.
	Reserved			**Reserved pins.** These pins are reserved when GPIO6 is low at reset.
	PGPIO[19:4]			**Parallel general-purpose I/O.** PGPIO[19:4] is selected when GPIO6 is low at reset.
XF		O/Z	F	**External output** (latched software-programmable signal)**.** XF is set high by the BSET XF instruction, set low by BCLR XF instruction, or by loading ST1. XF is used for signaling other processors in multiprocessor configurations or used as a general-purpose output pin.
Oscillator/Clock Signals				
CLKOUT		O/Z	F	**Master clock output signal.** CLKOUT can be set to reflect the clock of fast peripherals (DMA, HPI, ...), slow peripherals (I^2C, UART, ...0), or EMIF clock.
X2/CLKIN		I		**Clock/oscillator input.** If the internal oscillator is not being used, X2/CLKIN functions as the clock input.
X1		O		**Output pin from the internal oscillator for the crystal.** If the internal oscillator is not used, X1 should be left unconnected.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:
- A – Internal pullup (always enabled)
- B – Internal pulldown (always enabled)
- C – Hysteresis input
- D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
- E – Pin is high impedance in HOLD mode (due to HOLD pin).
- F – Pin is high impedance in OFF mode (due to EMU1/OFF pin).
- G – Pin can be configured as a general-purpose input.
- H – PIn can be configured as a general-purpose output.
- J – Internal pullup enabled through the External Bus Control Register (XBCR)
- K – Internal pulldown enabled through the External Bus Control Register (XBCR)
- L – Fail-safe pin
- M – Pin is in high-impedance during reset (RESET pin is low)

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Timer Signals				
TIM0		I/O/Z	F, G, H, M	**Input/Output terminal from Timer 0.** When configured as an output, TIM0 signals a pulse or a change of state when the on-chip timer count matches its period. When configured as an input, TIM0 provides the clock source for the internal timer module. At reset, this pin is configured as input. This pin can also be used as general-purpose I/O.
TIM1		I/O/Z	F, G, H, M	**Input/Output terminal from Timer 1.** When configured as an output, TIM1 signals a pulse or a change of state when the on-chip timer count matches its period. When configured as an input, TIM1 provides the clock source for the internal timer module. At reset, this pin is configured as input. This pin can also be used as general-purpose I/O.
Multichannel Buffered Serial Port Signals				
CLKR0		I/O/Z	C, F, G, H, M	**Receive clock input of McBSP0**
DR0		I	L, G	**Serial data receive input of McBSP0**
FSR0		I/O/Z	F, G, H, M	**Frame synchronization pulse for receive input of McBSP0**
CLKX0		I/O/Z	C, F, G, H, M	**Transmit clock of McBSP0**
DX0		O/Z	F, H, M	**Serial data transmit output of McBSP0**
FSX0		I/O/Z	F, G, H, M	**Frame synchronization pulse for transmit output of McBSP0**
CLKR1		I/O/Z	C, G, H, M	**Receive clock input of McBSP1**
DR1		I	L, G	**Serial data receive input of McBSP1**
FSR1		I/O/Z	F, G, H, M	**Frame synchronization pulse for receive input of McBSP1**
DX1		O/Z	F, H, M	**Serial data transmit output of McBSP1**
CLKX1		I/O/Z	C, F, G, H, M	**Transmit clock of McBSP1**
FSX1		I/O/Z	F, G, H, M	**Frame synchronization pulse for transmit output of McBSP1**

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:

A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to $\overline{\text{HOLD}}$ pin).
F – Pin is high impedance in OFF mode (due to EMU1/$\overline{\text{OFF}}$ pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset ($\overline{\text{RESET}}$ pin is low)

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
UART				
TX		O		**UART transmit data output**
RX		I		**UART receive data input**
I2C Pins				
SCL		I/O/Z	C, F, M	**I2C clock bidirectional port.** (Open collector I/O)
SDA		I/O/Z	C, F, M	**I2C data bidirectional port.** (Open collector I/O)
Supply Pins				
V_{SS}		S		**Digital Ground.** Dedicated ground for the device.
CV_{DD}		S		**Digital Power, + V_{DD}.** Dedicated power supply for the core CPU.
PV_{DD}		S		**Digital Power, + V_{DD}.** Dedicated power supply for the PLL module.
PSENSE				**For test purposes only.** Must be left unconnected.
DV_{DD}		S		**Digital Power, + V_{DD}.** Dedicated power supply for the I/O pins.
Test Pins				
TCK		I	C, J	**IEEE standard 1149.1 test clock.** TCK is normally a free-running clock signal with a 50% duty cycle. The changes on test access port (TAP) of input signals TMS and TDI are clocked into the TAP controller, instruction register, or selected test data register on the rising edge of TCK. Changes at the TAP output signal (TDO) occur on the falling edge of TCK.
TDI		I	J	**IEEE standard 1149.1 test data input.** Pin with internal pullup device. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK.
TDO		O/Z		**IEEE standard 1149.1 test data output.** The contents of the selected register (instruction or data) are shifted out of TDO on the falling edge of TCK. TDO is in the high-impedance state except when the scanning of data is in progress.
TMS		I	J	**IEEE standard 1149.1 test mode select.** Pin with internal pullup device. This serial control input is clocked into the TAP controller on the rising edge of TCK.
$\overline{TRST}$		I	C, L, K	**IEEE standard 1149.1 test reset.** $\overline{TRST}$, when high, gives the IEEE standard 1149.1 scan system control of the operations of the device. If $\overline{TRST}$ is not connected or driven low, the device operates in its functional mode, and the IEEE standard 1149.1 signals are ignored. Pin with internal pulldown device.

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:

A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to $\overline{HOLD}$ pin).
F – Pin is high impedance in OFF mode (due to EMU1/$\overline{OFF}$ pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset ($\overline{RESET}$ pin is low)

PRODUCT PREVIEW

Table 2–3. Signal Descriptions (Continued)

Pin Name	Multiplexed Signal Name	Pin Type†	Other‡	Function
Test Pins (Continued)				
EMU0		I/O/Z	J	**Emulator 0 pin.** When $\overline{\text{TRST}}$ is driven low, EMU0 must be high for activation of the $\overline{\text{OFF}}$ condition. When $\overline{\text{TRST}}$ is driven high, EMU0 is used as an interrupt to or from the emulator system and is defined as I/O by way of the IEEE standard 1149.1 scan system.
EMU1/$\overline{\text{OFF}}$		I/O/Z	J	**Emulator 1 pin/disable all outputs.** When $\overline{\text{TRST}}$ is driven high, EMU1/$\overline{\text{OFF}}$ is used as an interrupt to or from the emulator system and is defined as I/O by way of IEEE standard 1149.1 scan system. When $\overline{\text{TRST}}$ is driven low, EMU1/$\overline{\text{OFF}}$ is configured as $\overline{\text{OFF}}$. The EMU1/$\overline{\text{OFF}}$ signal, when active (low), puts all output drivers into the high-impedance state. Note that $\overline{\text{OFF}}$ is used exclusively for testing and emulation purposes (not for multiprocessing applications). Therefore, for the $\overline{\text{OFF}}$ condition, the following apply: $\overline{\text{TRST}}$ = low, EMU0 = high, EMU1/$\overline{\text{OFF}}$ = low

† I = Input, O = Output, S = Supply, Z = High impedance

‡ Other Pin Characteristics:

A – Internal pullup (always enabled)
B – Internal pulldown (always enabled)
C – Hysteresis input
D – Pin has bus holder enabled through the External Bus Control Register (XBCR).
E – Pin is high impedance in HOLD mode (due to $\overline{\text{HOLD}}$ pin).
F – Pin is high impedance in OFF mode (due to EMU1/$\overline{\text{OFF}}$ pin).
G – Pin can be configured as a general-purpose input.
H – PIn can be configured as a general-purpose output.
J – Internal pullup enabled through the External Bus Control Register (XBCR)
K – Internal pulldown enabled through the External Bus Control Register (XBCR)
L – Fail-safe pin
M – Pin is in high-impedance during reset ($\overline{\text{RESET}}$ pin is low)

3 Functional Overview

The following functional overview is based on the block diagram in Figure 3–1.

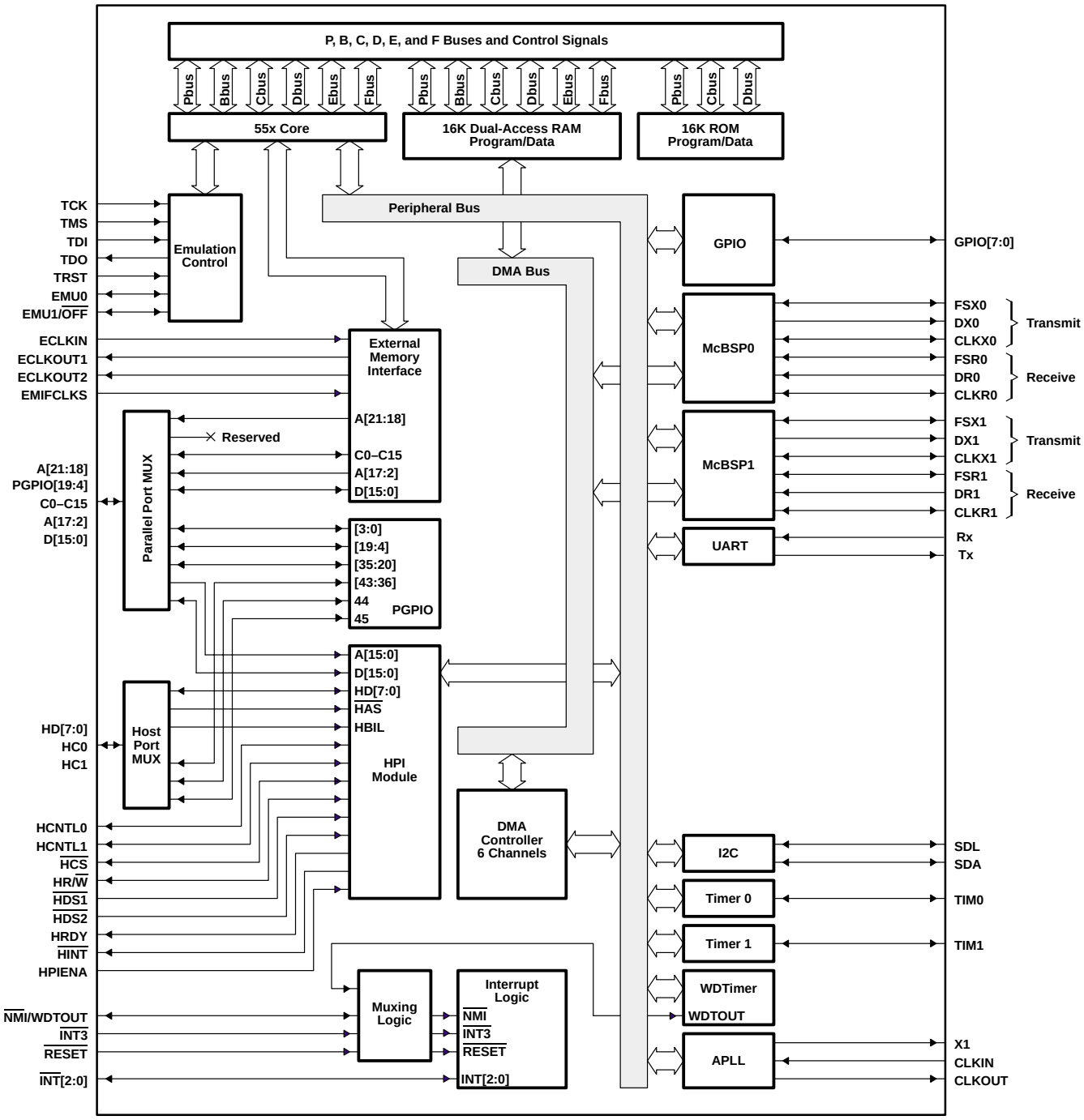


Figure 3–1. Block Diagram of the TMS320VC5501

3.1 Memory

The 5501 supports a unified memory map (program and data accesses are made to the same physical space). The total on-chip memory is 32K words (16K 16-bit words of RAM and 16K 16-bit words of ROM).

3.1.1 On-Chip ROM

TMS320VC5501 incorporates 16K x16-bit of on-chip, one-wait-state maskable ROM that can be mapped into program memory space. The ROM contains a bootloader program to facilitate system initialization. This bootloader can be used to automatically transfer user code from an external source to anywhere in the program memory at power up. The ROM can be masked.

The on-chip ROM is located at the byte address range FF8000h–FFFFFFh when bit MP/MC = 0. It can be accessed by the program bus and the three read data buses P, C, and D. The on-chip ROM is a one-cycle-per-word memory access, except for the first word access, which requires two cycles.

When MP/MC = 1 at reset, the on-chip ROM is disabled and not present in the memory map, and byte address range FF8000h–FFFFFFh is directed to external memory space. The status of the MP/MC bit (located in the ST3_55 status register) is determined by the logic level on the BOOTM[2:0] pins when sampled at reset. If BOOTM[2:0] is set to 00h, the MP/MC bit is set to 1 and the on-chip ROM is disabled. Otherwise, the MP/MC bit is cleared to 0 and the on-chip ROM is enabled. These pins are not sampled again until the next hardware reset. The software reset instruction does not affect the MP/MC bit. Software can also be used to set or clear the MP/MC bit.

The Bootloader mode selection is determined by the logic level on the GPIO[2:0] pins when sampled at reset. A branch instruction to the start of the Bootloader program is contained at FF8000h. TMS320VC5501 provides different ways to download the code to accommodate various system requirements:

- Parallel boot from 16-bit asynchronous memory connected to the EMIF at 200000h
- Serial boot from McBSP0 (McBSP0 in SPI mode or standard mode)
- Host-port interface boot load in multiplexed mode
- I^2C boot load
- UART boot load

The standard on-chip ROM layout is shown in Table 3–1.

Table 3–1. On-Chip ROM Layout

STARTING BYTE ADDRESS	CONTENTS
FF_8000h	Bootloader program
FF_ECB0h	Boot Mode Branch Table
FF_ED00h	Sine Table
FF_EF00h	Factory Test Code
FF_FF00h	Interrupt Vector Table
FF_FFFCh	ID Code

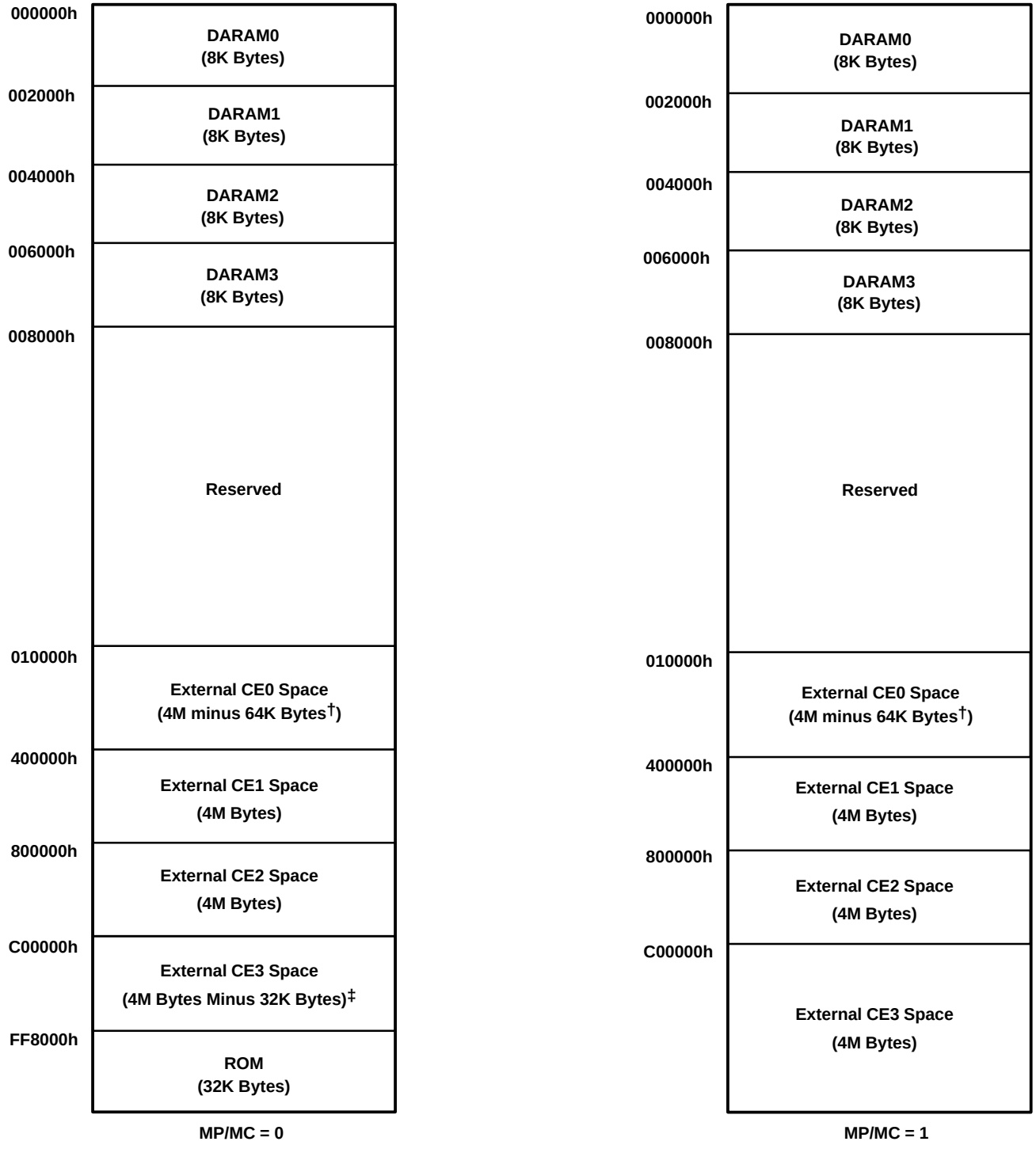
3.1.2 On-Chip Dual-Access RAM (DARAM)

TMS320VC5501 features 16K x 16-bit (32K bytes) of on-chip dual-access RAM. This memory enhances system performance, since the C55x CPU can access a DARAM block twice per machine cycle. The DARAM is composed of 4 blocks of 4K x 16-bit each (see Table 3–2). Each block in the DARAM can support two reads in one cycle, a read and a write in one cycle, or two writes in one cycle. The dual-access RAM is located in the (byte) address range 000000h–00FFFFh, it can be accessed by the program, data and DMA buses. The HPI has NO access to the DARAM block when device is in reset.

Table 3–2. DARAM Blocks

BYTE ADDRESS RANGE	MEMORY BLOCK
000000h – 001FFFh	DARAM 0
002000h – 003FFFh	DARAM 1
004000h – 005FFFh	DARAM 2
006000h – 007FFFh	DARAM 3

3.1.3 Memory Map



† The lower 64K bytes in CE0 Space include 32K bytes of DARAM space and 32K bytes of reserved space.
‡ The 32K bytes are for on-chip ROM block.

Figure 3–2. TMS320VC5501 Memory Map

3.1.4 Boot Configuration

The on-chip bootloader provides a way to transfer application code and tables from an external source to the on-chip RAM at power up. The 5501 provides several options to download the code to accommodate varying system requirements. These options include:

- Host-port interface (HPI) boot in multiplexed mode
- External memory boot (via EMIF) from 16-bit asynchronous memory
- Serial port boot (from McBSP0) with 16-bit element length
- SPI EPROM boot (from McBSP0) supporting EPROMs with 24-bit addresses
- I^2C EPROM boot (from I^2C) supporting EPROMs larger than 512K bits
- UART boot
- Direct execution (no boot) from 16-bit external asynchronous memory

The external pins BOOTM2, BOOTM1, and BOOTM0 select the boot configuration. The values of BOOTM[2:0] are latched with the rising edge of the $\overline{\text{RESET}}$ input. BOOTM2 is shared with GPIO2, BOOTM1 is shared with GPIO1, and BOOTM0 is shared with GPIO0.

The boot configurations available are summarized in Table 3–3.

Table 3–3. Boot Configuration Selection Via the BOOTM[2:0] Pins

BOOTM[2:0]	BOOT PROCESS
000	Direct execution from 16-bit external asynchronous memory
001	SPI EPROM boot
010	Serial port boot (from McBSP0)
011	External memory boot (via EMIF) from 16-bit asynchronous memory
100	Reserved
101	HPI boot
110	I^2C EPROM boot
111	UART boot

3.2 Peripherals

The 5501 includes the following on-chip peripherals:

- An external memory interface (EMIF)† supporting a 16-bit interface to asynchronous memory, SDRAM, and SBSRAM
- An 8-bit host-port interface (HPI)
- A six-channel direct memory access (DMA) controller
- Two multichannel buffered serial ports (McBSPs)
- A programmable analog phase-locked loop (APLL) clock generator
- General-purpose I/O (GPIO) pins and a dedicated output pin (XF)

† The 5501 can be configured as follows:
- EMIF/HPI mode: 16-bit external memory interface with 8-bit (multiplexed) host-port interface
- PGPIO mode: PGPIO support with no external memory interface and no host-post interface

- Four timers
  - Two 64-bit general-purpose timers
  - A programmable watchdog timer
  - A DSP/BIOS timer
- An Inter-integrated Circuit (I^2C) multi-master and slave interface
- A Universal Asynchronous Receiver/Transmitter (UART)

For detailed information on the C55x DSP peripherals, see the following documents:

- *TMS320C55x™ DSP Functional Overview* (literature number SPRU312)
- *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317)

3.3 Configurable External Ports and Signals

A number of pins on the 5501 have two functions, a feature that allows system designers to choose an appropriate media interface for his/her application without the need for a large pin-count package. Two muxes are included in the 5501 to control the configuration of these dual-function pins: the Parallel Port Mux and the Host Port Mux. The state of these muxes is set at reset based on the state of the GPIO6 pin. The External Bus Selection Register (XBSR) shows the configuration of these muxes after the 5501 comes out of reset.

3.3.1 Parallel Port Mux

The Parallel Port Mux of the 5501 controls the function of 20 address signals (pins A[21:2]), 16 data signals (pins D[15:0]), and 16 control signals (pins C0 through C15). The Parallel Port Mux supports two different modes:

- **Full EMIF mode:** The EMIF is enabled and its 20 address, 16 data, and 16 control signals are routed to their corresponding pins on the Parallel Port Mux.
- **Parallel general-purpose I/O mode:** The EMIF and HPI are disabled and 16 control and 4 address pins of the Parallel Port Mux are set to parallel general-purpose I/O (PGPIO).

The mode of the Parallel Port Mux is determined by the state of the GPIO6 pin at reset. If GPIO6 is low, the EMIF and the HPI will be disabled: pins A[17:2] and pins D[15:0] will become reserved pins. All other pins in the Parallel Port Mux are set to parallel general-purpose I/O. The Parallel/Host Port Mux Mode bit field in the External Bus Selection Register (XBSR) will also be set to 0 to reflect the PGPIO mode of the Parallel Port Mux.

If GPIO6 is high at reset, the HPI will be enabled in multiplexed mode and the EMIF will be fully enabled: pins A[21:2] are set to EMIF.A[21:2], pins D[15:0] are set to EMIF[31:0], and pins C[15:0] are set to their corresponding EMIF operation. The Parallel/Host Port Mux Mode bit field in the XBSR will be set to 1 to reflect the full EMIF mode of the Parallel Port Mux. Note that in multiplexed mode, the HPI will use the HD[7:0] pins to strobe in address and data information (see Section 3.7, Host-Port Interface (HPI), for more information on the operation of the HPI in multiplexed mode).

Table 3–4 lists the individual routing of the EMIF and PGPIO signals to the external parallel address, data, and control buses.

Table 3–4. TMS320VC5501 Routing of Parallel Port Mux Signals

PIN	PARALLEL PORT MUX MODE = 0 (PGPIO)	PARALLEL PORT MUX MODE = 1 (FULL EMIF)
Address Bus		
A[17:2]	Reserved	EMIF.A[17:2]
A[21:18]	PGPIO[3:0]	EMIF.A[21:18]
Data Bus		
D[15:0]	Reserved	EMIF.D[15:0]
Control Bus		
C0	PGPIO20	EMIF.ARE/SADS/SDCAS/SRE
C1	PGPIO21	EMIF.AOE/SOE/SDRAS
C2	PGPIO22	EMIF.AWE/SWE/SDWE
C3	PGPIO23	EMIF.ARDY
C4	PGPIO24	EMIF.CE0
C5	PGPIO25	EMIF.CE1
C6	PGPIO26	EMIF.CE2
C7	PGPIO27	EMIF.CE3
C8	PGPIO28	EMIF.BE0
C9	PGPIO29	EMIF.BE1
C10	PGPIO30	EMIF.BE2
C11	PGPIO31	EMIF.BE3
C12	PGPIO32	EMIF.SDCKE
C13	PGPIO33	EMIF.SOE3
C14	PGPIO34	EMIF.HOLD
C15	PGPIO35	EMIF.HOLDA

3.3.2 Host Port Mux

The 5501 Host Port Mux controls the function of 8 data signals (pins HD[7:0]) and 2 control signals (pins HC0 and HC1). The Host Port Mux supports two different modes:

- **8-bit multiplexed mode:** The HPI's 8 data and 2 control signals are routed to their corresponding pins on the Host Port Mux.
- **Parallel general-purpose I/O mode:** All pins on the Host Port Mux are routed to PGPIO. The HPI and EMIF are disabled.

The mode of the Host Port Mux is determined by the state of the GPIO6 pin at reset. If GPIO6 is low, the pins of the Host Port Mux will be set to PGPIO. In this mode, the EMIF and the HPI will be disabled. The Parallel/Host Port Mux Mode bit of the External Bus Control Register will be set to 0 to reflect the PGPIO mode of the Host Port Mux.

If GPIO6 is high, the HPI will be enabled in 8-bit (multiplexed) mode: pins HD[7:0] are set to HPI.HD[7:0], and HC0 and HC1 are set to HPI.$\overline{\text{HAS}}$ and HPI.HBIL, respectively. The Parallel/Host Port Mux Mode bit field in the XBSR will be set to 1 to reflect the HPI multiplexed mode of the Host Port Mux. See Section 3.7, Host-Port Interface (HPI), for more information on the operation of the HPI in multiplexed mode.

Table 3–5 lists the individual routing of the HPI and PGPIO signals to the Host Port Mux pins.

Table 3–5. TMS320VC5501 Routing of Host Port Mux Signals

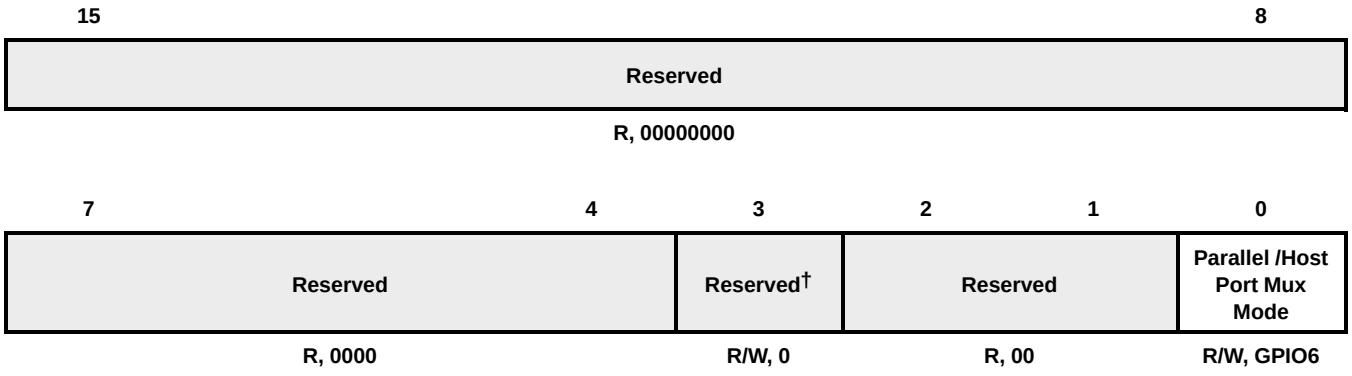
PIN	HOST PORT MUX MODE = 0 (PGPIO)	HOST PORT MUX MODE = 1 (8-BIT HPI MULTIPLEXED)
Data Bus		
HD[7:0]	PGPIO[43:36]	HPI.HD[7:0]
Control Bus		
HC0	PGPIO44	HPI.$\overline{\text{HAS}}$
HC1	PGPIO45	HPI.HBIL

3.3.3 External Bus Selection Register (XBSR)

The External Bus Selection Register controls the mode of the Parallel Port Mux and Host Port Mux. The Parallel Port Mux can be configured to support the 16-bit EMIF or to support parallel general-purpose I/O. The Host Port Mux can be configured to support the HPI in 8-bit (multiplexed) mode or parallel general-purpose I/O (PGPIO).

The XBSR configures the Parallel Port Mux and the Host Port Mux at reset based on the state of the GPIO6 pin at reset. When GPIO6 is high at reset, the Parallel Port Mux will be configured to support the 16-bit EMIF and the Host Port Mux will be configured to support the HPI in 8-bit (multiplexed) mode. When GPIO6 is low at reset, both the Parallel Port Mux and the Host Port Mux will be configured to support parallel general-purpose I/O; the EMIF and HPI will be disabled in this mode. The Paralle/Host Port Mux Mode bit of the XBSR will reflect the mode selected for the Parallel and Host Port Muxes.†

The clock to the EMIF module is disabled automatically when this module is not selected through the External Bus Selection Register. Note that any accesses to disabled modules will result in a bus error if the PERITOEN bit of the Time-Out Control Register is set to 1.



LEGEND: R = Read, W = Write, *n* = value at reset

† This reserved bit must be kept as zero during any writes to XBSR.

Figure 3–3. External Bus Selection Register Layout (0x6C00)

† Modifying the XBSR to change the mode of the Parallel Port Mux and Host Port Mux after the 5501 has been brought out of reset is *not* recommended.

Table 3–6. External Bus Selection Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–4	R	000000000000	Reserved
Reserved	3	R/W	0	Reserved. This reserved bit must be kept as zero during any writes to XBSR.
Reserved	2–1	R	00	Reserved
Parallel/Host Port Mux Mode	0	R/W	GPIO6	Parllel/Host Port Mux Mode bit. Determines the mode of the Parallel Port Mux and the Host Port Mux. • Parallel/Host Port Mux Mode = 0: The Parallel Port Mux is configured to support PGPIO. In this mode, the HPI and EMIF cannot be used. The Host Port Mux is configured to support PGPIO. In this mode, the Host Port Mux pins will be routed to PGPIO. • Parallel/Host Port Mux Mode = 1: The Parallel Port Mux is configured to support the 16-bit EMIF. In this mode, the EMIF is enabled and its 20 address, 16 data, and 16 control signals are routed to their corresponding pins on the Parallel Port Mux. The Host Port Mux is configured to support the HPI in 8-bit (multiplexed) mode. In this mode, the HPI is enabled and its eight data/address and two control signals are routed to their corresponding pins on the Host Port Mux.

3.4 Timers

The 5501 has four 64-bit timers: Timer 0, Timer 1, Watchdog Timer (WDT), and Timer 3. The input/output pin of Timer 0, Timer 1, and the Watchdog Timer can be configured as an input or an output via the Timer Signal Selection Register (TSSR). Furthermore, the TSSR can be used to connect the Watchdog Timer output to $\overline{\text{NMI}}$, $\overline{\text{RESET}}$, or $\overline{\text{INT3}}$.

- The first two timers, Timer 0 and Timer 1, are mainly used as general-purpose timers.
- The third timer, the Watchdog Timer, can be used as either a general-purpose timer or a watchdog timer. The output pin of the Watchdog Timer, WDTOUT, is multiplexed with the $\overline{\text{NMI}}$ input pin; its function can be controlled via the NMI/WDTOUT_CFG bit of the TSSR. If the $\overline{\text{NMI}}$/WDTOUT pin is configured as the Watchdog Timer pin, it can be configured as an input or an output via the WDT_MODE bit of the TSSR. The Watchdog Timer output can also be internally connected to the $\overline{\text{NMI}}$, $\overline{\text{RESET}}$, and $\overline{\text{INT3}}$ signals of the 5501 via the IWCON bits of the TSSR.
- The fourth timer is reserved as a DSP/BIOS counter. This timer has no input or output pin. No interrupts are needed from this timer; therefore, the timer output is not internally connected to the CPU interrupt logic.

3.4.1 Timer Signal Selection Register (TSSR)

The Timer Signal Selection Register (TSSR) controls several pin characteristics for Timer 0, Timer 1, and the Watchdog Timer. The TSSR can be used to specify whether the pins of Timer 0, Timer 1, and the Watchdog Timer are input or output. The TSSR also determines how the output signal of the Watchdog Timer is connected internally and sets the function for the $\overline{\text{NMI}}$/WDTOUT pin of the 5501. By default, all the timer pins (TIM0, TIM1, and $\overline{\text{NMI}}$/WDTOUT) are set as inputs, the output of the Watchdog Timer is not internally connected to anything, and the $\overline{\text{NMI}}$/WDTOUT pin has the function of the $\overline{\text{NMI}}$ signal.

15	14	13	12	11	10	9	8
Reserved							
R, 00000000							

7	6	5	4	3	2	1	0
Reserved		WDT_MODE	TIM1_MODE	TIM0_MODE	IWCON		NMI/WDTOUT_CFG
R, 00		R/W, 0	R/W, 0	R/W, 0	R/W, 00		R/W, 1

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–4. Timer Signal Selection Register Layout (0x8000)

Table 3–7. Timer Signal Selection Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–6	R	0000000000	Reserved
WDT_MODE	5	R/W	0	WDT pin mode WDT_MODE = 0: WDTOUT pin is used as the timer input pin. WDT_MODE = 1: WDTOUT pin is used as the timer output pin.
TIM1_MODE	4	R/W	0	TIM1 pin mode TIM1_MODE = 0: TIM1 pin is used as the timer input pin. TIM1_MODE = 1: TIM1 pin is used as the timer output pin.
TIM0_MODE	3	R/W	0	TIM0 pin mode TIM0_MODE = 0: TIM0 pin is used as the timer input pin. TIM0_MODE = 1: TIM0 pin is used as the timer output pin.

† If NMI/WDTOUT_CFG = 1 and IWCON = 10, only the WDTOUT signal will drive the $\overline{\text{NMI}}$ signal; the external source driving the $\overline{\text{NMI}}$/WDTOUT pin will be ignored (see Figure 3–5).

PRODUCT PREVIEW

Table 3–7. Timer Signal Selection Register Bit Field Description (Continued)

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
IWCON	2:1	R/W	00	Internal WDT output signal connection IWCON = 00: Internal watchdog timer output signal has no internal connection. IWCON = 01: Internal watchdog timer output signal has an internal connection to $\overline{\text{RESET}}$ pin. IWCON = 10: Internal watchdog timer output signal has an internal connection to $\overline{\text{NMI}}$ pin.† IWCON = 11: Internal watchdog timer output signal has an internal connection to $\overline{\text{INT3}}$ pin.
NMI/WDTOUT_CFG	0	R/W	1	$\overline{\text{NMI}}$/WDTOUT configuration NMI/WDTOUT_CFG = 0: $\overline{\text{NMI}}$/WDTOUT pin is used as the WDTOUT pin. NMI/WDTOUT_CFG = 1: $\overline{\text{NMI}}$/WDTOUT pin is used as the $\overline{\text{NMI}}$ input pin.†

† If NMI/WDTOUT_CFG = 1 and IWCON = 10, only the WDTOUT signal will drive the $\overline{\text{NMI}}$ signal; the external source driving the $\overline{\text{NMI}}$/WDTOUT pin will be ignored (see Figure 3–5).

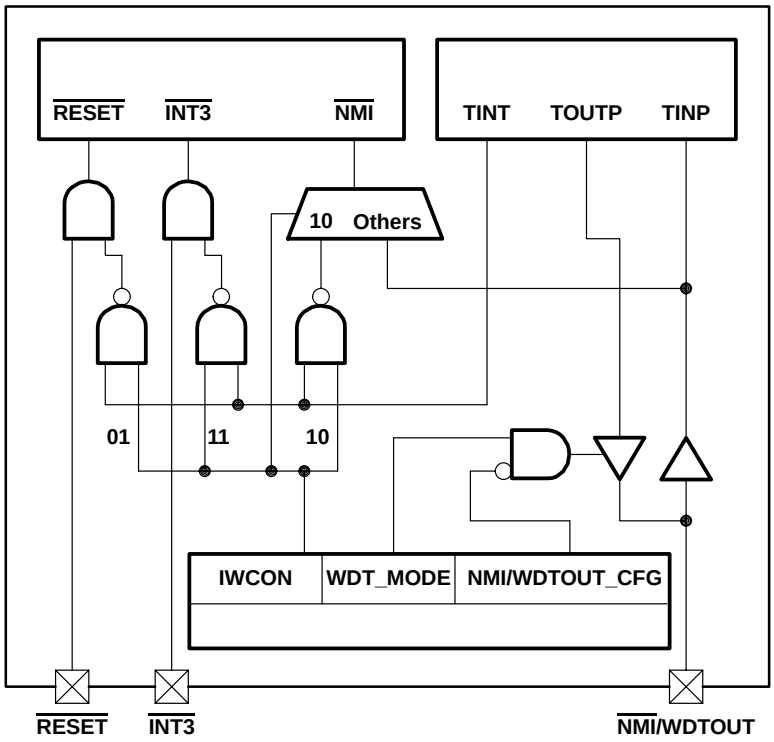


Figure 3–5. Watchdog Timer Hook-Up

3.5 Universal Asynchronous Receiver/Transmitter (UART)

The UART peripheral is based on the industry-standard TL16C550B asynchronous communications element, which in turn, is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the CPU of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes, including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device or modem and parallel-to-serial conversion on data received from the CPU. The CPU can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be configured to minimize software management of the communications link.

The UART includes a programmable baud rate generator capable of dividing the CPU clock by divisors from 1 to 65535 and producing a 16× reference clock for the internal transmitter and receiver logic.

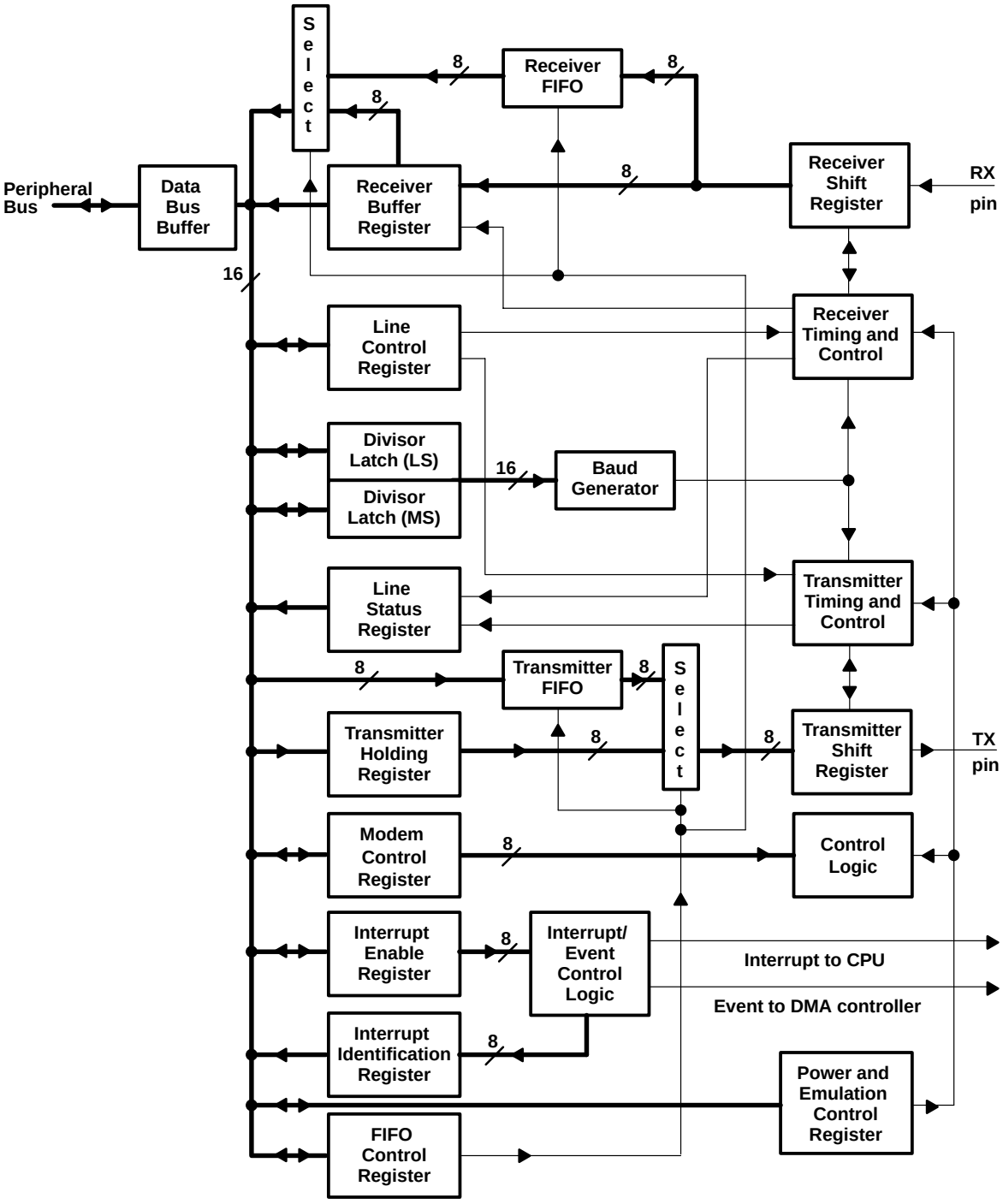


Figure 3–6. UART Functional Block Diagram

3.6 Inter-Integrated Circuit (I2C) Module

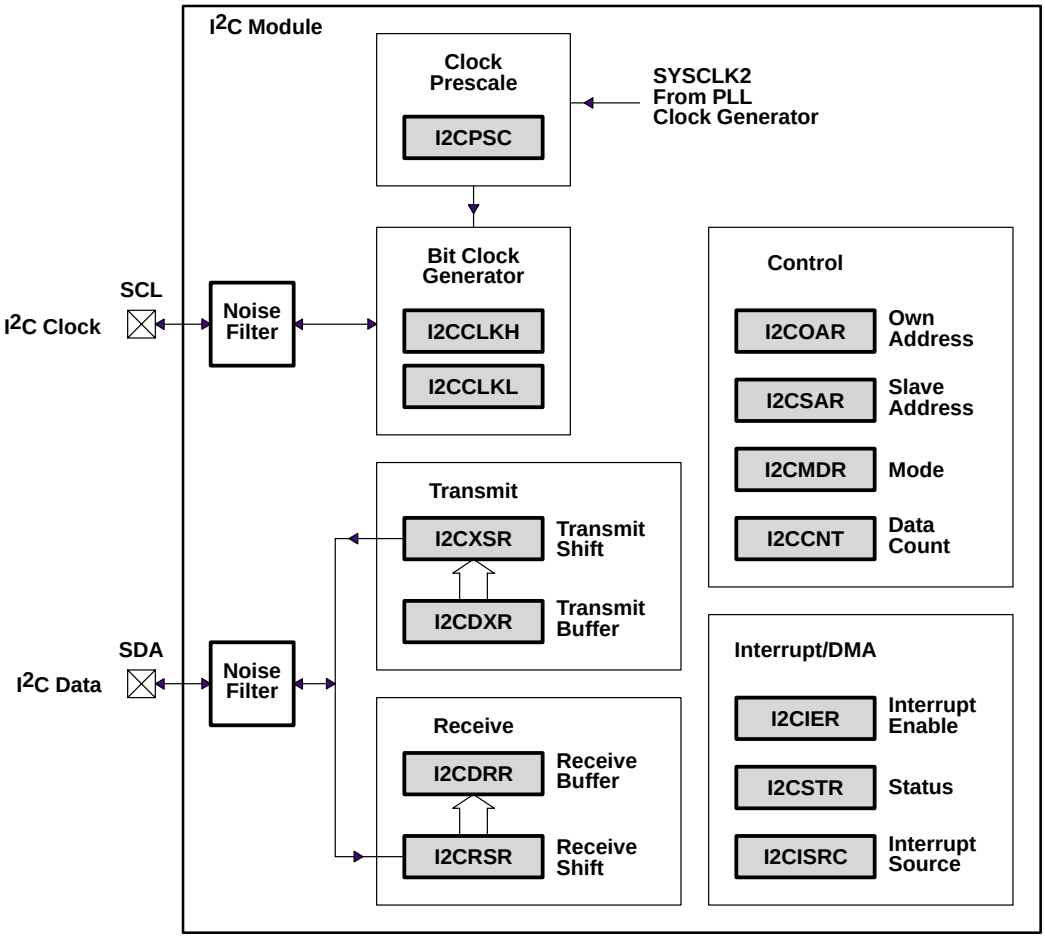
The TMS320VC5501 also includes an I2C serial port for control purposes. Features of the I2C port include:

- Compatibility with Philips™ I2C-Bus™ Specification, Version 2.1 (January 2000)
- Fast mode up to 400 Kbps (no fail-safe I/O buffers)
- Noise filters (on the SDA and SCL pins) to suppress noise of 50 ns or less (I2C module clock *must* be in the range of 7 MHz to 12 MHz)
- 7-bit and 10-bit device addressing modes
- Master (transmit/receive) and slave (transmit/receive) functionality
- Events: DMA, interrupt, or polling
- Slew-rate limited open-drain output buffers

The I2C module clock *must* be in the range of 7 MHz to 12 MHz. This is necessary for the proper operation of the I2C module.

NOTE: For additional information, see *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317).

Figure 3–7 is a block diagram of the I2C module.



NOTE A: Shading denotes control/status registers.

Figure 3–7. I2C Module Block Diagram

Philips and I2C Bus are trademarks of Koninklijke Philips Electronics N.V.

3.7 Host-Port Interface (HPI)

The 5501 HPI provides an 8-bit parallel interface (multiplexed mode) to a host with the following features:

- Host access to on-chip DARAM (excluding CPU memory-mapped registers)
- 16-bit address register with autoincrement capability for faster transfers
- Multiple address/data strobes provide a glueless interface to a variety of hosts
- HRDY signal for handshaking with host

The 5501 HPI can access all of internal DARAM space excluding memory-mapped CPU registers (which are located at word addresses 0x0000–0x0059), but it cannot access peripheral registers or external memory. Note that all memory accesses made through the HPI are word-addressed. The HPI cannot access internal DARAM space when the device is in reset.

The 5501 HPI only supports data transfers in multiplexed 8-bit mode. In multiplexed mode, the host can only send 8 bits of data at a time through the HD[7:0] bus; therefore, some extra steps have to be taken to read/write from the 5501's internal memory [see the *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317) for more information on the 5501 HPI].

NOTE: No host access should occur when the HPI is placed in IDLE. The host cannot wake up the DSP through the DSP_INT bit of the HPIC1 register when the DSP is in IDLE mode.

The 5501 HPI has its own register set, therefore the HINT bit of CPU register ST3_55 is not used for DSP-to-host interrrupts. The HINT bit in the Host Port Control Register (HPIC) should be used for DSP-to-host interrupts.

3.8 Direct Memory Access (DMA) Controller

The 5501 DMA provides the following features:

- Four standard ports for the following data resources: two for DARAM, one for Peripherals, and one for External Memory
- Six channels, which allow the DMA controller to track the context of six independent DMA channels
- Programmable low/high priority for each DMA channel
- One interrupt for each DMA channel
- Event synchronization. DMA transfers in each channel can be dependent on the occurrence of selected events.
- Programmable address modification for source and destination addresses
- Idle mode that allows the DMA controller to be placed in a low-power (idle) state under software control

The 5501 has an Auto-wakeup/Idle function for McBSP to DMA to on-chip memory data transfers when the DMA and the McBSP are both set to IDLE. In the case that the McBSP is set to external clock mode and the McBSP and the DMA are set to idle, the McBSP and the DMA can wake up from IDLE state automatically if the McBSP gets a new data transfer. The McBSP and the DMA enter the idle state automatically after data transfer is complete. [The clock generator (PLL) should be active and the PLL core should not be in power-down mode for the Auto-wakeup/Idle function to work.]

The 5501 DMA controller allows transfers to be synchronized to selected events. The 5501 supports 14 separate synchronization events and each channel can be tied to separate synchronization event independent of the other channels. Synchronization events are selected by programming the SYNC field in the channel-specific DMA Channel Control Register (DMA_CCR).

The 5501 DMA can access all the internal DARAM space as well as all external memory space. The 5501 DMA also has access to the registers for the following peripheral modules: McBSP, UART, GPIO, PGPIO, and I^2C.

3.8.1 DMA Channel 0 Control Register (DMA_CCR0)

The DMA Channel 0 Control Register (DMA_CCR0) bit layouts are shown in Figure 3–8. DMA_CCR1 to DMA_CCR5 have similar bit layouts. See the *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317) for more information on the DMA Channel n Control Register (n = 0, 1, 2, 3, 4, or 5).

15	14	13	12	11	10	9	8
DSTAMODE		SRCAMODE		ENDPROG	WP	REPEAT	AUTOINIT
R/W, 00		R/W, 00		R/W, 0	R/W, 0	R/W, 0	R/W, 0

7	6	5	4	0
EN	PRIO	FS	SYNC	
R/W, 0	R/W, 0	R/W, 0	R/W, 00000	

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–8. DMA Channel 0 Control Register Layout (0x0C01)

PRODUCT PREVIEW

The SYNC field (bits[4:0]) of the DMA_CCR register specifies the event that can initiate the DMA transfer for the corresponding DMA channel. The five bits allow several configurations as listed in Table 3–8. The bits are set to zero upon reset.

Table 3–8. Synchronization Control Function

SYNC FIELD IN DMA_CCR	SYNCHRONIZATION MODE
00000b	No event synchronized
00001b	McBSP 0 Receive Event (REVT0)
00010b	McBSP 0 Transmit Event (XEVT0)
00011b	Reserved (Do not use this value)
00100b	Reserved (Do not use this value)
00101b	McBSP1 Receive Event (REVT1)
00110b	McBSP1 Transmit Event (XEVT1)
00111b	Reserved (Do not use this value)
01000b	Reserved (Do not use this value)
01001b	Reserved (Do not use this value)
01010b	Reserved (Do not use this value)
01011b	UART Receive Event (UARTREVT)
01100b	UART Transmit Event (UARTXEVT)
01101b	Timer 0 Event
01110b	Timer 1 Event
01111b	External Interrupt 0
10000b	External Interrupt 1
10001b	External Interrupt 2
10010b	External Interrupt 3
10011b	I^2C Receive Event
10100b	I^2C Transmit Event
Other values	Reserved (Do not use these values)

3.9 System Clock Generator

The TMS320VC5501 includes a flexible clock generator module consisting of a PLL and oscillator, with several dividers so that different clocks may be generated for different parts of the system (i.e., DSP core, Internal Peripheral Bus, External Memory Interface).

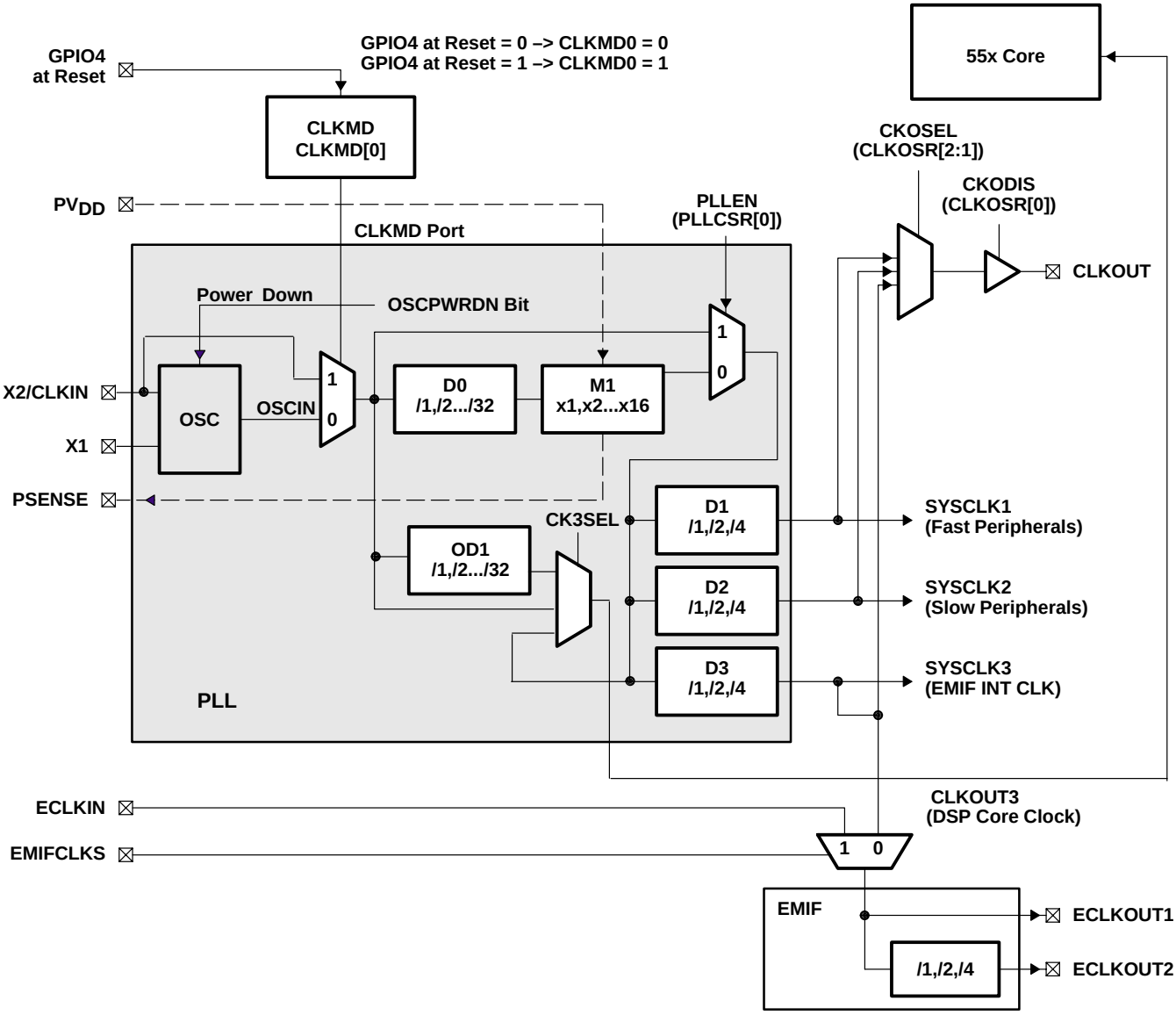


Figure 3–9. PLL and Clock Generator Logic

3.9.1 Input Clock Source

The clock input to the 5501 can be sourced from either an externally generated 3.3-V clock input on the X2/CLKIN pin, or from the on-chip oscillator if an external crystal circuit is attached to the device as shown in Figure 3–10. At reset, if GPIO4 is low, the internal oscillator will be enabled and the internal oscillator and an external crystal will generate the input clock. If GPIO4 is high, the internal oscillator will be set to power-down mode and the input clock will be taken from the X2/CLKIN pin. The CLKMD0 bit of the Clock Mode Control Register (CLKMD) will reflect the state of GPIO4 after reset. Please note that after reset, the GPIO4 pin may become active depending on the boot mode selected through the GPIO[2:0] pins.

3.9.1.1 Internal System Oscillator With External Crystal

The oscillator requires an external crystal or ceramic resonator connected across the X1 and X2/CLKIN pins. If the internal oscillator is not used, an external clock source must be applied to the X2/CLKIN pin and the X1 pin should be left unconnected. Since the internal oscillator can be used as a clock source to the PLL, the crystal oscillation frequency can be multiplied to generate the input clock to the different clock groups.

The crystal should be in fundamental-mode operation, and parallel resonant, with a maximum effective series resistance of 30 Ω and a power dissipation of 1 mW. The internal oscillator supports fundamental-mode crystals up to 25 MHz. The connection of the required circuit, consisting of the crystal and two load capacitors, is shown in Figure 3–10. The load capacitors, C1 and C2, should be chosen such that the equation below is satisfied. C_L in the equation is the load specified for the crystal, which must be 10 pF.

$$C_L = \frac{C_1 C_2}{(C_1 + C_2)}$$

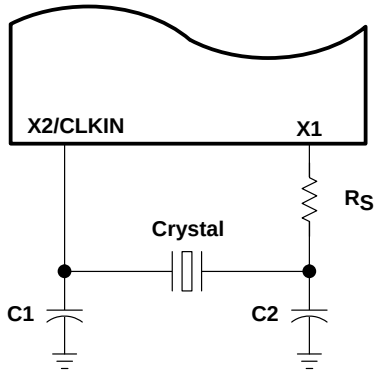


Figure 3–10. Internal System Oscillator External Crystal

The 5501 has internal circuitry that will count down a predetermined number of clock cycles (41,032 reference clock cycles) to allow the oscillator input to become stable after waking up from power-down state or after reset. If a reset is asserted, program flow will start after all stabilization periods have expired; this includes the oscillator stabilization period only if GPIO4 is low at reset. If the oscillator is coming out of power-down mode, program flow will start immediately after the oscillator stabilization period has completed. See Section 3.9.6, Reset Sequence, for more details on program flow after reset or after oscillator power-down. See Section 3.10, Idle Control, for more information on the oscillator power-down mode.

3.9.1.2 Clock Generation With PLL Disabled (Bypass Mode, Default)

After reset, the PLL multiplier (M1) and its divider (D0) will be bypassed by default and the input clock to point C in Figure 3–11 will be taken from, depending on the state of the GPIO4 pin after reset, either the internal oscillator or the X2/CLKIN pin. The PLL can be taken out of bypassed mode as described in Section 3.9.4.1, C55x Subsystem Clock Group.

3.9.1.3 Clock Generation With PLL Enabled (PLL Mode)

When not in bypass mode, the frequency of the input clock can be divided down by a programmable divider (D0) by any factor from 1 to 32. The output clock of the divider can be multiplied by any factor from 1 to 16 through a programmable multiplier (M1). The divider factor can be set through the PLLDIV0 bit of the PLL Divider 0 Register. The multiplier factor can be set through the PLLM bits of the PLL Multiplier Control Register. By default, the divider and multiplier factors are set to 1 after reset.

NOTE: There is a specific minimum and maximum input clock for the block labeled PLL in Figure 3–9, as well as for the DSP subsystem, peripherals, and EMIF. In addition, there is a maximum output frequency for the PLL. The clock generator must not be configured to exceed any of these constraints (certain combinations of external clock input, internal dividers, and PLL multiply ratios might not be supported).

3.9.2 Clock Groups

The TMS320VC5501 has four clock groups: the C55x Subsystem Clock Group, the Fast Peripherals Clock Group, the Slow Peripherals Clock Group, and the External Memory Interface Clock Group. Clock groups allow for lower power and performance optimization since the frequency of groups with no high-speed requirements can be set to 1/4 or 1/2 the frequency of other groups.

3.9.2.1 C55x Subsystem Clock Group

The C55x Subsystem Clock Group includes the C55x CPU core, internal memory (DARAM and ROM), the ICACHE, and all CPU-related modules. The input clock to this clock group is taken from the CLKOUT3 signal (as shown in Figure 3–9), the source of which can be controlled through the CLKOUT3 Select Register (CK3SEL). The frequency of CLKOUT3 can be set by adjusting the divider and multiplier values of D0 and M1 through the PLLDIV0 and PLLM registers, respectively.

3.9.2.2 Fast Peripherals Clock Group

The Fast Peripherals Clock Group includes the DMA, HPI, and the timers. The input clock to this clock group is taken from the output of divider 1 (D1) (as shown in Figure 3–9). By default, the divider is set to divide its input clock by four, but the divide value can be changed to divide-by-1 or divide-by-2 by modifying the PLLDIV1 bits of the PLL Divider1 Register (PLLDIV1) through software.

3.9.2.3 Slow Peripherals Clock Group

The Slow Peripherals Clock Group includes the McBSPs, I^2C, and the UART. The input clock to this clock group is taken from the output of divider 2 (D2). by default, the divider is set to divide its input clock by four, but the divide value can be changed to divide-by-1 or divide-by-2 by modifying the PLLDIV2 bits of the PLL Divider2 Register (PLLDIV2) through software. *The clock frequency of the Slow Peripherals Clock Group must be equal to or less than that of the Fast Peripherals Clock Group.*

3.9.2.4 External Memory Interface Clock Group

The External Memory Interface Clock Group includes the External Memory Interface (EMIF) module and the external data bridge modules. The input clock to this clock group is taken from the output of divider 3 (D3). By default, the divider is set to divide its input clock by four, but the divide value can be changed to divide-by-1 or divide-by-2 by modifying the PLLDIV3 bits of the PLL Divider3 Register (PLLDIV3) through software. *The clock frequency of the External Memory Interface Clock Group must be equal to or less than that of the Fast Peripherals Clock Group.*

3.9.3 EMIF Input Clock Selection

The EMIF may be clocked from an external asynchronous clock source through the ECLKIN pin if a specific EMIF frequency is needed. The source for the EMIF clock can be specified at reset through the EMIFCLKS pin. If EMIFCLKS is low at reset, then the EMIF will be clocked via the same internal clock that feeds the data bridge module and performance will be optimal. If EMIFCLKS is high at reset, then an external asynchronous clock, which can be taken up to 100 MHz (TBD), will clock the EMIF. The data throughput performance may be degraded due to synchronization issues when an external clock source is used for the EMIF.

3.9.4 Changing the Clock Group Frequencies

DSP software can be used to change the clock frequency of each clock group by setting adequate values in the PLL control registers. Figure 3–11 shows which PLL control registers affect the different portions of the clock generator. The following sections describe the procedures for changing the frequencies of each clock group.

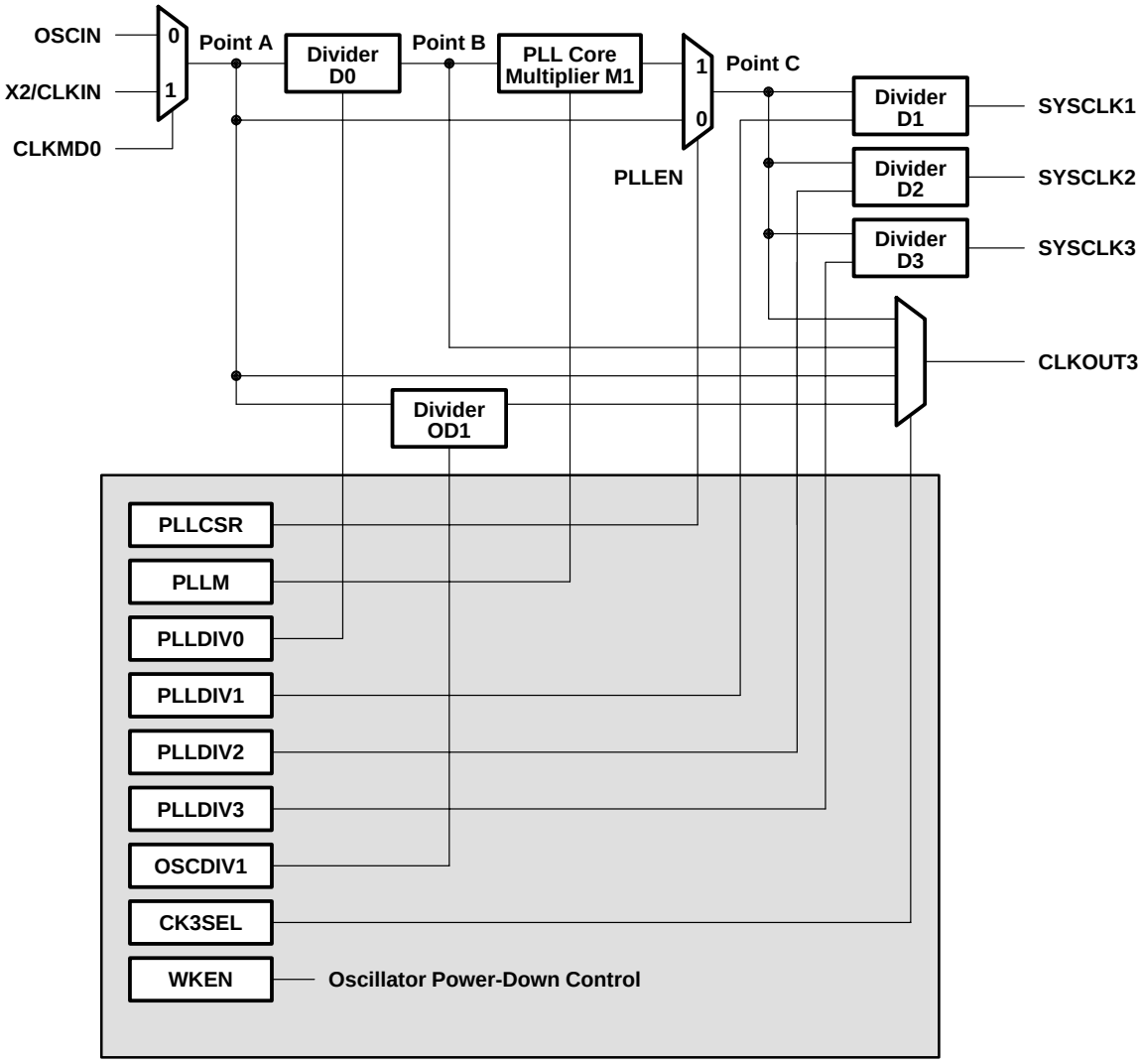


Figure 3–11. Clock Generator (PLL) Registers

3.9.4.1 C55x Subsystem Clock Group

Changes to the PLL Control Register (PLLCSR), the PLL Divider0 Register (PLLDIV0), and the PLL Multiplier Register (PLLM) affect the clock of this clock group. The following procedure must be followed to change or to set the PLL to a specific value:

1. Switch to bypass mode by setting the PLLEN bit to 0.
2. Set the PLL to its reset state by setting the PLLRST bit to 1.
3. Change the PLL setting through the PLLM and PLLDIV0 bits.
4. Wait for 1 μs.
5. Release the PLL from its reset state by setting PLLRST to 0.
6. Wait for the PLL to relock by polling the LOCK bit or by setting up a LOCK interrupt.
7. Switch back to PLL mode by setting the PLLEN bit to 1.

The frequency of the C55x Subsystem Clock Group can be up to 300 MHz.

3.9.4.2 Fast Peripherals Clock Group

Changes to the clock of the C55x Subsystem Clock Group affect the clock of the Fast Peripherals Clock Group. The PLLDIV1 value of the PLL Divider1 Register (PLLDIV1) should not be set in a manner that makes the frequency for this clock group greater than 150 MHz. There must be no activity in the modules included in the Fast Peripherals Clock Group when the value of PLLDIV1 is being changed. It is recommended that the fast peripheral modules be put in IDLE mode before changing the PLLDIV1 value.

3.9.4.3 Slow Peripherals Clock Group

Changes to the clock of the C55x Subsystem Clock Group affect the clock of the Slow Peripherals Clock Group. The PLLDIV2 value of the PLL Divider2 Register (PLLDIV2) should not be set in a manner that makes the frequency for this clock group greater than 150 MHz or greater than the frequency of the Fast Peripherals Clock Group. There must be no activity in the modules included in the Slow Peripherals Clock Group when the value of PLLDIV2 is being changed. It is recommended that the slow peripheral modules be put in IDLE mode before changing the PLLDIV2 value.

3.9.4.4 External Memory Interface Clock Group

Changes to the clock of the C55x Subsystem Clock Group affect the clock of the External Memory Interface Clock Group. The PLLDIV3 value of the PLL Divider3 Register (PLLDIV3) should not be set in a manner that makes the frequency for this clock group greater than 100 MHz or greater than the frequency of the Fast Peripherals Clock Group, whichever is smaller. If an external clock is used, the EMIF can operate up to 100 MHz as long as the external clock frequency does not exceed the frequency of the internal clock to the data bridge modules or the frequency of the Fast Peripherals Clock Group, whichever is smaller. There must be no external memory accesses when the value of PLLDIV3 is being changed, this means that the value of PLLDIV3 cannot be changed by a program that is being executed from external memory. It is recommended that the EMIF be put in IDLE mode before changing the PLLDIV3 value.

3.9.5 PLL Control Registers

The 5501 PLL control registers are accessible via the I/O memory map.

Table 3–9. PLL Control Registers

ADDRESS	REGISTER
1C80h	PLLCSR
1C82h	CK3SEL
1C88h	PLLM
1C8Ah	PLLDIV0
1C8Ch	PLLDIV1
1C8Eh	PLLDIV2
1C90h	PLLDIV3
1C92h	OSCDIV1
1C98h	WKEN

3.9.5.1 PLL Control / Status Register (PLLCSR)

15							8
Reserved							
R, 00000000							

7	6	5	4	3	2	1	0
Reserved	STABLE	LOCK	Reserved	PLLRST	OSCPWRDN	PLLPWRDN	PLLEN
R, 0	R, 1	R, 0	R, 0	R/W, 1	R/W, 0	R/W, 0	R/W, 0

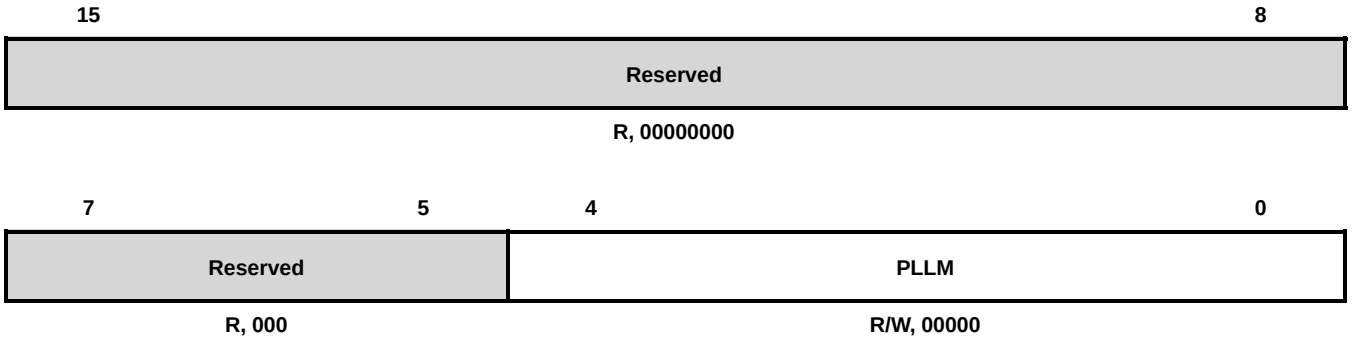
LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–12. PLL Control/Status Register Layout (0x1C80)

Table 3–10. PLL Control/Status Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15:7	R	000000000	Reserved. Reads return 0. Writes have no effect.
STABLE	6	R	1	Oscillator input stable. This bit indicates if the OSCIN/CLKIN input has stabilized. STABLE = 0: OSCIN/CLKIN input is not yet stable. Oscillator counter is not done counting 41,032 reference clock cycles. STABLE = 1: OSCIN/CLKIN input is stable. This is true if any one of the three cases is true: a) Oscillator counter has finished counting. b) Oscillator counter is disabled. c) Test mode.
LOCK	5	R	0	Lock mode indicator. This bit indicates whether the clock generator is in its lock mode. LOCK = 0: The PLL is in the process of getting a phase lock. LOCK = 1: The clock generator is in the lock mode. The PLL has a phase lock and the output clock of the PLL has the frequency determined by the PLLM register and PLLDIV0 register.
Reserved	4	R	0	Reserved. Reads return 0. Writes have no effect.
PLLRST	3	R/W	1	Asserts RESET to PLL PLLRST = 0: PLL reset released PLLRST = 1: PLL reset asserted
OSCPWRDN	2	R/W	0	Selects oscillator power down OSCPWRDN = 0: Oscillator operational OSCPWRDN = 1: Oscillator placed in power-down state
PLLPWRDN	1	R/W	0	Selects PLL power down PLLPWRDN = 0: PLL operational PLLPWRDN = 1: PLL placed in power-down state
PLLEN	0	R/W	0	PLL mode enable. This bit controls the multiplexer before dividers D1, D2, and D3. PLLEN = 0: Bypass mode. Divider D1 and PLL are bypassed. SYSCLK1 to 3 divided down directly from input reference clock. PLLEN = 1: PLL mode. Divider D1 and PLL are not bypassed. SYSCLK1 to 3 divided down from PLL output.

3.9.5.2 PLL Multiplier Control Register (PLLM)



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–13. PLL Multiplier Control Register Layout (0x1C88)

Table 3–11. PLL Multiplier Control Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15:5	R	00000000000	Reserved. Reads return 0. Writes have no effect.
PLLM	4:0	R/W	00000	PLL multiplier-select PLLM = 00000–00001: Reserved PLLM = 00010: Times 2 PLLM = 00011: Times 3 PLLM = 00100: Times 4 PLLM = 00101: Times 5 PLLM = 00110: Times 6 PLLM = 00111: Times 7 PLLM = 01000: Times 8 PLLM = 01001: Times 9 PLLM = 01010: Times 10 PLLM = 01011: Times 11 PLLM = 01100: Times 12 PLLM = 01101: Times 13 PLLM = 01110: Times 14 PLLM = 01111: Times 15 PLLM = 10000–11111: Reserved

3.9.5.3 PLL Divider 0 Register (PLLDIV0) (Prescaler)

This register controls the value of the PLL prescaler (Divider D0).

15	14	8
D0EN	Reserved	
R/W, 1	R, 0000000	

7	5	4	0
Reserved		PLLDIV0	
R, 000		R/W, 00000	

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–14. PLL Divider 0 Register Layout (0x1C8A)

Table 3–12. PLL Divider 0 Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
D0EN	15	R/W	1	Divider D0 enable D0EN = 0: Divider 0 disabled D0EN = 1: Divider 0 enabled
Reserved	14:5	R	0000000000	Reserved. Reads return 0. Writes have no effect.
PLLDIV0	4:0	R/W	00000	Divider D0 ratio PLLDIV0 = 00000: Divide by 1 PLLDIV0 = 00001: Divide by 2 PLLDIV0 = 00010: Divide by 3 PLLDIV0 = 00011: Divide by 4 PLLDIV0 = 00100: Divide by 5 PLLDIV0 = 00101: Divide by 6 PLLDIV0 = 00110: Divide by 7 PLLDIV0 = 00111: Divide by 8 PLLDIV0 = 01000: Divide by 9 PLLDIV0 = 01001: Divide by 10 PLLDIV0 = 01010: Divide by 11 PLLDIV0 = 01011: Divide by 12 PLLDIV0 = 01100: Divide by 13 PLLDIV0 = 01101: Divide by 14 PLLDIV0 = 01110: Divide by 15 PLLDIV0 = 01111: Divide by 16 PLLDIV0 = 10000: Divide by 17 PLLDIV0 = 10001: Divide by 18 PLLDIV0 = 10010: Divide by 19 PLLDIV0 = 10011: Divide by 20 PLLDIV0 = 10100: Divide by 21 PLLDIV0 = 10101: Divide by 22 PLLDIV0 = 10110: Divide by 23 PLLDIV0 = 10111: Divide by 24 PLLDIV0 = 11000: Divide by 25 PLLDIV0 = 11001: Divide by 26 PLLDIV0 = 11010: Divide by 27 PLLDIV0 = 11011: Divide by 28 PLLDIV0 = 11100: Divide by 29 PLLDIV0 = 11101: Divide by 30 PLLDIV0 = 11110: Divide by 31 PLLDIV0 = 11111: Divide by 32

3.9.5.4 PLL Divider1 Register (PLLDIV1) for SYSCLK1

This register controls the value of the divider D1 for SYSCLK1. It is in both the BYPASS and PLL paths.

15	14	8
D1EN	Reserved	
R/W, 1	R, 0000000	

7	5	4	0
Reserved		PLLDIV1	
R, 000		R/W, 00011	

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–15. PLL Divider 1 Register Layout (0x1C8C)

Table 3–13. PLL Divider 1 Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
D1EN	15	R/W	1	Divider D1 enable D1EN = 0: Divider 1 disabled D1EN = 1: Divider 1 enabled
Reserved	14:5	R	0000000000	Reserved. Reads return 0. Writes have no effect.
PLLDIV1	4:0	R/W	00011	Divider D1 ratio (SYSCLK1 divider) PLLDIV1 = 00000: Divide by 1 PLLDIV1 = 00001: Divide by 2 PLLDIV1 = 00010: Reserved PLLDIV1 = 00011: Divide by 4 PLLDIV1 = 00100–11111: Reserved

3.9.5.5 PLL Divider2 Register (PLLDIV2) for SYSCLK2

This register controls the value of the divider D2 for SYSCLK2. It is in both the BYPASS and PLL paths.

15	14	8
D2EN	Reserved	
R/W, 1	R, 0000000	

7	5	4	0
Reserved		PLLDIV2	
R, 000		R/W, 00011	

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–16. PLL Divider 2 Register Layout (0x1C8E)

Table 3–14. PLL Divider 2 Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
D2EN	15	R/W	1	Divider D2 enable D2EN = 0: Divider 2 disabled D2EN = 1: Divider 2 enabled
Reserved	14:5	R	0000000000	Reserved. Reads return 0. Writes have no effect.
PLLDIV2	4:0	R/W	00011	Divider D2 ratio (SYSCLK2 divider) PLLDIV2 = 00000: Divide by 1 PLLDIV2 = 00001: Divide by 2 PLLDIV2 = 00010: Reserved PLLDIV2 = 00011: Divide by 4 PLLDIV2 = 00100–11111: Reserved

3.9.5.6 PLL Divider3 Register (PLLDIV3) for SYSCLK3

This register controls the value of the divider D3 for SYSCLK3. It is in both the BYPASS and PLL paths.

15	14	8
D3EN	Reserved	
R/W, 1	R, 0000000	

7	5	4	0
Reserved		PLLDIV3	
R, 000		R/W, 00011	

LEGEND: R = Read, W = Write, *n* = value at reset

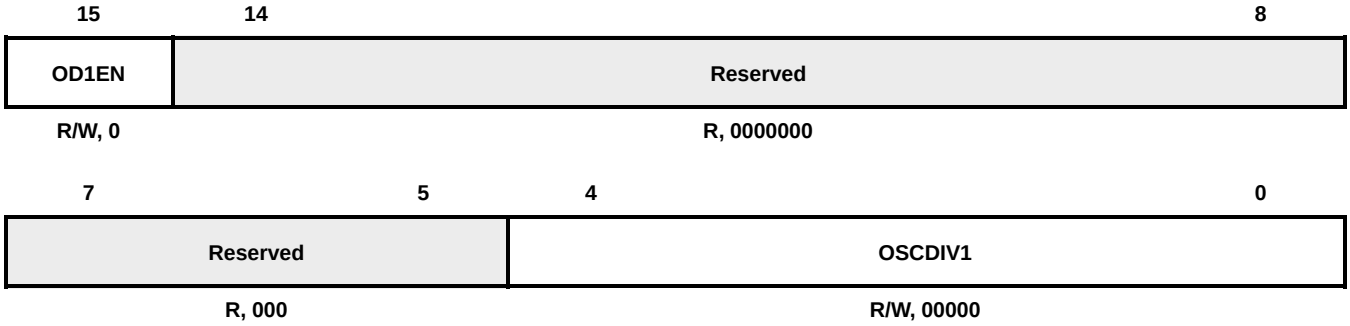
Figure 3–17. PLL Divider 3 Register Layout (0x1C90)

Table 3–15. PLL Divider3 Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
D3EN	15	R/W	1	Divider D3 enable D3EN = 0: Divider 3 disabled D3EN = 1: Divider 3 enabled
Reserved	14:5	R	0000000000	Reserved. Reads return 0. Writes have no effect.
PLLDIV3	4:0	R/W	00011	Divider D3 ratio (SYSCLK3 divider) PLLDIV3 = 00000: Divide by 1 PLLDIV3 = 00001: Divide by 2 PLLDIV3 = 00010: Reserved PLLDIV3 = 00011: Divide by 4 PLLDIV3 = 00100–11111: Reserved

3.9.5.7 Oscillator Divider1 Register (OSCDIV1) for CLKOUT3

This register controls the value of the divider OD1 for CLKOUT3. It does not go through the PLL path.



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–18. Oscillator Divider1 Register Layout (0x1C92)

Table 3–16. Oscillator Divider1 Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
OD1EN	15	R/W	0	Oscillator divider OD1 enable OD1EN = 0: Oscillator divider 1 disabled OD1EN = 1: Oscillator divider 1 enabled
Reserved	14:5	R	0000000000	Reserved. Reads return 0. Writes have no effect.
OSCDIV1	4:0	R/W	00000	Divider OD1 ratio (CLKOUT3 divider) OSCDIV1 = 00000: Divide by 1 OSCDIV1 = 00001: Divide by 2 OSCDIV1 = 00010: Divide by 3 OSCDIV1 = 00011: Divide by 4 OSCDIV1 = 00100: Divide by 5 OSCDIV1 = 00101: Divide by 6 OSCDIV1 = 00110: Divide by 7 OSCDIV1 = 00111: Divide by 8 OSCDIV1 = 01000: Divide by 9 OSCDIV1 = 01001: Divide by 10 OSCDIV1 = 01010: Divide by 11 OSCDIV1 = 01011: Divide by 12 OSCDIV1 = 01100: Divide by 13 OSCDIV1 = 01101: Divide by 14 OSCDIV1 = 01110: Divide by 15 OSCDIV1 = 01111: Divide by 16 OSCDIV1 = 10000: Divide by 17 OSCDIV1 = 10001: Divide by 18 OSCDIV1 = 10010: Divide by 19 OSCDIV1 = 10011: Divide by 20 OSCDIV1 = 10100: Divide by 21 OSCDIV1 = 10101: Divide by 22 OSCDIV1 = 10110: Divide by 23 OSCDIV1 = 10111: Divide by 24 OSCDIV1 = 11000: Divide by 25 OSCDIV1 = 11001: Divide by 26 OSCDIV1 = 11010: Divide by 27 OSCDIV1 = 11011: Divide by 28 OSCDIV1 = 11100: Divide by 29 OSCDIV1 = 11101: Divide by 30 OSCDIV1 = 11110: Divide by 31 OSCDIV1 = 11111: Divide by 32

3.9.5.8 Oscillator Wakeup Control Register (WKEN)

This register controls whether different events in the system are enabled to wake up the device after entering OSCPWRDN.

15						8
Reserved						
R, 00000000						

7	5	4	3	2	1	0
Reserved		WKEN4	WKEN3	WKEN2	WKEN1	WKEN0
R, 000		R/W, 1	R/W, 1	R/W, 1	R/W, 1	R/W, 1

LEGEND: R = Read, W = Write, *n* = value at reset

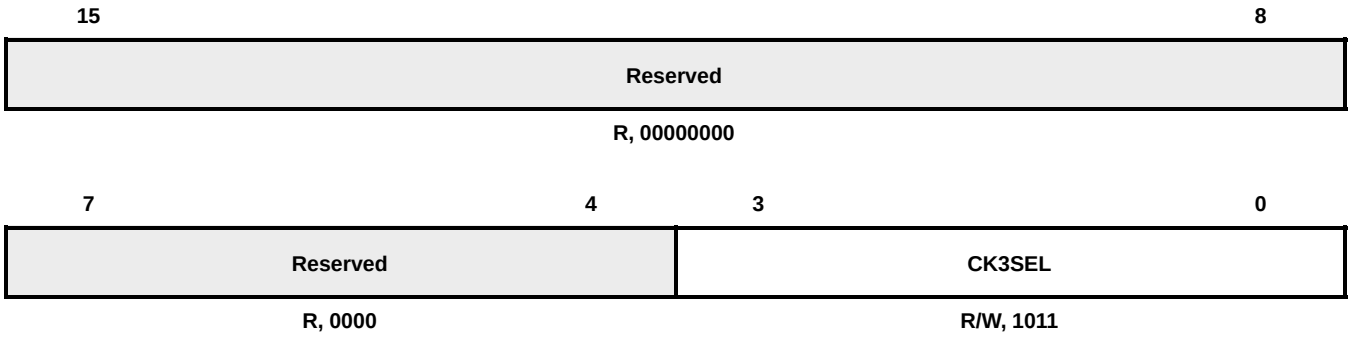
Figure 3–19. Oscillator Wakeup Control Register Layout (0x1C98)

Table 3–17. Oscillator Wakeup Control Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15:5	R	00000000000	Reserved. Reads return 0. Writes have no effect.
WKEN4	4	R/W	1	Input $\overline{\text{INT3}}$ can wake up the oscillator when the OSCPWRDN bit in PLLCSR is asserted to logic 1. WKEN4 = 0: Wake-up enabled. A low-to-high transition on $\overline{\text{INT3}}$ wakes up the oscillator and clears the OSCPWRDN bit. WKEN4 = 1: Wake-up disabled. A low-to-high transition on $\overline{\text{INT3}}$ does not wake up the oscillator.
WKEN3	3	R/W	1	Input $\overline{\text{INT2}}$ can wake up the oscillator when the OSCPWRDN bit in PLLCSR is asserted to logic 1. WKEN3 = 0: Wake-up enabled. A low-to-high transition on $\overline{\text{INT2}}$ wakes up the oscillator and clears the OSCPWRDN bit. WKEN3 = 1: Wake-up disabled. A low-to-high transition on $\overline{\text{INT2}}$ does not wake up the oscillator.
WKEN2	2	R/W	1	Input $\overline{\text{INT1}}$ can wake up the oscillator when the OSCPWRDN bit in PLLCSR is asserted to logic 1. WKEN2 = 0: Wake-up enabled. A low-to-high transition on $\overline{\text{INT1}}$ wakes up the oscillator and clears the OSCPWRDN bit. WKEN2 = 1: Wake-up disabled. A low-to-high transition on $\overline{\text{INT1}}$ does not wake up the oscillator.
WKEN1	1	R/W	1	Input $\overline{\text{INT0}}$ can wake up the oscillator when the OSCPWRDN bit in PLLCSR is asserted to logic 1. WKEN1 = 0: Wake-up enabled. A low-to-high transition on $\overline{\text{INT0}}$ wakes up the oscillator and clears the OSCPWRDN bit. WKEN1 = 1: Wake-up disabled. A low-to-high transition on $\overline{\text{INT0}}$ does not wake up the oscillator.
WKEN0	0	R/W	1	Input $\overline{\text{NMI}}$ can wake up the oscillator when the OSCPWRDN bit in PLLCSR is asserted to logic 1. WKEN0 = 0: Wake-up enabled. A low-to-high transition on $\overline{\text{NMI}}$ wakes up the oscillator and clears the OSCPWRDN bit. WKEN0 = 1: Wake-up disabled. A low-to-high transition on $\overline{\text{NMI}}$ does not wake up the oscillator.

3.9.5.9 CLKOUT3 Select Register (CK3SEL)

This register controls which clock is output onto the CLKOUT3 so that it may be used to test and debug the PLL (in addition to its normal function of being a direct input clock divider). Modes other than CK3SEL = 1011 are intended for debug use only and should not be used during normal operation.



LEGEND: R = Read, W = Write, *n* = value at reset

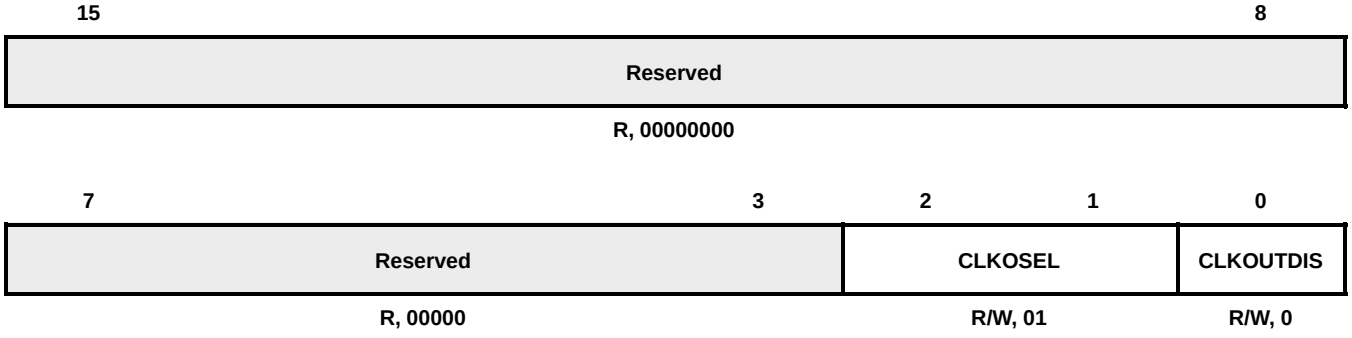
Figure 3–20. CLKOUT3 Select Register Layout (0x1C82)

Table 3–18. CLKOUT3 Select Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15:4	R	000000000000	Reserved. Reads return 0. Writes have no effect.
CK3SEL	3:0	R/W	1011	Output on CLK3SEL pin CK3SEL = 1001 CLKOUT3 becomes point A in Figure 3–11 CK3SEL = 1010 CLKOUT3 becomes point B in Figure 3–11 CK3SEL = 0000–0111 CLKOUT3 becomes oscillator divider output in Figure 3–11 CK3SEL = 1011 CLKOUT3 becomes point C in Figure 3–11 CK3SEL = Other Not supported

3.9.5.10 CLKOUT Selection Register (CLKOUTSR)

As described in Section 3.9.2, Clock Groups, the 5501 has different clock groups, each of which can be driven by a clock that is different from the CPU clock. The CLKOUT Selection Register determines which clock signal is reflected on the CLKOUT pin.



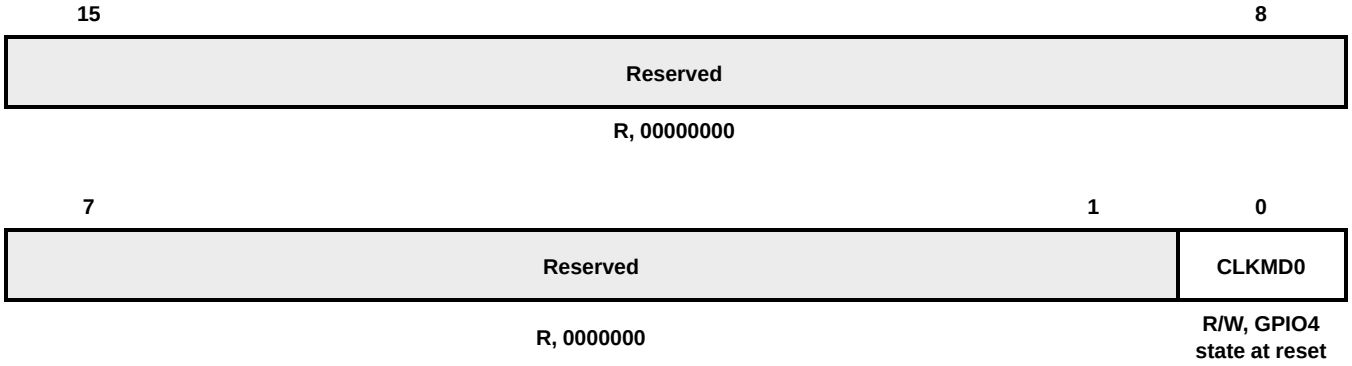
LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–21. CLKOUT Selection Register Layout (0x8400)

Table 3–19. CLKOUT Selection Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–3	R	0000000000000	Reserved
CLKOSEL	2:1	R/W	01	CLKOUT source-select CLKOSEL = 00: Reserved CLKOSEL = 01: CLKOUT source is SYSCLK1 CLKOSEL = 10: CLKOUT source is SYSCLK2 CLKOSEL = 11: CLKOUT source is SYSCLK3
CLKOUTDIS	0	R/W	0	Disable CLKOUT CLKOUTDIS = 0: CLKOUT enabled CLKOUTDIS = 1: CLKOUT disabled (driving 0)

3.9.5.11 Clock Mode Control Register (CLKMD)



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–22. Clock Mode Control Register Layout (0x8C00)

Table 3–20. Clock Mode Control Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–1	R	000000000000000	Reserved
CLKMD0	0	R/W	GPIO4 state at reset	Clock output source-select CLKMD0 = 0: OSCIN is selected as clock input source CLKMD0 = 1: X2/CLKIN is selected as clock input source

3.9.6 Reset Sequence

When reset is low, the clock generator is in bypass mode with the input clock set to OSCIN or X2/CLKIN, dependent upon the state of the GPIO4. After reset, the following conditions occur:

- GPIO6 is sampled on the rising edge of the reset signal.
- If GPIO6 is high at reset, the Parallel Port Mux will be configured to support the 16-bit EMIF and the Host Port Mux will be configured to support the HPI in 8-bit (multiplexed) mode. If GPIO6 is low at reset, both the Parallel Port Mux and the Host Port Mux will be configured to support parallel general-purpose I/O; the EMIF and HPI will be disabled in this mode. The Paralle/Host Port Mux Mode bit of the XBSR will reflect the mode selected for the Parallel and Host Port Muxes.†
- If GPIO4 is low, the CLKMD0 bit of the Clock Mode Control Register (CLKMD) is set to zero and the oscillator is enabled. Program flow begins after the reset signal has propagated throughout the device and the oscillator has stabilized. If GPIO4 is high, the CLKMD0 bit of the CLKMD register is set to one and the oscillator is set to power-down mode by setting the OSCPWRDN bit in the PLL Control/Status Register (PLLCSR) to one. Program flow begins after the reset signal has propagated throughout the device.

The 5501 has internal circuitry that will count down 70 reference clock cycles to allow reset signals to propagate correctly to all parts of the device after reset ($\overline{\text{RESET}}$ pin goes high). Furthermore, the 5501 also has internal circuitry that will count down 41,032 reference clock cycles to allow the oscillator input to become stable after waking up from power-down state or reset. If a reset is asserted, program flow will start after all stabilization periods have expired; this includes the oscillator stabilization period only if GPIO4 is low at reset. If the oscillator is coming out of power-down mode, program flow will start immediately after the oscillator stabilization period has completed. Table 3–21 summarizes the number of reference clock cycles needed before program flow begins.

Table 3–21. Number of Reference Clock Cycles Needed Until Program Flow Begins

CONDITION		REFERENCE CLOCK CYCLES
After Reset	Oscillator Not Used	70
	Oscillator Used	41,102
After Oscillator Power-Down		41,032

All output (O/Z) and input/output (I/O/Z) pins (except for CLKOUT, ECLKOUT2, and XF) will go into high-impedance mode during reset and will come out of high-impedance mode when the stabilization periods have expired. All output (O/Z) and input/output (I/O/Z) pins will retain their value when the device enters a power-down mode such as IDLE3 mode.

At power up, the reset pin must be kept at a low state until the 5501 has been completely powered up to prevent any unexpected behavior.

† Modifying the XBSR to change the mode of the Parallel Port Mux and Host Port Mux after the 5501 has been brought out of reset is *not* recommended.

3.10 Idle Control

The Idle function is implemented for low power consumption. The Idle function achieves low power consumption by gating the clock to unused parts of the chip, and/or setting the clock generator (PLL) and the internal oscillator to a power-down mode.

3.10.1 Clock Domains

The 5501 provides six clock domains to power-off the main clock to the portions of the device that are not being used. The six domains are:

- CPU Domain
- Master Port Domain (includes DMA and HPI modules)
- ICACHE
- Peripherals Domain
- Clock Generator Domain
- EMIF Domain

3.10.2 IDLE Procedures

Before entering idle mode (executing the IDLE instruction), the user has first to determine which part of the system needs to be disabled and then program the Idle Control Register (ICR) accordingly. When the IDLE instruction is executed, the ICR will be copied into the Idle Status Register (ISTR). The different bits of the ISTR register will be propagated to disable the chosen domains. Special care has to be taken in programming the ICR as some IDLE domain combinations are not valid (for example: CPU on and clock generator off).

3.10.2.1 CPU Domain Idle Procedure

The 5501 CPU can be idled by executing the following procedure.

1. Write ‘1’ to the CPUI bit (bit 0 of ICR).
2. Execute the IDLE instruction.
3. CPU will go to idle state

3.10.2.2 Master Port Domain (DMA/HPI) Idle Procedure

The clock to the DMA module and/or the HPI module will be stopped when the DMA and/or the HPI bit in the MICR is set to 1 and the MPIS bit in the ISTR becomes 1. The DMA will go into idle immediately if there is no data transfer taking place. If there is a data transfer taking place, then it will finish the current transfer and then go into idle. The HPI will go into idle regardless of whether or not there is a data transfer taking place. Software must confirm that the HPI has no activity before setting it to idle.

The 5501 DMA module and the HPI module can be disabled by executing the following procedure.

1. Write ‘1’ to the DMA bit and/or the HPI bit in MICR.
2. Write ‘1’ to the MPI bit in ICR.
3. Execute the IDLE instruction.
4. DMA and/or HPI go/goes to idle.

3.10.2.3 Peripheral Modules Idle Procedure

The clock to the modules included in the Peripherals Domain will be stopped when their corresponding bit in the PICR is set to 1 and the PERIS bit in the ISTR becomes 1. Each module in this domain will go into idle immediately if it has no activity. If the module being set to idle has activity, it will wait until the activity completes before going into idle.

Each peripheral module can be idled by executing the following procedure.

1. Write ‘1’ to the corresponding bit in PICR for each peripheral to be idled.
2. Write ‘1’ to the PERI bit in ICR.
3. Execute the IDLE instruction.
4. Every peripheral with its corresponding PICR bit set will go to idle.

3.10.2.4 EMIF Module Idle Procedure

The 5501 EMIF can be idled in one of two ways: through the ICR and through the PICR. The EMIF will go into idle immediately if there is no data transfer taking place within the DMA. If there is a data transfer taking place, then the EMIF will wait until the DMA finishes the current transfer and goes into idle before going into idle itself. Please note that while the EMIF is in idle, the SDRAM refresh function of the EMIF will not be available.

The 5501 EMIF can be idled through the ICR only when the following modules are set to idle: CPU, I-Port, ICACHE, DMA, and HPI. To place the EMIF in idle using the ICR, execute the following procedure:

1. Write ‘1’ to the DMA and HPI bits in MICR.
2. Write ‘1’ to the CPUI, MPI, ICACHEI, EMIFI, and IPORTI bits in ICR.
3. Execute the IDLE instruction.
4. EMIF and all modules listed in Step 2 will go to idle.

The 5501 EMIF can also be idled through the PICR. To place the EMIF in idle using the PICR, execute the following procedure:

1. Write a ‘1’ to the EMIF bit in PICR.
2. Write a ‘1’ to the PERI bit in ICR.
3. Execute IDLE instruction.
4. EMIF will go to IDLE.

3.10.2.5 IDLE2 Mode

In IDLE2 mode, all modules except the CLOCK module are set to idle state. To place the 5501 in IDLE2 mode, perform the following steps.

1. Write a ‘1’ to all peripheral module bits in the PICR.
2. Write a ‘1’ to the HPI and DMA bits in MICR.
3. Write a ‘1’ to all domain bits in the ICR except the CLOCK domain bit (CLKI).
4. Execute the IDLE instruction.
5. All internal clocks will be disabled, the CLOCK module will remain active.

3.10.2.6 IDLE3 Mode

In IDLE3 mode, all modules (including the CLOCK module) are set to idle state. To place the 5501 in IDLE3 mode, perform the following steps.

1. Clear (i.e., set to ‘0’) the PLLEN bit in PLLCSR to place the PLL in bypass mode.
2. Set the PLLPWRDN and PLLRST bits in PLLCSR to ‘1’.
3. Write a ‘1’ to all peripheral module bits in PICR.
4. Write a ‘1’ to the HPI and DMA bits in MICR.
5. Write a ‘1’ to all domain bits in ICR.
6. Execute the IDLE instruction.
7. PLL core is set to power-down mode and all internal clocks are disabled.

3.10.2.7 IDLE3 Mode With Internal Oscillator Disabled

In this state, all modules (including the CLOCK module) are set to the idle state and the internal oscillator is set to the power-down mode. This is the lowest power-consuming state that 5501 can be placed under.

1. Clear (i.e., set to ‘0’) the PLLEN bit in PLLCSR to place the PLL in bypass mode.
2. Set the PLLPWRDN, PLLRST, and OSCPWRDN bits in PLLCSR to ‘1’.
3. Set the WKEN register to specify which event will wake up internal oscillator [e.g., set bit 1 to have interrupt 0 ($\overline{\text{INT0}}$) wake up the oscillator].†
4. Write a ‘1’ to all peripheral module bits in the PICR.
5. Write a ‘1’ to the HPI and DMA bits in MICR.
6. Write a ‘1’ to all domain bits in the ICR.
7. Execute the IDLE instruction.
8. Internal oscillator is set to power-down mode, PLL core is set to power-down mode, and all internal clocks are disabled.

† External interrupt being used must be enabled through IER prior to setting the 5501 to IDLE.

3.10.3 Module Behavior at Entering IDLE State

All transactions must be completed before entering the IDLE state. Table 3–22 lists the behavior of each module before entering the IDLE state.

Table 3–22. Peripheral Behavior at Entering IDLE State

5501 IDLE DOMAIN	MODULES	MODULE BEHAVIOR AT ENTERING IDLE STATE (ASSUMING THE IDLE CONTROL IS SET)
CPU	CPU	Enter IDLE after CPU stops pipeline.
	Interrupt Controller	Enter IDLE after CPU stops.
	IDLE Controller	Enter IDLE after CPU stops.
	PLL Controller	Enter IDLE after CPU stops.
	Bus Selection Register	Enter IDLE after CPU stops.
	Timer Signal Selection Register	Enter IDLE after CPU stops.
	CLKOUT Selection Register	Enter IDLE after CPU stops.
	External Bus Control Register	Enter IDLE after CPU stops.
	Clock Mode Control Register	Enter IDLE after CPU stops.
Master Port	DMA	Enter IDLE state after current DMA transfer to internal memory, EMIF, or peripheral, or enter IDLE state immediately if no transfer exists. DMA has function of Auto-wakeup/Idle with McBSP data transfer during IDLE.
	HPI	Enter IDLE state immediately. Software has to take care of HPI activity.
ICACHE	ICACHE	Enter IDLE state after current data transfer from EMIF or program fetch from CPU finishes, or enter IDLE state immediately if no transfer and no access exist.
Peripheral	Timer0/1 and WDT	Enter IDLE state immediately
	DSP/BIOS Timer	Enter IDLE state immediately
	MCBSP0/1	Enter IDLE state after current McBSP activity is finished or enter IDLE state immediately if no activity exists. McBSP has function of Auto-wakeup/Idle with DMA data transfer during IDLE.
	GPIO	Enter IDLE state immediately.
	DIEID	Enter IDLE state immediately.
	I2C	Enter IDLE state after current I^2C activity is finished or enter IDLE state immediately if no activity exists.
	UART	Enter IDLE state after current UART activity is finished or enter IDLE state immediately if no activity exists.
	Parallel GPIO	Enter IDLE state immediately.
CLKGEN (PLL)	PLL divider	Enter IDLE state immediately.
	PLL core	Power-down state if set by software before IDLE
	Oscillator	Power-down state if set by software before IDLE
EMIF	EMIF	Enter IDLE mode after current DMA transfer or enter IDLE mode immediately if no activity exists.

3.10.4 Wake-Up Procedure

It is the user's responsibility to ensure that there exists a valid wake-up procedure before entering idle mode. Keep in mind that a hardware reset will restore all modules to their active state. All wake-up procedures are described in the next sections.

3.10.4.1 CPU Domain Wake-up Procedure

The CPU domain can be taken out of idle though an enabled external interrupt or an $\overline{\text{NMI}}$ signal. External interrupts can be enabled through the use of the IER0 and IER1 registers. Other modules, such as the EMIF module, will be taken out of idle automatically when the CPU wakes up. Please see the wake-up procedures for other modules for more information.

3.10.4.2 Master Port Domain (DMA/HPI) Wake-up Procedure

The 5501 DMA module and the HPI module can be taken out of idle simultaneously by executing the following procedure.

1. Write '0' to the MPI bit in ICR.
2. Execute the IDLE instruction.
3. DMA **and** HPI wake up.

It is also possible to wake up the DMA and HPI modules individually through the use of the Master Idle Control Register. To wake up only the DMA or the HPI module, perform the following steps:

1. Write '0' to the DMA bit **or** the HPI bit in MICR.
2. Selected module wakes up.

3.10.4.3 Peripheral Modules Wake-up Procedure

All 5501 peripherals can be taken out of idle simultaneously by executing the following procedure.

1. Write '0' to the PERI bit in ICR.
2. Execute the IDLE instruction.
3. All idled peripherals wake up.

It is also possible to wake up individual peripherals through the use of the Peripheral Idle Control Register by executing the following procedure.

1. Write '0' to the idle control bit of peripheral(s) in PICR.
2. Idled peripherals with '0' in PICR wake up.

3.10.4.4 EMIF Module Wake-up Procedure

If both the CPU and the EMIF are in idle, then the EMIF will come out of idle when the CPU is taken out of idle. The CPU can be taken out of idle through the use of an NMI or an enabled external interrupt. External interrupts can be enabled through the IER0 and IER1 registers.

If the CPU is not in idle, then the EMIF can be taken out of idle through either of the following two procedures:

1. Write ‘0’ to the PERI bit in ICR.
2. Execute the IDLE instruction.
3. All idled peripherals, including the EMIF, wake up.

Or:

1. Write ‘0’ to the EMIF bit in PICR.
2. The EMIF module will wake up.

3.10.4.5 IDLE2 Mode Wake-up Procedure

The 5501 can be taken completely out of IDLE2 mode by executing the following procedure.

1. CPU wakes up from idle through NMI or enabled external interrupt.
2. Write ‘0’ to all bits in the ICR.
3. Execute the IDLE instruction.
4. All internal clocks are enabled and all modules come out of idle.

3.10.4.6 IDLE3 Mode Wake-up Procedure

The 5501 can be taken completely out of IDLE3 mode by executing the following procedure.

1. CPU wakes up from idle through NMI or enabled external interrupt.
2. Write ‘0’ to all bits in the ICR.
3. Execute the IDLE instruction.
4. All internal clocks are enabled and all modules come out of idle.
5. Write ‘0’ to the PLLPWRDN and PLLRST bits in PLLCSR.
6. Wait for the PLL to relock by polling the LOCK bit or by setting up a LOCK interrupt.
7. Set the PLLEN bit in PLLCSR to ‘1’.
8. All internal clocks will now come from the PLL core.

   NOTE: Step 3 can be modified to only wake up certain modules, see previous sections for more information on the wake-up procedures for the 5501 modules.

3.10.4.7 IDLE3 Mode With Internal Oscillator Disabled Wake-up Procedure

The internal oscillator of the 5501 will be woken up along with the CLOCK module through an NMI or an enabled external interrupt. The source (INT0, INT1, INT2, INT3, or NMI) for the wake-up signal can be selected through the use of the WKEN register. The maskable external interrupts must be enabled through IER0 and IER1 prior to setting the 5501 to Idle 3 mode.

The 5501 has internal circuitry that will count down a predetermined number of clock cycles (41,032 reference clock cycles) to allow the oscillator input to become stable after waking up from power-down state or reset. When waking up from idle mode, program flow will start after the stabilization period of the oscillator has expired.

To take the 5501 (including the internal oscillator) out of the idle 3 state, execute the following procedure:

1. External interrupt or NMI occurs (as specified in the WKEN register) and program flow begins after 41,032 reference clock cycles.
2. CPU wakes up.
3. Write ‘0’ to all bits in the ICR.
4. Execute the IDLE instruction.
5. All internal clocks are enabled and all modules come out of idle.
6. Write ‘0’ to the PLLPWRDN, PLLRST, and OSCPWRDN bits in PLLCSR.
7. Wait for the PLL to relock by polling the LOCK bit or by setting up a LOCK interrupt.
8. Set the PLLEN bit in PLLCSR to ‘1’.
9. All internal clocks will now come from the PLL core.

NOTE: Step 2 can be modified to only wake up certain modules, see previous sections for more information on the wake-up procedures for the 5501 modules.

3.10.4.8 Summary of Wake-up Procedures

Table 3–23 summarizes the wake-up procedures.

Table 3–23. Wake-Up Procedures

ISTR VALUE	CLOCK DOMAIN STATUS	EXIT FROM IDLE	ICR AFTER WAKE-UP	ISTR AFTER WAKE-UP
xxx0xxx0	CPU – ON Clock Generator – ON Other – ON/OFF	1. DSP software modifies ICR and executes “IDLE” instruction 2. Reset	1. Modified value 2. All “0”	1. Updated to ICR modified value after “IDLE” instruction 2. All “0”
xxx0xxx1	CPU – OFF Clock Generator – ON Other – ON/OFF	1. Unmasked interrupt from external or on-chip module 2. Reset	1. Not modified 2. All “0”	1. CPUIS, CLKIS, and EMIFIS/XPORTIS/IPORTIS are set to “0” 2. All “0”
xxx11111	CPU – OFF Clock Generator – OFF Other – OFF	1. Unmasked interrupt from external 2. Reset	1. Not modified 2. All “0”	1. CPUIS, CLKIS, and EMIFIS/XPORTIS/IPORTIS are set to “0” 2. All “0”

PRODUCT PREVIEW

3.10.5 Auto-Wakeup/Idle Function for McBSP and DMA

The 5501 has an Auto-wakeup/Idle function for McBSP to DMA to on-chip memory data transfers when the DMA and the McBSP are both set to IDLE. In the case that the McBSP is set to external clock mode and the McBSP and the DMA are set to idle, the McBSP and the DMA can wake up from IDLE state automatically if the McBSP gets a new data transfer. The McBSP and the DMA enter the idle state automatically after data transfer is complete. [The clock generator (PLL) should be active and the PLL core should not be in power-down mode for the Auto-wakeup/Idle function to work.]

3.10.6 Clock State of Multiplexed Modules

The clock to the EMIF module is disabled automatically when this module is not selected through the External Bus Selection Register (XBSR). Note that any accesses to disabled modules will result in a bus error.

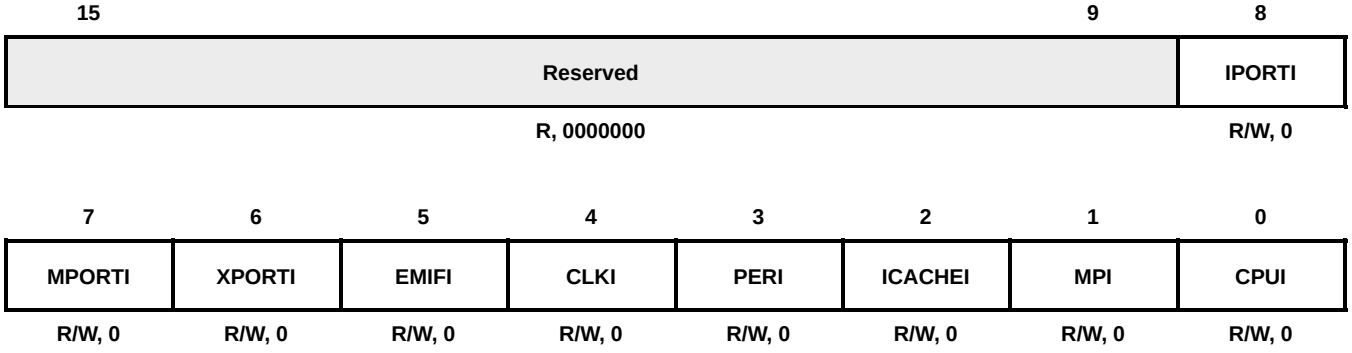
3.10.7 IDLE Control and Status Registers

The clock domains are controlled by the IDLE Configuration Register (ICR) that allows the user to place different parts of the device in Idle mode. The IDLE Status Register (ISTR) reflects the portion of the device that remains active. The peripheral domain is controlled by the Peripheral IDLE Control Register (PICR). The Peripheral IDLE Status Register (PISTR) reflects the portion of the peripherals that are in the IDLE state. The PLL Control/Status Register (PLLCSR) is used to power down the PLL core when the IDLE instruction is executed.

Table 3–24. Clock Domain Memory-Mapped Registers

ADDRESS	REGISTER NAME
0x0001	IDLE Configuration Register (ICR)
0x0002	IDLE Status Register (ISTR)
0x9400	Peripheral IDLE Control Register (PICR)
0x9401	Peripheral IDLE Status Register (PISTR)
0x9402	Master IDLE Control Register (MICR)
0x9403	Master IDLE Status Register (MISR)

3.10.7.1 IDLE Configuration Register (ICR)



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–23. IDLE Configuration Register Layout (0x0001)

Table 3–25. IDLE Configuration Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–9	R	0000000	Reserved
IPORTI	8	R/W	0	IPORT idle control bit**.** The IPORT is used for all ICACHE transactions. IPORTI = 0: IPORT remains active after execution of an IDLE instruction IPORTI = 1: IPORT is disabled after execution of an IDLE instruction
MPORTI	7	R/W	0	MPORT idle control bit. The MPORT is used for all DMA transactions. MPORTI = 0: MPORT remains active after execution of an IDLE instruction MPORTI = 1: MPORT is disabled after execution of an IDLE instruction
XPORTI	6	R/W	0	XPORT idle control bit. The XPORT is used for all I/O memory transactions. XPORTI = 0: XPORT remains active after execution of an IDLE instruction XPORTI = 1: XPORT is disabled after execution of an IDLE instruction
EMIFI	5	R/W	0	External Memory Interface (EMIF) idle control bit EMIFI = 0: EMIF module remains active after execution of an IDLE instruction EMIFI = 1: EMIF module is disabled after execution of an IDLE instruction
CLKI	4	R/W	0	Device clock generator idle control bit CLKI = 0: Device clock generator module remains active after execution of an IDLE instruction. CLKI = 1: Device clock generator is disabled after execution of an IDLE instruction. Disabling the clock generator provides the lowest level of power reduction by stopping the system clock. When CLKI is set to 1, the CPUI and DPI bits will be set to 1 in order to ensure a proper power-down mode.
PERI	3	R/W	0	Peripheral Idle control bit PERI = 0: All peripheral modules become/remain active after execution of an IDLE instruction PERI = 1: All peripheral modules with 1 in PICR are disabled after execution of an IDLE instruction
ICACHEI	2	R/W	0	ICACHE idle control bit ICACHEI = 0: ICACHE module remains active after execution of an IDLE instruction ICACHEI = 1: ICACHE module is disabled after execution of an IDLE instruction
MPI	1	R/W	0	Master peripheral (DMA and HPI) idle control bit MPI = 0: DMA and HPI modules remain active after execution of an IDLE instruction MPI = 1: DMA and HPI modules are disabled after execution of an IDLE instruction
CPUI	0	R/W	0	CPU idle control bit CPUI = 0: CPU module remains active after execution of an IDLE instruction CPUI = 1: CPU module is disabled after execution of an IDLE instruction

NOTE: For example, writing xxx000001b into the ICR does not indicate that the CPU domain is in IDLE mode; rather, it indicates that after the IDLE instruction, the CPU domain will be in IDLE mode.

3.10.7.2 IDLE Status Register (ISTR)

15						9	8
Reserved							IPORTIS
R, 0000000							R, 0

7	6	5	4	3	2	1	0
MPORTIS	XPORTIS	EMIFIS	CLKIS	PERIS	ICACHEIS	MPIS	CPUIS
R, 0	R, 0	R, 0	R, 0	R, 0	R, 0	R, 0	R, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–24. IDLE Status Register Layout (0x0002)

Table 3–26. IDLE Status Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–9	R	0000000	Reserved
IPORTIS	8	R	0	IPORT idle status bit. The IPORT is used for all ICACHE transactions. IPORTIS = 0: IPORT is active IPORTIS = 1: IPORT is disabled
MPORTIS	7	R	0	MPORT idle status bit. The MPORT is used for all DMA transactions. MPORTIS = 0: MPORT is active MPORTIS = 1: MPORT is disabled
XPORTIS	6	R	0	XPORT idle status bit. The XPORT is used for all I/O memory transactions. XPORTIS = 0: XPORT is active XPORTIS = 1: XPORT is disabled
EMIFIS	5	R	0	External Memory Interface (EMIF) idle status bit EMIFIS = 0: EMIF module is active EMIFIS = 1: EMIF module is disabled
CLKIS	4	R	0	Device clock generator idle status bit CLKIS = 0: Device clock generator module is active CLKIS = 1: Device clock generator is disabled
PERIS	3	R	0	Peripheral idle status bit PERIS = 0: All peripheral modules are active PERIS = 1: All peripheral modules are disabled
ICACHEIS	2	R	0	ICACHE idle status bit ICACHEIS = 0: ICACHE module is active ICACHEIS = 1: ICACHE module is disabled
MPIS	1	R	0	DMA and HPI idle status bit MPIS = 0: DMA and HPI modules are active MPIS = 1: DMA and HPI modules are disabled
CPUIS	0	R	0	CPU idle status bit CPUIS = 0: CPU module is active CPUIS = 1: CPU module is disabled

3.10.7.3 Peripheral IDLE Control Register (PICR)

15	14	13	12	11	10	9	8
Reserved		MISC	EMIF	BIOST	WDT	PIO	URT
R, 00		R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

7	6	5	4	3	2	1	0
I2C	ID	IO	Reserved	SP1	SP0	TIM1	TIM0
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–25. Peripheral IDLE Control Register Layout (0x9400)

Table 3–27. Peripheral IDLE Control Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–14	R	00	Reserved
MISC	13†	R/W	0	MISC bit MISC = 0: Miscellaneous modules remain active when ISTR.PERIS = 1 and IDLE instruction is executed. MISC = 1: MIscellaneous module is disabled when ISTR.PERIS = 1 and IDLE instruction is executed. Miscellaneous modules include the XBSR, TIMEOUT Error Register, XBCR, Timer Signal Selection Register, CLKOUT Select Register, and Clock Mode Control Register.
EMIF	12†	R/W	0	EMIF bit EMIF = 0: EMIF module remains active when ISTR.PERIS = 1 and IDLE instruction is executed. EMIF = 1: EMIF module is disabled when ISTR.PERIS = 1 and IDLE instruction is executed.
BIOST	11†	R/W	0	BIOS timer bit BIOST = 0: DSP/BIOS timer remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. BIOST = 1: DSP/BIOS timer is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
WDT	10†	R/W	0	Watchdog timer bit WDT = 0: WDT remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. WDT = 1: WDT is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
PIO	9†	R/W	0	Parallel GPIO timer bit PIO = 0: Parallel GPIO remains active when ISTR.PERIS = 1 (ISTR.[3]) and the IDLE instruction is executed. PIO = 1: Parallel GPIO is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.

† If the peripheral is already in IDLE, setting PERIS (bit 3 of ISTR) to 0 and executing the IDLE instruction will wake up all peripherals, and PICR bit settings will be ignored. If PERIS = 1, executing the IDLE instruction will wake up the peripheral if the PICR bit is 0.

PRODUCT PREVIEW

Table 3–27. Peripheral IDLE Control Register Bit Field Description (Continued)

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
URT	8†	R/W	0	UART timer bit URT = 0: UART remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. URT = 1: UART is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
I2C	7†	R/W	0	I2C timer bit I2C = 0: I^2C remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. I2C = 1: I^2C is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
ID	6†	R/W	0	ID timer bit ID = 0: ID remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. ID = 1: ID is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
IO	5†	R/W	0	IO timer bit IO = 0: GPIO remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. IO = 1: GPIO is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
Reserved	4	R/W	0	Reserved
SP1	3†	R/W	0	McBSP1 timer bit SP1 = 0: McBSP1 remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. SP1 = 1: McBSP1 is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
SP0	2†	R/W	0	McBSP0 timer bit SP0 = 0: McBSP0 remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. SP0 = 1: McBSP0 is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
TIM1	1†	R/W	0	TIMER1 timer bit TIM1 = 0: TIMER1 remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. TIM1 = 1: TIMER1 is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.
TIM0	0†	R/W	0	TIMER0 timer bit TIM0 = 0: TIMER0 remains active when ISTR.PERIS = 1 and the IDLE instruction is executed. TIM0 = 1: TIMER0 is disabled when ISTR.PERIS = 1 and the IDLE instruction is executed.

† If the peripheral is already in IDLE, setting PERIS (bit 3 of ISTR) to 0 and executing the IDLE instruction will wake up all peripherals, and PICR bit settings will be ignored. If PERIS = 1, executing the IDLE instruction will wake up the peripheral if the PICR bit is 0.

3.10.7.4 Peripheral IDLE Status Register (PISTR)

15	14	13	12	11	10	9	8
Reserved		MISC	EMIF	BIOST	WDT	PIO	URT
R, 00		R, 0	R, 0	R, 0	R, 0	R, 0	R, 0

7	6	5	4	3	2	1	0
I2C	ID	IO	Reserved	SP1	SP0	TIM1	TIM0
R, 0	R, 0	R, 0	R, 0	R, 0	R, 0	R, 0	R, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–26. Peripheral IDLE Status Register Layout (0x9401)

Table 3–28. Peripheral IDLE Status Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–14	R	00	Reserved
MISC	13	R	0	MISC bit MISC = 0: Miscellaneous modules are active MISC = 1: Miscellaneous modules are disabled Miscellaneous modules include the XBSR, TIMEOUT Error Register, XBCR, Timer Signal Selection Register, CLKOUT Select Register, and Clock Mode Control Register.
EMIF	12	R	0	EMIF bit EMIF = 0: EMIF module is active EMIF = 1: EMIF module is disabled
BIOST	11	R	0	BIOS timer bit BIOST = 0: DSP/BIOS timer is active BIOST = 1: DSP/BIOS timer is disabled
WDT	10	R	0	Watchdog timer bit WDT = 0: WDT is active WDT = 1: WDT is disabled
PIO	9	R	0	Parallel GPIO timer bit PIO = 0: Parallel GPIO is active PIO = 1: Parallel GPIO is disabled
URT	8	R	0	UART timer bit URT = 0: UART is active URT = 1: UART is disabled
I2C	7	R	0	I2C timer bit I2C = 0: I^2C is active I2C = 1: I^2C is disabled
ID	6	R	0	ID timer bit ID = 0: ID is active ID = 1: ID is disabled
IO	5	R	0	IO timer bit IO = 0: GPIO is active IO = 1: GPIO is disabled

PRODUCT PREVIEW

Table 3–28. Peripheral IDLE Status Register Bit Field Description (Continued)

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	4	R	0	Reserved
SP1	3	R	0	McBSP1 timer bit SP1 = 0: McBSP1 is active SP1 = 1: McBSP1 is disabled
SP0	2	R	0	McBSP0 timer bit SP0 = 0: McBSP0 is active SP0 = 1: McBSP0 is disabled
TIM1	1	R	0	TIMER1 timer bit TIM1 = 0: TIMER1 is active TIM1 = 1: TIMER1 is disabled
TIM0	0	R	0	TIMER0 timer bit TIM0 = 0: TIMER0 is active TIM0 = 1: TIMER0 is disabled

3.10.7.5 Master IDLE Control Register (MICR)

15	8
Reserved	
R, 00000000	

7	2	1	0
Reserved		HPI	DMA
R, 000000		R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–27. Master IDLE Control Register Layout (0x9402)

Table 3–29. Master IDLE Control Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–2	R	00000000000000	Reserved
HPI	1	R/W	0	HPI bit HPI = 0: HPI remains active when ISTR.MPIS becomes 1 HPI = 1: HPI is disabled when ISTR.MPIS becomes 1
DMA	0	R/W	0	DMA bit DMA = 0: DMA remains active when ISTR.MPIS becomes 1 DMA = 1: DMA is disabled when ISTR.MPIS becomes 1

3.10.7.6 Master IDLE Status Register (MISR)

15	8
Reserved	
R, 00000000	

7	2	1	0
Reserved		HPI	DMA
R, 000000		R, 0	R, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–28. Master IDLE Status Register Layout (0x9403)

Table 3–30. Master IDLE Status Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–2	R	00000000000000	Reserved
HPI	1	R	0	HPI bit HPI = 0: HPI is active HPI = 1: HPI is in IDLE status
DMA	0	R	0	DMA bit DMA = 0: DMA is active DMA = 1: DMA is in IDLE status

3.11 General-Purpose I/O (GPIO)

The 5501 includes an 8-bit I/O port solely for general-purpose inputs and outputs. Several dual-purpose (multiplexed) pins complement the dedicated GPIO pins. The following sections describe the 8-bit GPIO port as well as the dual GPIO functions of the Parallel Port Mux and Host Port Mux pins.

3.11.1 General-Purpose I/O Port

The general-purpose I/O port consists of eight individually bit-selectable I/O pins GPIO0 (LSB) through GPIO7 (MSB). The I/O port is controlled using two registers—IODIR and IODATA—that can be accessed by the CPU or by the DMA, via the peripheral bus controller. The General-Purpose I/O Direction Register (IODIR) is mapped at address 0x3400, and the General-Purpose I/O Data Register (IODATA) is mapped at address 0x3401.

Figure 3–29 and Figure 3–30 show the bit layout of IODIR and IODATA, respectively. Table 3–31 and Table 3–32 describe the bit fields of these registers.

3.11.1.1 General-Purpose I/O Direction Register (IODIR)

15							8
Reserved							
R, 00000000							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IO7DIR	IO6DIR	IO5DIR	IO4DIR	IO3DIR	IO2DIR	IO1DIR	IO0DIR
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–29. GPIO Direction Register Layout (0x3400)

Table 3–31. GPIO Direction Register Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–8	R	00000000	Reserved
IOxDIR	7–0	R/W	00000000	Data direction bits that configure the GPIO pins as inputs or outputs. IOxDIR = 0: Configure corresponding GPIO pin as an input IOxDIR = 1: Configure corresponding GPIO pin as an output

† x = value from 0 to 7

3.11.1.2 General-Purpose I/O Data Register (IODATA)

15							8
Reserved							
R, 00000000							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IO7D	IO6D	IO5D	IO4D	IO3D	IO2D	IO1D	IO0D
R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin

LEGEND: R = Read, W = Write, *n* = value at reset, *pin* = the reset value depends on the signal level on the corresponding I/O pin.

Figure 3–30. GPIO Data Register Layout (0x3401)

Table 3–32. GPIO Data Register Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–8	R	00000000	Reserved
IOxD	7–0	R/W	Depends on the signal level on the corresponding I/O pin	Data bits that are used to control the level of the I/O pins configured as outputs and to monitor the level of the I/O pins configured as inputs. If IOxDIR = 0, then: IOxD = 0: Corresponding GPIO pin is read as a low IOxD = 1: Corresponding GPIO pin is read as a high If IOxDIR = 1, then: IOxD = 0: Set corresponding GPIO pin to low IOxD = 1: Set corresponding GPIO pin to high

† x = value from 0 to 7

3.11.2 Parallel Port General-Purpose I/O (PGPIO)

The 4 address signals (EMIF.A[21:18]), 8 host data signals (HD[7:0]), 2 HPI control signals (HC1–HC0), and 16 control signals (C[15:0]) can be individually enabled as PGPIO when the Parallel Port Mux Mode bit field of the External Bus Selection Register (XBSR) is cleared for PGPIO mode. (See Table 3–33.) Sixteen pins, PGPIO[19:4], are dedicated to parallel general-purpose I/O. These pins are controlled by three set of registers: the PGPIO enable registers, the PGPIO direction registers, and the PGPIO data registers.

- The PGPIO enable registers PGPIOEN0–PGPIOEN2 (see Figure 3–31, Figure 3–34, and Figure 3–37) determine if the pin serves as PGPIO or if it is placed in high-impedance state.
- The PGPIO direction registers PGPIODIR0–PGPIODIR2 (see Figure 3–32, Figure 3–35, and Figure 3–38) determine if the pin is an input or output.
- The PGPIO data registers PGPIODAT0–PGPIODAT2 (see Figure 3–33, Figure 3–36, and Figure 3–39) store the value read or written externally.

NOTE: The enable registers PGPIOENn cannot override the External Bus Selection Register (XBSR) setting.

Table 3–33. TMS320VC5501 PGPIO Cross-Reference

PIN	PARALLEL PORT MUX MODE = 0 (PGPIO)	PARALLEL PORT MUX MODE = 1 (FULL EMIF)
EMIF Address Bus		
A[21:18]	PGPIO[3:0]	EMIF.A[21:18]
EMIF Data Bus		
PGPIO[19:4]	PGPIO[19:4]	Reserved
EMIF Control Bus		
C0	PGPIO20	EMIF.ARE/SADS/SDCAS/SRE
C1	PGPIO21	EMIF.AOE/SOE/SDRAS
C2	PGPIO22	EMIF.AWE/SWE/SDWE
C3	PGPIO23	EMIF.ARDY
C4	PGPIO24	EMIF.CE0
C5	PGPIO25	EMIF.CE1
C6	PGPIO26	EMIF.CE2
C7	PGPIO27	EMIF.CE3
C8	PGPIO28	EMIF.BE0
C9	PGPIO29	EMIF.BE1
C10	PGPIO30	EMIF.BE2
C11	PGPIO31	EMIF.BE3
C12	PGPIO32	EMIF.SDCKE
C13	PGPIO33	EMIF.SOE3
C14	PGPIO34	EMIF.HOLD
C15	PGPIO35	EMIF.HOLDA
HPI Data Bus		
HD[7:0]	PGPIO[43:36]	HPI.HD[7:0]
HPI Control Bus		
HC0	PGPIO44	HPI.HAS
HC1	PGPIO45	HPI.HBIL

3.11.2.1 Parallel GPIO Enable Register 0 (PGPIOEN0)

15	14	13	12	11	10	9	8
IO15EN	IO14EN	IO13EN	IO12EN	IO11EN	IO10EN	IO9EN	IO8EN
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0
7	6	5	4	3	2	1	0
IO7EN	IO6EN	IO5EN	IO4EN	IO3EN	IO2EN	IO1EN	IO0EN
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–31. Parallel GPIO Enable Register 0 Layout (0x4400)

Table 3–34. Parallel GPIO Enable Register 0 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
IOxEN	15–0	R/W	0000000000000000	Enable or disable GPIO function of the EMIF address bus EMIF.A[21:18] and the EMIF data bus EMIF.D[27:16]. See Table 3–4. IOxEN = 0: GPIO function of corresponding signal is disabled, i.e., the pin goes into a high-impedance state. IOxEN = 1: GPIO function of corresponding signal is enabled, i.e., the signal supports its GPIO function. Bits [3:0] correspond to EMIF.A[21:18]

† x = value from 0 to 15

3.11.2.2 Parallel GPIO Direction Register 0 (PGPIODIR0)

15	14	13	12	11	10	9	8
IO15DIR	IO14DIR	IO13DIR	IO12DIR	IO11DIR	IO10DIR	IO9DIR	IO8DIR
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0
7	6	5	4	3	2	1	0
IO7DIR	IO6DIR	IO5DIR	IO4DIR	IO3DIR	IO2DIR	IO1DIR	IO0DIR
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–32. Parallel GPIO Direction Register 0 Layout (0x4401)

Table 3–35. Parallel GPIO Direction Register 0 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
IOxDIR	15–0	R/W	0000000000000000	Data direction bits that configure corresponding I/O pins either as inputs or outputs. See Table 3–4. IOxDIR = 0: Configure corresponding pin as an input. IOxDIR = 1: Configure corresponding pin as an output. Bits [3:0] correspond to EMIF.A[21:18]

† x = value from 0 to 15

3.11.2.3 Parallel GPIO Data Register 0 (PGPIODAT0)

15	14	13	12	11	10	9	8
IO15DAT	IO14DAT	IO13DAT	IO12DAT	IO11DAT	IO10DAT	IO9DAT	IO8DAT
R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin
7	6	5	4	3	2	1	0
IO7DAT	IO6DAT	IO5DAT	IO4DAT	IO3DAT	IO2DAT	IO1DAT	IO0DAT
R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin

LEGEND: R = Read, W = Write, *n* = value at reset, *pin* = the reset value depends on the signal level on the corresponding I/O pin.

Figure 3–33. Parallel GPIO Data Register 0 Layout (0x4402)

Table 3–36. Parallel GPIO Data Register 0 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
IOxDAT	15–0	R/W	Depends on the signal level on the corresponding I/O pin	Data bits that are used to control the level of the corresponding I/O pins configured as output pins and to monitor the level of the corresponding I/O pins configured as input pins. See Table 3–4. If IOxDIR = 0, then: IOxDAT = 0: Corresponding I/O pin is read as a low IOxDAT = 1: Corresponding I/O pin is read as a high If IOxDIR = 1, then: IOxDAT = 0: Set corresponding I/O pin to low IOxDAT = 1: Set corresponding I/O pin to high Bits [3:0] correspond to EMIF.A[21:18]

† x = value from 0 to 15

PRODUCT PREVIEW

3.11.2.4 Parallel GPIO Enable Register 1 (PGPIOEN1)

15	14	13	12	11	10	9	8
IO31EN	IO30EN	IO29EN	IO28EN	IO27EN	IO26EN	IO25EN	IO24EN
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0
7	6	5	4	3	2	1	0
IO23EN	IO22EN	IO21EN	IO20EN	IO19EN	IO18EN	IO17EN	IO16EN
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–34. Parallel GPIO Enable Register 1 Layout (0x4403)

Table 3–37. Parallel GPIO Enable Register 1 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
IOxEN	15–0	R/W	0000000000000000	Enable or disable GPIO function of the EMIF data bus EMIF.D[31:28] and the EMIF control signals C[11:0]. See Table 3–4. IOxEN = 0: GPIO function of corresponding signal is disabled, i.e., the pin goes into a high-impedance state. IOxEN = 1: GPIO function of corresponding signal is enabled, i.e., the signal supports its GPIO function. Bits [15:4] correspond to C[11:0]

† x = value from 16 to 31

3.11.2.5 Parallel GPIO Direction Register 1 (PGPIODIR1)

15	14	13	12	11	10	9	8
IO31DIR	IO30DIR	IO29DIR	IO28DIR	IO27DIR	IO26DIR	IO25DIR	IO24DIR
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0
7	6	5	4	3	2	1	0
IO23DIR	IO22DIR	IO21DIR	IO20DIR	IO19DIR	IO18DIR	IO17DIR	IO16DIR
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–35. Parallel GPIO Direction Register 1 Layout (0x4404)

Table 3–38. Parallel GPIO Direction Register 1 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
IOxDIR	15–0	R/W	0000000000000000	Data direction bits that configure corresponding I/O pins either as inputs or outputs. See Table 3–4. IOxDIR = 0: Configure corresponding pin as an input. IOxDIR = 1: Configure corresponding pin as an output. Bits [15:4] correspond to C[11:0]

† x = value from 16 to 31

3.11.2.6 Parallel GPIO Data Register 1 (PGPIODAT1)

15	14	13	12	11	10	9	8
IO31DAT	IO30DAT	IO29DAT	IO28DAT	IO27DAT	IO26DAT	IO25DAT	IO24DAT
R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin
7	6	5	4	3	2	1	0
IO23DAT	IO22DAT	IO21DAT	IO20DAT	IO19DAT	IO18DAT	IO17DAT	IO16DAT
R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin

LEGEND: R = Read, W = Write, *n* = value at reset, *pin* = the reset value depends on the signal level on the corresponding I/O pin.

Figure 3–36. Parallel GPIO Data Register 1 Layout (0x4405)

Table 3–39. Parallel GPIO Data Register 1 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
IOxDAT	15–0	R/W	Depends on the signal level on the corresponding I/O pin	Data bits used to control the level of the corresponding I/O pins configured as output pins and to monitor the level of the corresponding I/O pins configured as input pins. See Table 3–4. If IOxDIR = 0, then: IOxDAT = 0: Corresponding I/O pin is read as a low IOxDAT = 1: Corresponding I/O pin is read as a high If IOxDIR = 1, then: IOxDAT = 0: Set corresponding I/O pin to low IOxDAT = 1: Set corresponding I/O pin to high Bits [15:4] correspond to C[11:0]

† x = value from 16 to 31

PRODUCT PREVIEW

3.11.2.7 Parallel GPIO Enable Register 2 (PGPIOEN2)

15	14	13	12	11	10	9	8
Reserved		IO45EN	IO44EN	IO43EN	IO42EN	IO41EN	IO40EN
R/W, 00		R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IO39EN	IO38EN	IO37EN	IO36EN	IO35EN	IO34EN	IO33EN	IO32EN
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–37. Parallel GPIO Enable Register 2 Layout (0x4406)

Table 3–40. Parallel GPIO Enable Register 2 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–14	R/W	00	Reserved
IOxEN	13–0	R/W	00000000000000	Enable or disable GPIO function of the EMIF control signals C[15:12], the HPI data pins HD[7:0], and the HPI control signals HC[1:0]. See Table 3–4 and Table 3–5. IOxEN = 0: GPIO function of corresponding signal is disabled, i.e., the pin goes into a high-impedance state. IOxEN = 1: GPIO function of corresponding signal is enabled, i.e., the signal supports its GPIO function. Bits [13:12] correspond to HC[1:0] Bits [11:4] correspond to HD[7:0] Bits [3:0] correspond to C[15:12]

† x = value from 32 to 45

3.11.2.8 Parallel GPIO Direction Register 2 (PGPIODIR2)

15	14	13	12	11	10	9	8
Reserved		IO45DIR	IO44DIR	IO43DIR	IO42DIR	IO41DIR	IO40DIR
R/W, 00		R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IO39DIR	IO38DIR	IO37DIR	IO36DIR	IO35DIR	IO34DIR	IO33DIR	IO32DIR
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–38. Parallel GPIO Direction Register 2 Layout (0x4407)

Table 3–41. Parallel GPIO Direction Register 2 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–14	R/W	00	Reserved
IOxDIR	13–0	R/W	00000000000000	Data direction bits that configure corresponding I/O pins either as inputs or outputs. See Table 3–4 and Table 3–5. IOxDIR = 0: Configure corresponding pin as an input. IOxDIR = 1: Configure corresponding pin as an output. Bits [13:12] correspond to HC[1:0] Bits [11:4] correspond to HD[7:0] Bits [3:0] correspond to C[15:12]

† x = value from 32 to 45

3.11.2.9 Parallel GPIO Data Register 2 (PGPIODAT2)

15	14	13	12	11	10	9	8
Reserved		IO45DAT	IO44DAT	IO43DAT	IO42DAT	IO41DAT	IO40DAT
R/W, 00		R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IO39DAT	IO38DAT	IO37DAT	IO36DAT	IO35DAT	IO34DAT	IO33DAT	IO32DAT
R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin	R/W, pin

LEGEND: R = Read, W = Write, *n* = value at reset, *pin* = the reset value depends on the signal level on the corresponding I/O pin.

Figure 3–39. Parallel GPIO Data Register 2 Layout (0x4408)

Table 3–42. Parallel GPIO Data Register 2 Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–14	R/W	00	Reserved
IOxDAT	13–0	R/W	Depends on the signal level on the corresponding I/O pin	Data bits used to control the level of the I/O pins configured as output pins, and to monitor the level of the corresponding I/O pins configured as input pins. See Table 3–4 and Table 3–5. If IOxDIR = 0, then: IOxDAT = 0: Corresponding I/O pin is read as a low IOxDAT = 1: Corresponding I/O pin is read as a high If IOxDIR = 1, then: IOxDAT = 0: Set corresponding I/O pin to low IOxDAT = 1: Set corresponding I/O pin to high Bits [13:12] correspond to HC[1:0] Bits [11:4] correspond to HD[7:0] Bits [3:0] correspond to C[15:12]

† x = value from 32 to 45

3.12 External Bus Control Register

The External Bus Control Register is used to disable/enable the bus pullups, pulldowns, and bus holders of the 5501 pins. Table 3–43 lists which 5501 pins have pullups, pulldowns, and bus holders and which bit on the XBCR enables/disables that feature. Note: for pins with dual functionality (e.g., HC0, HC1, C0, etc.), the bus holder, pullup, and pulldown feature of each pin can be enabled or disabled regardless of the function of the pin at the time.

Table 3–43. Pins With Pullups, Pulldowns, and Bus Holders

XBCR CONTROL BIT	PIN	FEATURE
EMU	TCK	Pullup
	TDI	Pullup
	TMS	Pullup
	$\overline{\text{TRST}}$	Pulldown
TEST	EMU0	Pullup
	EMU1/$\overline{\text{OFF}}$	Pullup
WDT	$\overline{\text{NMI}}$/WDTOUT	Pullup
HC	HC0	Pullup
	HC1	Pulldown
HD	HD[7:0]	Bus Holder
PC	C0	Bus Holder
	C1	Bus Holder
	C2	Bus Holder
	C3	Pullup
	C4	Bus Holder
	C5	Bus Holder
	C6	Bus Holder
	C7	Bus Holder
	C8	Bus Holder
	C9	Bus Holder
	C10	Bus Holder
	C11	Bus Holder
	C12	Bus Holder
	C13	Bus Holder
	C14	Pullup
	C15	Bus Holder
PD	D[15:0]	Bus Holder
	PGPIO[19:4]	Bus Holder
PA	A[21:2]	Bus Holder

3.12.1 External Bus Control Register (XBCR)

15							8
Reserved							
R, 00000000							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EMU	**TEST**	**WDT**	**HC**	**HD**	**PC**	**PD**	**PA**
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–40. External Bus Control Register Layout (0x6C01)

Table 3–44. External Bus Control Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–8	R	00000000	Reserved
EMU	7	R/W	0	EMU bit EMU = 0: Pullups on EMU1 and EMU0 pins are enabled. EMU = 1: Pullups on EMU1 and EMU0 pins are disabled.
TEST	6	R/W	0	TEST bit TEST = 0: Pullups/pulldowns on test pins are enabled (does not include EMU1 and EMU0 pins) TEST = 1: Pullups/pulldowns on test pins are disabled (does not include EMU1 and EMU0 pins)
WDT	5	R/W	0	WDT bit WDT = 0: Pullup on $\overline{\text{NMI}}$/WDTOUT pin is enabled WDT = 1: Pullup on $\overline{\text{NMI}}$/WDTOUT pin is disabled
HC	4	R/W	0	HPI control signal bit HC = 0: Pullups/pulldowns on HPI control pins (HC0 and HC1) are enabled HC = 1: Pullups/pulldowns on HPI control pins (HC0 and HC1) are disabled
HD	3	R/W	0	HPI data bus bit HD = 0: Bus holders on HPI data bus (pins HD[7:0]) are enabled HD = 1: Bus holders on HPI data bus (pins HD[7:0]) are disabled
PC	2	R/W	0	EMIF control signals PC = 0: Bus holders and pullups on EMIF control pins are enabled PC = 1: Bus holders and pullups on EMIF control pins are disabled
PD	1	R/W	0	EMIF data bus signals PD = 0: Bus holders on EMIF data bus (pins D[15:0]) and parallel general-purpose I/O pins (PGPIO[19:4]) are enabled PD = 1: Bus holders on EMIF data bus (pins D[15:0]) and parallel general-purpose I/O pins (PGPIO[19:4]) are disabled
PA	0	R/W	0	EMIF address bus signals PA = 0: Bus holders on EMIF address bus (pins A[21:2]) are enabled PA = 1: Bus holders on EMIF address bus (pins A[21:2]) are disabled

3.13 Internal Ports and System Registers

The 5501 includes three internal ports that interface the CPU core with the peripheral modules. Although these ports cannot be directly controlled by user code, the registers associated with each port can be used to monitor a number of error conditions that could be generated through illegal operation of the 5501. The port registers are described in the following sections.

The 5501 also includes two registers that can be used to monitor and control several aspects of the interface between the CPU and the system-level peripherals, these registers are also described in the following sections.

3.13.1 XPORT Interface

The XPORT interfaces the CPU core to all peripheral modules. The XPORT will generate bus errors for invalid accesses to any registers that fall under the ranges shown in Table 3–45. The INTERREN bit of the XPORT Configuration Register (XCR) controls the bus error feature of the XPORT. The INTERR bit of the XPORT Bus Error Register (XERR) is set to “1” when an error occurs during an access to a register listed in Table 3–45. The EBUS and DBUS bits can be used to distinguish whether the error occurred during a write or read access.

Table 3–45. I/O Addresses Under Scope of XPORT

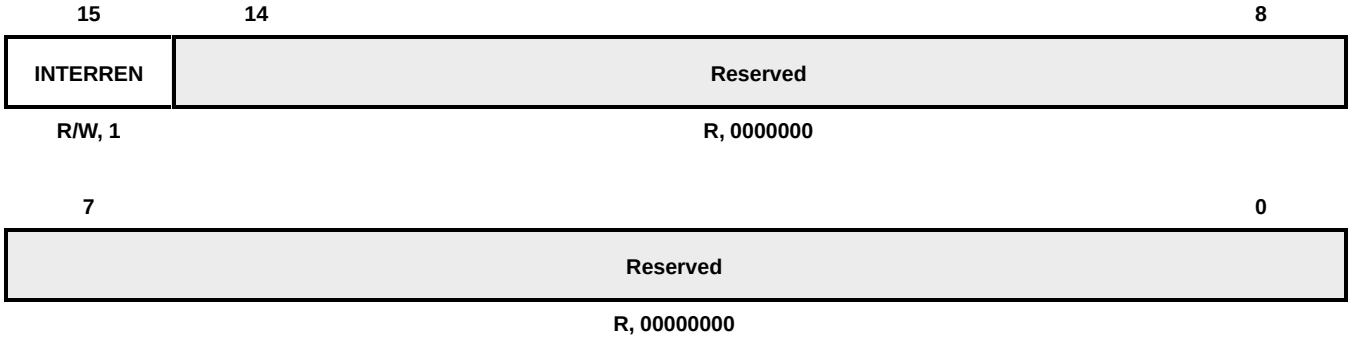
I/O ADDRESS RANGE
0x0000–0x03FF
0x1400–0x17FF
0x2000–0x23FF

The PERITO bit of the XERR is used to indicate that a CPU, DMA, or HPI access to a disabled/idled peripheral module has generated a time-out error. The time-out error feature is enabled through the PERITOEN bit of the Time-Out Control Register (TOCR). A time-out error is generated when 512 clock cycles pass without a response from the peripheral register.

The XPORT can be placed into idle by setting the XPORTI bit of the Idle Control Register (ICR) and executing the IDLE instruction. When the XPORT is in idle, it will stop accepting new peripheral module requests and it will also not check for internal I/O bus errors. If there is a request from the CPU core or a peripheral module, the XPORT will not respond and hang. The ICR register will generate a bus error if the XPORT is idled without the CPU or Master Port domains being in idle mode.

3.13.1.1 XPORT Configuration Register (XCR)

The XPORT Configuration Register bit layout is shown in Figure 3–41 and the bits are described in Table 3–46.



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–41. XPORT Configuration Register Layout (0x0100)

Table 3–46. XPORT Configuration Register Bit Field Description

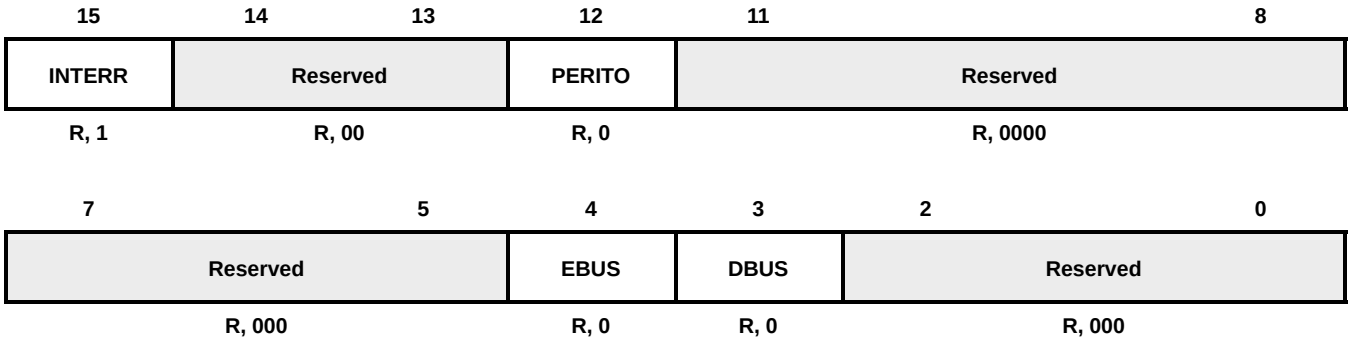
BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
INTERREN	15	R/W	1	INTERREN bit INTERREN = 0: The XPORT will not generate a bus error for invalid accesses to registers listed in Table 3–45. Note that any invalid accesses to these registers will hang the pipeline. INTERREN = 1: The XPORT will generate a bus error for invalid accesses to registers listed in Table 3–45.† Note that when a bus error occurs, any data returned by the read instruction will not be valid.
Reserved	14–0	R	000000000000000	Reserved

† This feature will not work if the XPORT is placed in idle through the ICR. However, a bus error will be generated if the XPORT is placed in idle without the CPU being in idle.

PRODUCT PREVIEW

3.13.1.2 XPORT Bus Error Register (XERR)

The XPORT Bus Error Register bit layout is shown in Figure 3–42 and the bits are described in Table 3–47.



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–42. XPORT Bus Error Register Layout (0x0102)

Table 3–47. XPORT Bus Error Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
INTERR	15	R	1	INTERR bit INTERR = 0: No error INTERR = 1: An error occurred during an access to one of the registers listed in Table 3–45.
Reserved	14–13	R	00	Reserved
PERITO	12	R	0	PERITO bit PERITO = 0: No error PERITO = 1: A time-out error occurred during an access to a peripheral register.
Reserved	11–5	R	0000000	Reserved
EBUS	4	R	0	EBUS error bit† EBUS = 0: No error EBUS = 1: An error occurred during an EBUS access (write) to one of the registers listed in Table 3–45.
DBUS	3	R	0	DBUS error bit† DBUS = 0: No error DBUS = 1: An error occurred during a DBUS access (read) to one of the registers listed in Table 3–45.
Reserved	2–0	R	000	Reserved

† See the *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371) for more information on the D-bus and E-bus.

3.13.2 DPORT Interface

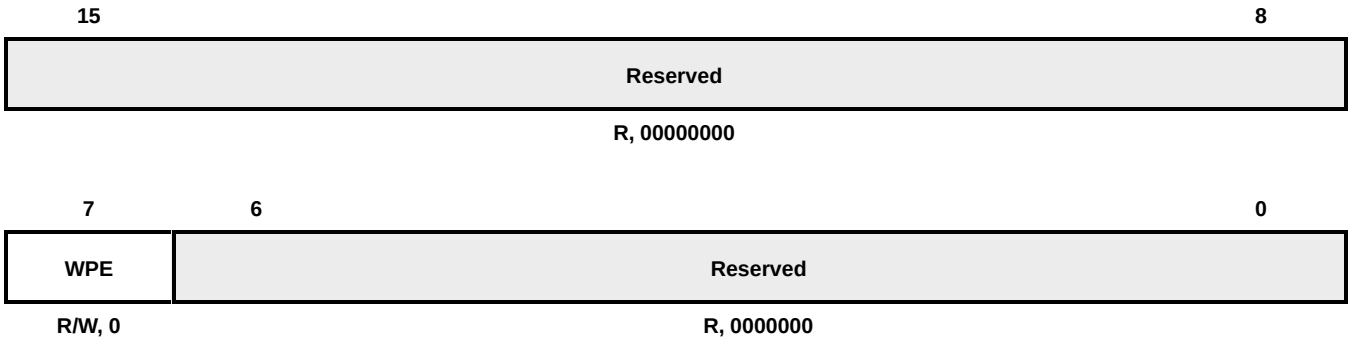
The DPORT interfaces the CPU to the EMIF module. The DPORT is capable of enabling write posting on the EMIF module. Write posting prevents stalls to the CPU during external memory writes. Two write posting registers, which are freely associated with E and F bus writes, exist within the DPORT and are used to store the write address and data so that writes can be zero wait state for the CPU. External memory writes will not generate stalls to the CPU unless the two write posting registers are filled. Write posting is enabled by setting the WPE bit of the DCR to 1.

The EMIFTO bit of the DERR is used to indicate that a CPU, DMA, HPI, or IPORT access to external memory has generated a time-out error. The time-out error feature is enabled through the EMIFTOEN bit of the Time-Out Control Register (TOCR). This function is not recommended during normal operation of the 5501.

The DPORT can be placed into idle through the EMIFI bit of the Idle Control Register (ICR) and executing the IDLE instruction. When the DPORT is in idle, it will stop accepting new EMIF requests. If there is a request from the CPU or the EMIF, the DPORT will not respond and hang. The ICR register will generate a bus error if the DPORT is idled without the CPU or Master Port domains being in idle.

3.13.2.1 DPORT Configuration Register (DCR)

The DPORT Configuration Register bit layout is shown in Figure 3–43 and the bits are described in Table 3–48.



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–43. DPORT Configuration Register Layout (0x0200)

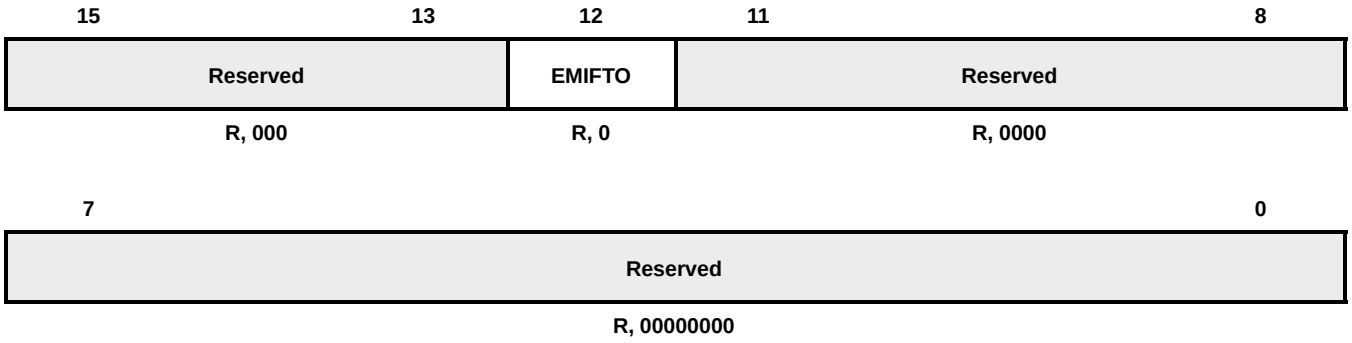
Table 3–48. DPORT Configuration Register Bit Field Description†

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–8	R	00000000	Reserved
WPE	7	R/W	0	Write Posting Enable bit† WPE = 0: Write posting disabled WPE = 1: Write posting enabled
Reserved	6–0	R	0000000	Reserved

† Write posting should not be enabled or disabled while the EMIF is conducting a transaction with external memory.

3.13.2.2 DPORT Bus Error Register (DERR)

The DPORT Bus Error Register bit layout is shown in Figure 3–44 and the bits are described in Table 3–49.



LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–44. DPORT Bus Error Register Layout (0x0202)

Table 3–49. DPORT Bus Error Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–13	R	000	Reserved
EMIFTO	12	R	0	EMIFTO bit EMIFTO = 0: No error EMIFTO = 1: Error 1 error
Reserved	11–0	R	000000000000	Reserved

3.13.3 IPORT Interface

The IPORT interfaces the I-Cache to the EMIF module. The ICACHETO bit of the IPORT Bus Error Register (IERR) can be used to determine if a time-out error has occurred during an ICACHE access to external memory. The time-out feature is enabled through the EMIFTOEN bit of the Time-Out Control Register (TOCR).

The IPORT can be placed into idle through the IPORTI bit of the Idle Control Register (ICR) and executing the IDLE instruction. The IPORT will go into idle when there are no new requests from the ICACHE. When the IPORT is in idle, it will stop accepting new requests from the CPU, it is important that the program flow not use external memory in this case. If there are requests from the CPU, the IPORT will not respond and hang. The ICR register will generate a bus error if the IPORT is idled without the CPU domain being in idle.

3.13.3.1 IPORT Bus Error Register (IERR)

The IPORT Bus Error Register bit layout is shown in Figure 3–45 and the bits are described in Table 3–50.

15	13	12	11	8
Reserved		ICACHETO	Reserved	
R, 000		R, 0	R, 0000	

7	0
Reserved	
R, 00000000	

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–45. IPORT Bus Error Register Layout (0x0302)

Table 3–50. IPORT Bus Error Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–13	R	000	Reserved
ICACHETO	12	R	0	ICACHETO bit ICACHETO = 0: No error ICACHETO = 1: A time-out error occurred during an ICACHE access to external memory.
Reserved	11–0	R	000000000000	Reserved

† See the *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371) for more information on the P-bus.

PRODUCT PREVIEW

3.13.4 System Configuration Register (CONFIG)

The System Configuration Register can be used to determine the operational state of the ICACHE. If the ICACHE is not functioning, the CACHEPRES bit of the CONFIG register will be cleared. If the ICACHE is functioning normally, this bit will be set.

The System Configuration Register bit layout is shown in Figure 3–46 and the bits are described in Table 3–51.

15 – 8	7 – 6	5	4	3 – 0
Reserved	Reserved	CACHEPRES	Reserved	Reserved
R, 00000010	R, 00	R, 0	RW, 0†	R, 0000

LEGEND: R = Read, W = Write, *n* = value at reset

† This Reserved bit *must* be kept as zero during any writes to CONFIG.

Figure 3–46. System Configuration Register Layout (0x07FD)

Table 3–51. System Configuration Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–6	R	0000001000	Reserved
CACHEPRES	5	R	0	ICACHE present CACHEPRES = 0: ICACHE is not functioning CACHEPRES = 1: ICACHE is enabled and working
Reserved	4	R/W	0†	Reserved
Reserved	3–0	R	0000	Reserved

† This Reserved bit *must* be kept as zero during any writes to CONFIG.

3.13.5 Time-Out Control Register (TOCR)

The Time-Out Control Register can be used to select whether or not a time-out error is generated when an access to a disabled/idled peripheral module occurs. If the CPU or DMA access a disabled/idle peripheral module and 512 CPU clock cycles pass without an acknowledgement from the peripheral module, then a time-out error will be sent to the corresponding module if bit 1 in the Time-Out Control Register is set. A time-out error will generate a CPU bus error that can be serviced through software by using the bus error interrupt (BERR) (see Section 3.16, Interrupts, for more information on interrupts). If the DMA gets a time-out error, it will set the TIMEOUT bit in the DMA Status Register (DMACSR) and generate a time-out error that can be serviced through software by the CPU [see the *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317) for more information on using this feature of the DMA].

The Time-Out Control Register can also be used to select whether or not a time-out error is generated when a memory access through the EMIF module stalls for more than 512 CPU clock cycles. It is recommended that this feature not be used for it can cause unexpected results.

15	8
Reserved	
R, 00000000	

7	2	1	0
Reserved		EMIFTOEN	PERITOEN
R, 000000		R/W, 0	R/W, 1

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–47. Time-Out Control Register Layout (0x9000)

Table 3–52. Time-Out Control Register Bit Field Description

BIT NAME	BIT NO.	ACCESS	RESET VALUE	DESCRIPTION
Reserved	15–2	R	00000000000000	Reserved
EMIFTOEN	1	R/W	0	EMIF time-out control bit EMIFTOEN = 0: A time-out error is not generated when an EMIF access stalls for more than 512 CPU clock cycles. EMIFTOEN = 1: A time-out error is generated when an EMIF access stalls for more than 512 CPU clock cycles.
PERITOEN	0	R/W	1	Peripheral module time-out control bit PERITOEN = 0: A time-out error is not generated when a CPU access to a disabled/idle peripheral module stalls for more than 512 CPU clock cycles. PERITOEN = 1: A time-out error is generated when a CPU access to a disabled/idle peripheral module stalls for more than 512 CPU clock cycles.

3.14 CPU Memory-Mapped Registers

The 5501 has 78 memory-mapped CPU registers that are mapped in data memory space address 0h to 4Fh. Table 3–53 provides a list of the CPU memory-mapped registers (MMRs) available. The corresponding TMS320C54x™ (C54x™) CPU registers are also indicated where applicable.

Table 3–53. CPU Memory-Mapped Registers

C54X REGISTER	C55X REGISTER	WORD ADDRESS (HEX)	C55X REGISTER DESCRIPTION	BIT FIELD
IER	IER0	00	Interrupt Enable Register 0	[15–0]
IFR	IFR0	01	Interrupt Flag Register 0	[15–0]
–	ST0_55	02	Status Register 0	[15–0]
–	ST1_55	03	Status Register 1	[15–0]
–	ST3_55	04	Status Register 3	[15–0]
–	–	05	Reserved	[15–0]
ST0	ST0	06	Status Register 0 (protected address for C54x code)	[15–0]
ST1	ST1	07	Status Register 1 (protected address for C54x code)	[15–0]
AL	AC0L	08	Accumulator 0	[15–0]
AH	AC0H	09		[31–16]
AG	AC0G	0A		[39–32]
BL	AC1L	0B	Accumulator 1	[15–0]
BH	AC1H	0C		[31–16]
BG	AC1G	0D		[39–32]
TREG	T3	0E	Temporary Register 3	[15–0]
TRN	TRN0	0F	Transition Register 0	[15–0]
AR0	AR0	10	Auxiliary Register 0	[15–0]
AR1	AR1	11	Auxiliary Register 1	[15–0]
AR2	AR2	12	Auxiliary Register 2	[15–0]
AR3	AR3	13	Auxiliary Register 3	[15–0]
AR4	AR4	14	Auxiliary Register 4	[15–0]
AR5	AR5	15	Auxiliary Register 5	[15–0]
AR6	AR6	16	Auxiliary Register 6	[15–0]
AR7	AR7	17	Auxiliary Register 7	[15–0]
SP	SP	18	Data Stack Pointer	[15–0]
BK	BK03	19	Circular Buffer Size Register for AR[0–3]	[15–0]
BRC	BRC0	1A	Block Repeat Counter 0	[15–0]
RSA	RSA0L	1B	Low Part of Block Repeat Start Address Register 0	[15–0]
REA	REA0L	1C	Low Part of Block Repeat End Address Register 0	[15–0]
PMST	PMST	1D	Status Register 3 (protected address for C54x code)	[15–0]
XPC	XPC	1E	Program Counter Extension Register for C54x code	[7–0]
–	–	1F	Reserved	[15–0]
–	T0	20	Temporary Register 0	[15–0]
–	T1	21	Temporary Register 1	[15–0]
–	T2	22	Temporary Register 2	[15–0]
–	T3	23	Temporary Register 3	[15–0]
–	AC2L	24	Accumulator 2	[15–0]
–	AC2H	25		[31–16]
–	AC2G	26		[39–32]

TMS320C54x and C54x are trademarks of Texas Instruments.

Table 3–53. CPU Memory-Mapped Registers (Continued)

C54X REGISTER	C55X REGISTER	WORD ADDRESS (HEX)	C55X REGISTER DESCRIPTION	BIT FIELD
–	CDP	27	Coefficient Data Pointer	[15–0]
–	AC3L	28	Accumulator 3	[15–0]
–	AC3H	29		[31–16]
–	AC3G	2A		[39–32]
–	DPH	2B	High Part of the Extended Data Page Register (XDP = DPH:DP)	[6–0]
–	–	2C	Reserved	[6–0]
–	–	2D	Reserved	[6–0]
–	DP	2E	Data Page Register	[15–0]
–	PDP	2F	Peripheral Data Page Register	[8–0]
–	BK47	30	Circular Buffer Size Register for AR[4–7]	[15–0]
–	BKC	31	Circular Buffer Size Register for CDP	[15–0]
–	BSA01	32	Circular Buffer Start Address Register for AR[0–1]	[15–0]
–	BSA23	33	Circular Buffer Start Address Register for AR[2–3]	[15–0]
–	BSA45	34	Circular Buffer Start Address Register for AR[4–5]	[15–0]
–	BSA67	35	Circular Buffer Start Address Register for AR[6–7]	[15–0]
–	BSAC	36	Circular Buffer Start Address Register for CDP	[15–0]
–	BIOS	37	Data Page Pointer Storage Location for 128-word Data Table	[15–0]
–	TRN1	38	Transition Register 1	[15–0]
–	BRC1	39	Block Repeat Counter 1	[15–0]
–	BRS1	3A	BRC1 Save Register	[15–0]
–	CSR	3B	Computed Single Repeat Register	[15–0]
–	RSA0H	3C	Block Repeat Start Address Register 0	[23–16]
–	RSA0L	3D		[15–0]
–	REA0H	3E	Block Repeat End Address Register 0	[23–16]
–	REA0L	3F		[15–0]
–	RSA1H	40	Block Repeat Start Address Register 1	[23–16]
–	RSA1L	41		[15–0]
–	REA1H	42	Block Repeat End Address Register 1	[23–16]
–	REA1L	43		[15–0]
–	RPTC	44	Single Repeat Counter	[15–0]
–	IER1	45	Interrupt Enable Register 1	[15–0]
–	IFR1	46	Interrupt Flag Register 1	[15–0]
–	DBIER0	47	Debug Interrupt Enable Register 0	[15–0]
–	DBIER1	48	Debug Interrupt Enable Register 0	[15–0]
–	IVPD	49	Interrupt Vector Pointer	[15–0]
–	IVPH	4A	Interrupt Vector Pointer	[15–0]
–	ST2_55	4B	Status Register 2	[15–0]
–	SSP	4C	System Stack Pointer	[15–0]
–	SP	4D	Data Stack Pointer	[15–0]
–	SPH	4E	High Part of the Extended Stack Pointers (XSP = SPH:SP, XSSP = SPH:SSP)	[6–0]
–	CDPH	4F	High Part of the Extended Coefficient Data Pointer (XCDP = CDPH:CDP)	[6–0]

3.15 Peripheral Registers

Each 5501 device has a set of memory-mapped registers associated with peripherals as listed in Table 3–54 through Table 3–72. Some registers use less than 16 bits. When reading these registers, unused bits are always read as 0.

Table 3–54. Peripheral Bus Controller Configuration Registers

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE†
0x0000	CMR	Control Mode Register	1111 1110 1001 0011
0x0001	ICR	Idle Configuration Register	0000 0000 0000 0000
0x0002	ISTR	Idle Status Register	0000 0000 0000 0000
0x0003	PER1	Priority Encoder Register 1	00001–00000000
0x0004	PER2	Priority Encoder Register 2	01100–00000010
0x0005	PER3	Priority Encoder Register 3	00111–00000110
0x0006	PER4	Priority Encoder Register 4	01001–00001000
0x0007	PER5	Priority Encoder Register 5	01111–00001110
0x0008	PER6	Priority Encoder Register 6	10101–00010000
0x0009	PER7	Priority Encoder Register 7	00100–00000011
0x000A	PER8	Priority Encoder Register 8	01011–00001000
0x000B	PER9	Priority Encoder Register 9	10010–00010001
0x000C	PER10	Priority Encoder Register 10	10100–00010011
0x000D	PER11	Priority Encoder Register 11	00101–00011101
0x000E	Reserved		
0x000F	BOOT_MOD	Boot Mode Register (read only)	Value of GPIO[2:0] at reset
0x0010	Reserved		
0x0011	Reserved		
0x0100	XCR	XPORT Configuration Register	1000 0000 0000 0000
0x0102	XERR	XPORT Bus Error Register	1000 0000 0000 0000
0x0200	DCR	DPORT Configuration Register	0000 0000 0000 0000
0x0202	DERR	DPORT Bus Error Register	0000 0000 0000 0000
0x0302	IERR	IPORT Bus Error Register	0000 0000 0000 0000
0x07FD	CONFIG	System Configuration Register	0000 0001 0000 0000

† x denotes a “don’t care.”

Table 3–55. External Memory Interface Registers

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE†
0x0800	EGCR1	EMIF Global Control Register 1	0000 0000 0110 0000
0x0801	EGCR2	EMIF Global Control Register 2	0000 0000 0000 1001
0x0802	CE1_1	EMIF CE1 Space Control Register 1	1111 1111 1111 0011
0x0803	CE1_2	EMIF CE1 Space Control Register 2	1111 1111 1111 1111
0x0804	CE0_1	EMIF CE0 Space Control Register 1	0000 0000 0000 0010
0x0805	CE0_2	EMIF CE0 Space Control Register 2	0000 0000 0000 0000
0x0806	–	Reserved	
0x0807	–	Reserved	
0x0808	CE2_1	EMIF CE2 Space Control Register 1	1111 1111 1111 0011
0x0809	CE2_2	EMIF CE2 Space Control Register 2	1111 1111 1111 1111
0x080A	CE3_1	EMIF CE3 Space Control Register 1	1111 1111 1111 0011
0x080B	CE3_2	EMIF CE3 Space Control Register 2	1111 1111 1111 1111
0x080C	SDC1	EMIF SDRAM Control Register 1	1111 0000 0000 0000
0x080D	SDC2	EMIF SDRAM Control Register 2	0000 0011 0100 1000
0x080E	SDRC1	EMIF SDRAM Refresh Control Register 1	1100 0101 1101 1100
0x080F	SDRC2	EMIF SDRAM Refresh Control Register 2	0000 0000 0101 1101
0x0810	SDX1	EMIF SDRAM Extension Register 1	1111 1001 0100 1000
0x0811	SDX2	EMIF SDRAM Extension Register 2	1111 1001 0100 1000
0x0812	–	Reserved	
:	–	:	
0x0821	–	Reserved	
0x0822	CE1_SC1	EMIF CE1 Secondary Control Register 1	0000 0000 0000 0010
0x0823	CE1_SC2	EMIF CE1 Secondary Control Register 2	0000 0000 0000 0000
0x0824	CE0_SC1	EMIF CE0 Secondary Control Register 1	0000 0000 0000 0010
0x0825	CE0_SC2	EMIF CE0 Secondary Control Register 2	0000 0000 0000 0000
0x0826	–	Reserved	
0x0827	–	Reserved	
0x0828	CE2_SC1	EMIF CE2 Secondary Control Register 1	0000 0000 0000 0010
0x0829	CE2_SC2	EMIF CE2 Secondary Control Register 2	0000 0000 0000 0000
0x082A	CE3_SC1	EMIF CE3 Secondary Control Register 1	0000 0000 0000 0010
0x082B	CE3_SC2	EMIF CE3 Secondary Control Register 2	0000 0000 0000 0000
0x082C	–	Reserved	
:	–	:	
0x0839	–	Reserved	

† x denotes a “don’t care.”

Table 3–56. DMA Configuration Registers

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
		GLOBAL REGISTER	
0x0E00	DMA_GCR(2:0)	DMA Global Control Register	000
0x0E01	DMA_GTCR(3:0)	DMA Global Timeout Control Register	0000
		CHANNEL #0 REGISTERS	
0x0C00	DMA_CSDP0	DMA Channel 0 Source Destination Parameters Register	0000 0000 0000 0000
0x0C01	DMA_CCR0(15:0)	DMA Channel 0 Control Register	0000 0000 0000 0000
0x0C02	DMA_CICR0(5:0)	DMA Channel 0 Interrupt Control register	00 0011
0x0C03	DMA_CSR0(6:0)	DMA Channel 0 Status register	00 0000
0x0C04	DMA_CSSA_L0	DMA Channel 0 Source Start Address, lower bits, register	Undefined
0x0C05	DMA_CSSA_U0	DMA Channel 0 Source Start Address, upper bits, register	Undefined
0x0C06	DMA_CDSA_L0	DMA Channel 0 Source Destination Address, lower bits, register	Undefined
0x0C07	DMA_CDSA_U0	DMA Channel 0 Source Destination Address, upper bits, register	Undefined
0x0C08	DMA_CEN0	DMA Channel 0 Element Number register	Undefined
0x0C09	DMA_CFN0	DMA Channel 0 Frame Number register	Undefined
0x0C0A	DMA_CSFI0	DMA Channel 0 Source Frame Index register	Undefined
0x0C0B	DMA_CSEI0	DMA Channel 0 Source Element Index register	Undefined
0x0C0C	DMA_CSAC0	DMA Channel 0 Source Address Counter register	Undefined
0x0C0D	DMA_CDAC0	DMA Channel 0 Destination Address Counter register	Undefined
0x0C0E	DMA_CDEI0	DMA Channel 0 Destination Element Index register	Undefined
0x0C0F	DMA_CDFI0	DMA Channel 0 Destination Frame Index register	Undefined
		CHANNEL #1 REGISTERS	
0x0C20	DMA_CSDP1	DMA Channel 1 Source Destination Parameters Register	0000 0000 0000 0000
0x0C21	DMA_CCR1(15:0)	DMA Channel 1 Control Register	0000 0000 0000 0000
0x0C22	DMA_CICR1(5:0)	DMA Channel 1 Interrupt Control register	00 0011
0x0C23	DMA_CSR1(6:0)	DMA Channel 1 Status register	00 0000
0x0C24	DMA_CSSA_L1	DMA Channel 1 Source Start Address, lower bits, register	Undefined
0x0C25	DMA_CSSA_U1	DMA Channel 1 Source Start Address, upper bits, register	Undefined
0x0C26	DMA_CDSA_L1	DMA Channel 1 Source Destination Address, lower bits, register	Undefined
0x0C27	DMA_CDSA_U1	DMA Channel 1 Source Destination Address, upper bits, register	Undefined
0x0C28	DMA_CEN1	DMA Channel 1 Element Number register	Undefined
0x0C29	DMA_CFN1	DMA Channel 1 Frame Number register	Undefined
0x0C2A	DMA_CSFI1	DMA Channel 1 Source Frame Index register	Undefined
0x0C2B	DMA_CSEI1	DMA Channel 1 Source Element Index register	Undefined
0x0C2C	DMA_CSAC1	DMA Channel 1 Source Address Counter register	Undefined
0x0C2D	DMA_CDAC1	DMA Channel 1 Destination Address Counter register	Undefined
0x0C2E	DMA_CDEI1	DMA Channel 1 Destination Element Index register	Undefined
0x0C2F	DMA_CDFI1	DMA Channel 1 Destination Frame Index register	Undefined

Table 3–56. DMA Configuration Registers (Continued)

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
CHANNEL #2 REGISTERS			
0x0C40	DMA_CSDP2	DMA Channel 2 Source Destination Parameters Register	0000 0000 0000 0000
0x0C41	DMA_CCR2(15:0)	DMA Channel 2 Control Register	0000 0000 0000 0000
0x0C42	DMA_CICR2(5:0)	DMA Channel 2 Interrupt Control register	00 0011
0x0C43	DMA_CSR2(6:0)	DMA Channel 2 Status register	00 0000
0x0C44	DMA_CSSA_L2	DMA Channel 2 Source Start Address, lower bits, register	Undefined
0x0C45	DMA_CSSA_U2	DMA Channel 2 Source Start Address, upper bits, register	Undefined
0x0C46	DMA_CDSA_L2	DMA Channel 2 Source Destination Address, lower bits, register	Undefined
0x0C47	DMA_CDSA_U2	DMA Channel 2 Source Destination Address, upper bits, register	Undefined
0x0C48	DMA_CEN2	DMA Channel 2 Element Number register	Undefined
0x0C49	DMA_CFN2	DMA Channel 2 Frame Number register	Undefined
0x0C4A	DMA_CSFI2	DMA Channel 2 Source Frame Index register	Undefined
0x0C4B	DMA_CSEI2	DMA Channel 2 Source Element Index register	Undefined
0x0C4C	DMA_CSAC2	DMA Channel 2 Source Address Counter register	Undefined
0x0C4D	DMA_CDAC2	DMA Channel 2 Destination Address Counter register	Undefined
0x0C4E	DMA_CDEI2	DMA Channel 2 Destination Element Index register	Undefined
0x0C4F	DMA_CDFI2	DMA Channel 2 Destination Frame Index register	Undefined
CHANNEL #3 REGISTERS			
0x0C60	DMA_CSDP3	DMA Channel 3 Source Destination Parameters Register	0000 0000 0000 0000
0x0C61	DMA_CCR3(15:0)	DMA Channel 3 Control Register	0000 0000 0000 0000
0x0C62	DMA_CICR3(5:0)	DMA Channel 3 Interrupt Control register	00 0011
0x0C63	DMA_CSR3(6:0)	DMA Channel 3 Status register	00 0000
0x0C64	DMA_CSSA_L3	DMA Channel 3 Source Start Address, lower bits, register	Undefined
0x0C65	DMA_CSSA_U3	DMA Channel 3 Source Start Address, upper bits, register	Undefined
0x0C66	DMA_CDSA_L3	DMA Channel 3 Source Destination Address, lower bits, register	Undefined
0x0C67	DMA_CDSA_U3	DMA Channel 3 Source Destination Address, upper bits, register	Undefined
0x0C68	DMA_CEN3	DMA Channel 3 Element Number register	Undefined
0x0C69	DMA_CFN3	DMA Channel 3 Frame Number register	Undefined
0x0C6A	DMA_CSFI3	DMA Channel 3 Source Frame Index register	Undefined
0x0C6B	DMA_CSEI3	DMA Channel 3 Source Element Index register	Undefined
0x0C6C	DMA_CSAC3	DMA Channel 3 Source Address Counter register	Undefined
0x0C6D	DMA_CDAC3	DMA Channel 3 Destination Address Counter register	Undefined
0x0C6E	DMA_CDEI3	DMA Channel 3 Destination Element Index register	Undefined
0x0C6F	DMA_CDFI3	DMA Channel 3 Destination Frame Index register	Undefined

Table 3–56. DMA Configuration Registers (Continued)

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
CHANNEL #4 REGISTERS			
0x0C80	DMA_CSDP4	DMA Channel 4 Source Destination Parameters Register	0000 0000 0000 0000
0x0C81	DMA_CCR4(15:0)	DMA Channel 4 Control Register	0000 0000 0000 0000
0x0C82	DMA_CICR4(5:0)	DMA Channel 4 Interrupt Control register	00 0011
0x0C83	DMA_CSR4(6:0)	DMA Channel 4 Status register	00 0000
0x0C84	DMA_CSSA_L4	DMA Channel 4 Source Start Address, lower bits, register	Undefined
0x0C85	DMA_CSSA_U4	DMA Channel 4 Source Start Address, upper bits, register	Undefined
0x0C86	DMA_CDSA_L4	DMA Channel 4 Source Destination Address, lower bits, register	Undefined
0x0C87	DMA_CDSA_U4	DMA Channel 4 Source Destination Address, upper bits, register	Undefined
0x0C88	DMA_CEN4	DMA Channel 4 Element Number register	Undefined
0x0C89	DMA_CFN4	DMA Channel 4 Frame Number register	Undefined
0x0C8A	DMA_CSFI4	DMA Channel 4 Source Frame Index register	Undefined
0x0C8B	DMA_CSEI4	DMA Channel 4 Source Element Index register	Undefined
0x0C8C	DMA_CSAC4	DMA Channel 4 Source Address Counter register	Undefined
0x0C8D	DMA_CDAC4	DMA Channel 4 destination Address Counter register	Undefined
0x0C8E	DMA_CDEI4	DMA Channel 4 Destination Element Index register	Undefined
0x0C8F	DMA_CDFI4	DMA Channel 4 Destination Frame Index register	Undefined
CHANNEL #5 REGISTERS			
0x0CA0	DMA_CSDP5	DMA Channel 5 Source Destination Parameters Register	0000 0000 0000 0000
0x0CA1	DMA_CCR5(15:0)	DMA Channel 5 Control Register	0000 0000 0000 0000
0x0CA2	DMA_CICR5(5:0)	DMA Channel 5 Interrupt Control register	00 0011
0x0CA3	DMA_CSR5(6:0)	DMA Channel 5 Status register	00 0000
0x0CA4	DMA_CSSA_L5	DMA Channel 5 Source Start Address, lower bits, register	Undefined
0x0CA5	DMA_CSSA_U5	DMA Channel 5 Source Start Address, upper bits, register	Undefined
0x0CA6	DMA_CDSA_L5	DMA Channel 5 Source Destination Address, lower bits, register	Undefined
0x0CA7	DMA_CDSA_U5	DMA Channel 5 Source Destination Address, upper bits, register	Undefined
0x0CA8	DMA_CEN5	DMA Channel 5 Element Number register	Undefined
0x0CA9	DMA_CFN5	DMA Channel 5 Frame Number register	Undefined
0x0CAA	DMA_CSFI5	DMA Channel 5 Source Frame Index register	Undefined
0x0CAB	DMA_CSEI5	DMA Channel 5 Source Element Index register	Undefined
0x0CAC	DMA_CSAC5	DMA Channel 5 Source Address Counter register	Undefined
0x0CAD	DMA_CDAC5	DMA Channel 5 Destination Address Counter register	Undefined
0x0CAE	DMA_CDEI5	DMA Channel 5 Destination Element Index register	Undefined
0x0CAF	DMA_CDFI5	DMA Channel 5 Destination Frame Index register	Undefined

Table 3–57. Instruction Cache Registers

WORD ADDRESS	REGISTER NAME	DESCRIPTION
0x1400	ICGC	ICache Global Control Register
0x1401	ICFLARL	ICache Flush Line Address Register Low Part
0x1402	ICFLARH	ICache Flush Line Address Register High Part
0x1403	ICWMC	ICache N-Way Control Register

Table 3–58. Trace FIFO†

WORD ADDRESS	REGISTER NAME	DESCRIPTION
0x2000 – 0x203F	TRC00 – TRC63	Trace Register Discontinuity Section
0x2040 – 0x204F	TRC64 – TRC79	Trace Register Last PC Section
0x2050	TRC_LPCOFFSET1	Trace LPC Offset Register 1
0x2051	TRC_LPCOFFSET2	Trace LPC Offset Register 2
0x2052	TRC_PTR	Trace Pointer Register
0x2053	TRC_CNTL	Trace Control Register
0x2054	TRC_ID	Trace ID Register

† The Trace FIFO registers are used by the emulator only and do not require any intervention from the user.

Table 3–59. Timer Signal Selection Register

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x8000	TSSR	Timer Signal Selection Register	0000 0000 0000 0000

Table 3–60. Timers

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x1000	GPTPID1_0	Peripheral ID register 1, Timer #0	0000 0111 0000 0001
0x1001	GPTPID2_0	Peripheral ID register 2, Timer #0	0000 0000 0000 0001
0x1002	GPTEMU_0	Emulation Management Register, Timer #0	0000 0000 0000 0000
0x1003	GPTCLK_0	Timer Clock Speed Register, Timer #0	0000 0001 0000 1000
0x1004	GPTGPINT_0	GPIO Interrupt Control Register, Timer #0	0000 0000 0000 0000
0x1005	GPTGPEN_0	GPIO Enable Register, Timer #0	0000 0000 0000 0000
0x1006	GPTGPDAT_0	GPIO Data Register, Timer #0	0000 0000 0000 0000
0x1007	GPTGPDIR_0	GPIO Direction Register, Timer #0	0000 0000 0000 0000
0x1008	GPTCNT1_0	Timer Counter 1 Register, Timer #0	0000 0000 0000 0000
0x1009	GPTCNT2_0	Timer Counter 2 Register, Timer #0	0000 0000 0000 0000
0x100A	GPTCNT3_0	Timer Counter 3 Register, Timer #0	0000 0000 0000 0000
0x100B	GPTCNT4_0	Timer Counter 4 Register, Timer #0	0000 0000 0000 0000
0x100C	GPTPRD1_0	Period Register 1, Timer #0	0000 0000 0000 0000
0x100D	GPTPRD2_0	Period Register 2, Timer #0	0000 0000 0000 0000
0x100E	GPTPRD3_0	Period Register 3, Timer #0	0000 0000 0000 0000
0x100F	GPTPRD4_0	Period Register 4, Timer #0	0000 0000 0000 0000
0x1010	GPTCTL1_0	Timer Control Register 1, Timer #0	0000 0000 0000 0000
0x1011	GPTCTL2_0	Timer Control Register 2, Timer #0	1111 1111 0000 0000
0x1012	GPTGCTL1_0	Global Timer Control Register 1, Timer #0	0000 0000 0000 0000
0x2400	GPTPID1_1	Peripheral ID register 1, Timer #1	0000 0111 0000 0001
0x2401	GPTPID2_1	Peripheral ID register 2, Timer #1	0000 0000 0000 0001
0x2402	GPTEMU_1	Emulation Management Register, Timer #1	0000 0000 0000 0000
0x2403	GPTCLK_1	Timer Clock Speed Register, Timer #1	0000 0001 0000 1000
0x2404	GPTGPINT_1	GPIO Interrupt Control Register, Timer #1	0000 0000 0000 0000
0x2405	GPTGPEN_1	GPIO Enable Register, Timer #1	0000 0000 0000 0000

Table 3–60. Timers (Continued)

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x2406	GPTGPDAT_1	GPIO Data Register, Timer #1	0000 0000 0000 0000
0x2407	GPTGPDIR_1	GPIO Direction Register, Timer #1	0000 0000 0000 0000
0x2408	GPTCNT1_1	Timer Counter 1 Register, Timer #1	0000 0000 0000 0000
0x2409	GPTCNT2_1	Timer Counter 2 Register, Timer #1	0000 0000 0000 0000
0x240A	GPTCNT3_1	Timer Counter 3 Register, Timer #1	0000 0000 0000 0000
0x240B	GPTCNT4_1	Timer Counter 4 Register, Timer #1	0000 0000 0000 0000
0x240C	GPTPRD1_1	Period Register 1, Timer #1	0000 0000 0000 0000
0x240D	GPTPRD2_1	Period Register 2, Timer #1	0000 0000 0000 0000
0x240E	GPTPRD3_1	Period Register 3, Timer #1	0000 0000 0000 0000
0x240F	GPTPRD4_1	Period Register 4, Timer #1	0000 0000 0000 0000
0x2410	GPTCTL1_1	Timer Control Register 1, Timer #1	0000 0000 0000 0000
0x2411	GPTCTL2_1	Timer Control Register 2, Timer #1	1111 1111 0000 0000
0x2412	GPTGCTL1_1	Global Timer Control Register 1, Timer #1	0000 0000 0000 0000
0x4000	WDTPID1	Peripheral ID register 1, Watchdog Timer	0000 0111 0000 0001
0x4001	WDTPID2	Peripheral ID register 2, Watchdog Timer	0000 0000 0000 0001
0x4002	WDTEMU	Emulation Management Register, Watchdog Timer	0000 0000 0000 0000
0x4003	WDTCLK	Timer Clock Speed Register, Watchdog Timer	0000 0001 0000 1000
0x4004	WDTGPINT	GPIO Interrupt Control Register, Watchdog Timer	0000 0000 0000 0000
0x4005	WDTGPEN	GPIO Enable Register, Watchdog Timer	0000 0000 0000 0000
0x4006	WDTGPDAT	GPIO Data Register, Watchdog Timer	0000 0000 0000 0000
0x4007	WDTGPDIR	GPIO Direction Register, Watchdog Timer	0000 0000 0000 0000
0x4008	WDTCNT1	Timer Counter 1 Register, Watchdog Timer	0000 0000 0000 0000
0x4009	WDTCNT2	Timer Counter 2 Register, Watchdog Timer	0000 0000 0000 0000
0x400A	WDTCNT3	Timer Counter 3 Register, Watchdog Timer	0000 0000 0000 0000
0x400B	WDTCNT4	Timer Counter 4 Register, Watchdog Timer	0000 0000 0000 0000
0x400C	WDTPRD1	Period Register 1, Watchdog Timer	0000 0000 0000 0000
0x400D	WDTPRD2	Period Register 2, Watchdog Timer	0000 0000 0000 0000
0x400E	WDTPRD3	Period Register 3, Watchdog Timer	0000 0000 0000 0000
0x400F	WDTPRD4	Period Register 4, Watchdog Timer	0000 0000 0000 0000
0x4010	WDTCTL1	Timer Control Register 1, Watchdog Timer	0000 0000 0000 0000
0x4011	WDTCTL2	Timer Control Register 2, Watchdog Timer	1111 1111 0000 0000
0x4012	WDTGCTL1	Global Timer Control Register 1, Watchdog Timer	0000 0000 0000 0000
0x4014	WDTWCTL1	WD Timer Control Register 1, Watchdog Timer	0000 0000 0000 0000
0x4015	WDTWCTL2	WD Timer Control Register 2, Watchdog Timer	0000 0000 0000 0000

Table 3–61. Multichannel Serial Port #0

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x2800	DRR1_0	Data Receive Register 1, McBSP #0	0000 0000 0000 0000
0x2801	DRR2_0	Data Receive Register 2, McBSP #0	0000 0000 0000 0000
0x2802	DXR1_0	Data Transmit Register 1, McBSP #0	0000 0000 0000 0000
0x2803	DXR2_0	Data Transmit Register 2, McBSP #0	0000 0000 0000 0000
0x2804	SPCR1_0	Serial Port Control Register 1, McBSP #0	0000 0000 0000 0000
0x2805	SPCR2_0	Serial Port Control Register 2, McBSP #0	0000 0000 0000 0000
0x2806	RCR1_0	Receive Control Register 1, McBSP #0	0000 0000 0000 0000
0x2807	RCR2_0	Receive Control Register 2, McBSP #0	0000 0000 0000 0000
0x2808	XCR1_0	Transmit Control Register 1, McBSP #0	0000 0000 0000 0000
0x2809	XCR2_0	Transmit Control Register 2, McBSP #0	0000 0000 0000 0000
0x280A	SRGR1_0	Sample Rate Generator Register 1, McBSP #0	0020 0000 0000 0000
0x280B	SRGR2_0	Sample Rate Generator Register 2, McBSP #0	0000 0000 0000 0001
0x280C	MCR1_0	Multichannel Control Register 1, McBSP #0	0000 0000 0000 0000
0x280D	MCR2_0	Multichannel Control Register 2, McBSP #0	0000 0000 0000 0000
0x280E	RCERA_0	Receive Channel Enable Register Partition A, McBSP #0	0000 0000 0000 0000
0x280F	RCERB_0	Receive Channel Enable Register Partition B, McBSP #0	0000 0000 0000 0000
0x2810	XCERA_0	Transmit Channel Enable Register Partition A, McBSP #0	0000 0000 0000 0000
0x2811	XCERB_0	Transmit Channel Enable Register Partition B, McBSP #0	0000 0000 0000 0000
0x2812	PCR0	Pin Control Register, McBSP #0	0000 0000 0000 0000
0x2813		Reserved	
0x2814	RCERC_0	Receive Channel Enable Register Partition C, McBSP #0	0000 0000 0000 0000
0x2815	RCERD_0	Receive Channel Enable Register Partition D, McBSP #0	0000 0000 0000 0000
0x2816	XCERC_0	Transmit Channel Enable Register Partition C, McBSP #0	0000 0000 0000 0000
0x2817	XCERD_0	Transmit Channel Enable Register Partition D, McBSP #0	0000 0000 0000 0000
0x2818	RCERE_0	Receive Channel Enable Register Partition E, McBSP #0	0000 0000 0000 0000
0x2819	RCERF_0	Receive Channel Enable Register Partition F, McBSP #0	0000 0000 0000 0000
0x281A	XCERE_0	Transmit Channel Enable Register Partition E, McBSP #0	0000 0000 0000 0000
0x281B	XCERF_0	Transmit Channel Enable Register Partition F, McBSP #0	0000 0000 0000 0000
0x281C	RCERG_0	Receive Channel Enable Register Partition G, McBSP #0	0000 0000 0000 0000
0x281D	RCERH_0	Receive Channel Enable Register Partition H, McBSP #0	0000 0000 0000 0000
0x281E	XCERG_0	Transmit Channel Enable Register Partition G, McBSP #0	0000 0000 0000 0000
0x281F	XCERH_0	Transmit Channel Enable Register Partition H, McBSP #0	0000 0000 0000 0000

Table 3–62. Multichannel Serial Port #1

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x2C00	DRR1_1	Data Receive Register 1, McBSP #1	0000 0000 0000 0000
0x2C01	DRR2_1	Data Receive Register 2, McBSP #1	0000 0000 0000 0000
0x2C02	DXR1_1	Data Transmit Register 1, McBSP #1	0000 0000 0000 0000
0x2C03	DXR2_1	Data Transmit Register 2, McBSP #1	0000 0000 0000 0000
0x2C04	SPCR1_1	Serial Port Control Register 1, McBSP #1	0000 0000 0000 0000
0x2C05	SPCR2_1	Serial Port Control Register 2, McBSP #1	0000 0000 0000 0000
0x2C06	RCR1_1	Receive Control Register 1, McBSP #1	0000 0000 0000 0000
0x2C07	RCR2_1	Receive Control Register 2, McBSP #1	0000 0000 0000 0000
0x2C08	XCR1_1	Transmit Control Register 1, McBSP #1	0000 0000 0000 0000
0x2C09	XCR2_1	Transmit Control Register 2, McBSP #1	0000 0000 0000 0000
0x2C0A	SRGR1_1	Sample Rate Generator Register 1, McBSP #1	0020 0000 0000 0000
0x2C0B	SRGR2_1	Sample Rate Generator Register 2, McBSP #1	0000 0000 0000 0001
0x2C0C	MCR1_1	Multichannel Register 1, McBSP #1	0000 0000 0000 0000
0x2C0D	MCR2_1	Multichannel Register 2, McBSP #1	0000 0000 0000 0000
0x2C0E	RCERA_1	Receive Channel Enable Register Partition A, McBSP #1	0000 0000 0000 0000
0x2C0F	RCERB_1	Receive Channel Enable Register Partition B, McBSP #1	0000 0000 0000 0000
0x2C10	XCERA_1	Transmit Channel Enable Register Partition A, McBSP #1	0000 0000 0000 0000
0x2C11	XCERB_1	Transmit Channel Enable Register Partition B, McBSP #1	0000 0000 0000 0000
0x2C12	PCR1	Pin Control Register, McBSP #1	0000 0000 0000 0000
0x2C13		Reserved	
0x2C14	RCERC_1	Receive Channel Enable Register Partition C, McBSP #1	0000 0000 0000 0000
0x2C15	RCERD_1	Receive Channel Enable Register Partition D, McBSP #1	0000 0000 0000 0000
0x2C16	XCERC_1	Transmit Channel Enable Register Partition C, McBSP #1	0000 0000 0000 0000
0x2C17	XCERD_1	Transmit Channel Enable Register Partition D, McBSP #1	0000 0000 0000 0000
0x2C18	RCERE_1	Receive Channel Enable Register Partition E, McBSP #1	0000 0000 0000 0000
0x2C19	RCERF_1	Receive Channel Enable Register Partition F, McBSP #1	0000 0000 0000 0000
0x2C1A	XCERE_1	Transmit Channel Enable Register Partition E, McBSP #1	0000 0000 0000 0000
0x2C1B	XCERF_1	Transmit Channel Enable Register Partition F, McBSP #1	0000 0000 0000 0000
0x2C1C	RCERG_1	Receive Channel Enable Register Partition G, McBSP #1	0000 0000 0000 0000
0x2C1D	RCERH_1	Receive Channel Enable Register Partition H, McBSP #1	0000 0000 0000 0000
0x2C1E	XCERG_1	Transmit Channel Enable Register Partition G, McBSP #1	0000 0000 0000 0000
0x2C1F	XCERH_1	Transmit Channel Enable Register Partition H, McBSP #1	0000 0000 0000 0000

Table 3–63. HPI

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE†
0xA000	PID LSW	PID [15:0]	0000 0010
0xA001	PID MSW	PID [31:16]	0000 0000 0000 0001
0xA002	PWREMU_MGMT1	Power and Management Register 1	0000 0000 0000 0000
0xA003 – 0xA005		Reserved	
0xA006	HGPIOEN	HPI GPIO enable register	0000 0000 0000 0000
0xA007		Reserved	
0xA008	HGPIODIR	HPI GPIO direction register	0000 0000 0000 0000
0xA009		Reserved	
0xA00A	GPIODAT	GPIO data register	0000 0000 0000 0000
0xA00B – 0xA017		Reserved	
0xA018	HPIC	Host Port Control Register	0000 0000 0000 1000
0xA019		Reserved	
0xA01A	HPIAW	Host Port Write Address Register	xxxx xxxx xxxx xxxx
0xA01B		Reserved	
0xA01C	HPIAR	Host Port Read Address Register	xxxx xxxx xxxx xxxx
0xA01D – 0xA020		Reserved	

† x denotes a “don’t care.”

Table 3–64. GPIO

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE†
0x3400	IODIR	General-purpose I/O Direction Register	0000 0000 0000 0000
0x3401	IODATA	General-purpose I/O Data Register	0000 0000 xxxx xxxx
0x4400	PGPIOEN0	Parallel GPIO Enable Register 0	0000 0000 0000 0000
0x4401	PGPIODIR0	Parallel GPIO Direction Register 0	0000 0000 0000 0000
0x4402	PGPIODAT0	Parallel GPIO Data Register 0	0000 0000 0000 0000
0x4403	PGPIOEN1	Parallel GPIO Enable Register 1	0000 0000 0000 0000
0x4404	PGPIODIR1	Parallel GPIO Direction Register 1	0000 0000 0000 0000
0x4405	PGPIODAT1	Parallel GPIO Data Register 1	0000 0000 0000 0000
0x4406	PGPIOEN2	Parallel GPIO Enable Register 2	0000 0000 0000 0000
0x4407	PGPIODIR2	Parallel GPIO Direction Register 2	0000 0000 0000 0000
0x4408	PGPIODAT2	Parallel GPIO Data Register 2	0000 0000 0000 0000

† x denotes a “don’t care.”

Table 3–65. Device Revision ID

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE†
0x3800 – 0x3803	Die ID	Die ID	
0x3804	Chip ID (LSW)	Defines F# 3LS digits and PG rev	1001 0100 0110 0001
0x3805	Chip ID (MSW)	Defines F# 3MS digits	0000 0111 0101 0001
0x3806	Sub ID	Defines subsytem ID	0000 0000 0000 0000‡
0x3807	Cat ID	Defines catalog device	0101 0101 0000 0001 (5501h)

† x denotes a “don’t care.”
‡ Denotes single core

Table 3–66. I^2C

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE†
0x3C00	I2COAR	I^2C Own Address Register	0000 0000 0000 0000
0x3C01	I2CIER	I^2C Interrupt Enable Register	0000 0000 0000 0000
0x3C02	I2CSTR	I^2C Status Register	0000 0001 0000 0000
0x3C03	I2CCLKL	I^2C Clock Low-Time Divider Register	0000 0000 0000 0000
0x3C04	I2CCLKH	I^2C Clock High-Time Divider Register	0000 0000 0000 0000
0x3C05	I2CCNT	I^2C Data Count	0000 0000 0000 0000
0x3C06	I2CDRR	I^2C Data Receive Register	0000 0000 0000 0000
0x3C07	I2CSAR	I^2C Slave Address Register	0000 0011 1111 1111
0x3C08	I2CDXR	I^2C Data Transmit Register	0000 0000 0000 0000
0x3C09	I2CMDR	I^2C Mode Register	0000 0000 0000 0000
0x3C0A	I2CISRC	I^2C Interrupt Source Register	0000 0000 0000 0000
0x3C0B	I2CGPIO	I^2C General-Purpose Register (Not supported)	xxxx xxxx xxxx xxxx
0x3C0C	I2CPSC	I^2C Prescaler Register	0000 0000 0000 0000
0x3C0D	PID1	I^2C Peripheral ID Register 1	–
0x3C0E	PID2	I^2C Peripheral ID Register 2	–
–	I2CXSR	I^2C Transmit Shift Register	–
–	I2CRSR	I^2C Receive Shift Register	–

† x denotes a “don’t care.”

Table 3–67. UART

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE†
0x9C00	URRBR/ URTHR/ URDLL‡	Receive Buffer Register Transmit Holding Register Divisor Latch LSB Register	xxxx xxxx
0x9C01	URIER/ URDLM§	Interrupt Enable Register Divisor Latch MSB Register	0000 0000
0x9C02	URIIR/ URFCR¶	Interrupt Identification Register FIFO Control Register	0000 0001 0000 0000
0x9C03	URLCR	Line Control Register	0000 0000
0x9C04	URMCR	Modem Control Register	0000 0000
0x9C05	URLSR	Line Status Register	0110 0000
0x9C07	URSCR	Scratch Register	xxxx xxxx
0x9C08	URDLL‡	Divisor Latch LSB Register	–
0x9C09	URDLM§	Divisor Latch MSB Register	–
0x9C0A	URPID1	Peripheral ID Register (LSW)	–
0x9C0B	URPID2	Peripheral ID Register (MSW)	–
0x9C0C	URPECR	Power and Emulation Control Register	0000 0000 0000 0000

† x denotes a “don’t care.”

‡ The registers URRBR, URTHR, and URDLL share one address. URDLL also has a dedicated address. When using the dedicated address, the DLAB bit can be kept cleared, so that URRBR and URTHR are always selected at the shared address.

If DLAB = 0 : Read Only: URRBR
Write Only: URTHR

If DLAB = 1: Read/Write: URDLL

§ The registers URIER and URDLM share one address. URDLM also has a dedicated address. When using the dedicated address, the DLAB bit can be kept cleared, so that URIER is always selected at the shared address.

If DLAB = 0: Read/WRite: URIER

If DLAB = 1: Read/Write: URDLM

¶ The registers URIIR and URFCR share one address.

Read Only: URIIR

Write Only: URFCR

Table 3–68. External Bus Selection

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x6C00	XBSR	External Bus Selection Register	0000 0000 0000 0000
0x6C01	XBCR	External Bus Control Register	0000 0000 0000 0000

Table 3–69. Clock Mode Register

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x8C00	CLKMD	Clock Mode Control Register	0000 0000 0000 0000

Table 3–70. CLKOUT Selector Register

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x8400	CLKOUTSR	CLKOUT Selection Register	0000 0000 0000 0010

Table 3–71. Clock Controller Registers

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x1C80	PLLCSR	PLL Control Status Register	0000 0000 0000 0000
0x1C82	CK3SEL	CLKOUT3 Select Register	0000 0000 0000 1011
0x1C88	PLLM	PLL Multiplier Control Register	0000 0000 0000 0000
0x1C8A	PLLDIV0	PLL Divider 0 Register	1000 0000 0000 0000
0x1C8C	PLLDIV1	PLL Divider 1 Register	0000 0000 0000 0000
0x1C8E	PLLDIV2	PLL Divider 2 Register	0000 0000 0000 0000
0x1C90	PLLDIV3	PLL Divider 3 Register	0000 0000 0000 0000
0x1C92	OSCDIV1	Oscillator Divider 1 Register	0000 0000 0000 0000
0x1C98	WKEN	Oscillator Wakeup Control Register	0000 0000 0000 0000

Table 3–72. IDLE Control Registers

WORD ADDRESS	REGISTER NAME	DESCRIPTION	RESET VALUE
0x9400	PICR	Peripheral IDLE Control Register	0000 0000 0000 0000
0x9401	PISTR	Peripheral IDLE Status Register	0000 0000 0000 0010
0x9402	MICR	Master IDLE Control Register	0000 0000 0000 0000
0x9403	MISR	Master IDLE Status Register	0000 0000 0000 0010

3.16 Interrupts

Vector-relative locations and priorities for all internal and external interrupts are shown in Table 3–73. For more information on setting up and using interrupts, please refer to the *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371).

Table 3–73. Interrupt Table

NAME	SOFTWARE (TRAP) EQUIVALENT	LOCATION (HEX BYTES)	PRIORITY	FUNCTION
RESET	SINT0	0	0	Reset (hardware and software)
NMI	SINT1	8	1	Nonmaskable interrupt
INT0	SINT2	10	3	External interrupt #0
INT2	SINT3	18	5	External interrupt #2
TINT0	SINT4	20	6	Timer #0 interrupt
RINT0	SINT5	28	7	McBSP #0 receive interrupt
RINT1	SINT6	30	9	McBSP #1 receive interrupt
XINT1	SINT7	38	10	McBSP #1 transmit interrupt
LCKINT	SINT8	40	11	PLL lock interrupt
DMAC1	SINT9	48	13	DMA Channel #1 interrupt
DSPINT	SINT10	50	14	Interrupt from host
INT3/WDTINT†	SINT11	58	15	External interrupt #3 or Watchdog timer interrupt
RINT2	SINT12	60	17	McBSP #2 receive interrupt
XINT2	SINT13	68	18	McBSP #2 transmit interrupt
DMAC4	SINT14	70	21	DMA Channel #4 interrupt
DMAC5	SINT15	78	22	DMA Channel #5 interrupt
INT1	SINT16	80	4	External interrupt #1
XINT0	SINT17	88	8	McBSP #0 transmit interrupt
DMAC0	SINT18	90	12	DMA Channel #0 interrupt
–	SINT19	98	16	Software interrupt #19
DMAC2	SINT20	A0	19	DMA Channel #2 interrupt
DMAC3	SINT21	A8	20	DMA Channel #3 interrupt
TINT1	SINT22	B0	23	Timer #1 interrupt
IIC	SINT23	B8	24	I^2C interrupt
BERR	SINT24	C0	2	Bus Error interrupt
DLOG	SINT25	C8	25	Data Log interrupt
RTOS	SINT26	D0	26	Real-time Operating System interrupt
–	SINT27	D8	27	Software interrupt #27
–	SINT28	E0	28	Software interrupt #28
–	SINT29	E8	29	Software interrupt #29
–	SINT30	F0	30	Software interrupt #30
	SINT31	F8	31	Software interrupt #31

† WDTINT is only available when the Watchdog Timer is used as a general-purpose timer.

3.16.1 IFR and IER Registers

The Interrupt Enable Registers (IER0 and IER1) control which interrupts will be masked or enabled during normal operation. The Interrupt Flag Registers (IFR0 and IFR1) contain flags that indicate interrupts that are currently pending.

The Debug Interrupt Enable Registers (DBIER0 and DBIER1) are used only when the CPU is *halted* in the real-time emulation mode. If the CPU is *running* in real-time mode, the standard interrupt processing (IER0/1) is used and DBIER0/1 are ignored.

A maskable interrupt enabled in DBIER0/1 is defined as a time-critical interrupt. When the CPU is halted in the real-time mode, the only interrupts that are serviced are time-critical interrupts that are also enabled in an interrupt enable register (IER0 or IER1).

Write the DBIER0/1 to enable or disable time-critical interrupts. To enable an interrupt, set its corresponding bit. To disable an interrupt, clear its corresponding bit. Initialize these registers before using the real-time emulation mode.

A DSP hardware reset clears IFR0/1, IER0/1, and DBIER0/1 to 0. A software reset instruction clears IFR0/1 to 0 but does not affect IER0/1 and DBIER0/1.

The bit layouts of these registers for each interrupt are shown in Figure 3–48 and Figure 3–49. For more information on the IER, IFR, and DBIER registers, refer to the *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371).

15	14	13	12	11	10	9	8
DMAC5	DMAC4	Reserved	UART	INT3/ WDTINT‡	DSPINT	DMAC1	Reserved
R/W, 0	R/W, 0	R/W, 0†	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

7	6	5	4	3	2	1	0
XINT1	RINT1	RINT0	TINT0	INT2	INT0	Reserved	
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R, 0	

LEGEND: R = Read, W = Write, *n* = value at reset

† This bit must be kept zero when writing to IER0.
‡ The WDT interrupt pin must be connected to the $\overline{\text{INT3}}$ pin through the TSSR.

Figure 3–48. IFR0, IER0, DBIFR0, and DBIER0 Registers Layout

15					11	10	9	8
Reserved						RTOS	DLOG	BERR
R, 0						R/W, 0	R/W, 0	R/W, 0

7	6	5	4	3	2	1	0
I^2C	TINT1	DMAC3	DMAC2	INT4	DMAC0	XINT0	INT1
R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0	R/W, 0

LEGEND: R = Read, W = Write, *n* = value at reset

Figure 3–49. IFR1, IER1, DBIFR1, and DBIER1 Registers Layout

3.16.2 Interrupt Timing

The external interrupts ($\overline{\text{NMI}}$ and $\overline{\text{INT}}$) are synchronized to the CPU by way of a two-flip-flop synchronizer. The interrupt inputs are sampled on falling edges of the CPU clock. A sequence on the interrupt pin of 1–0–0–0 on consecutive cycles is required for an interrupt to be detected. Therefore, the minimum low pulse duration on the external interrupts on the 5501 is three CPU clock periods.

4 Documentation Support

Extensive documentation supports all TMS320™ DSP family of devices from product announcement through applications development. The following types of documentation are available to support the design and use of the TMS320C5000™ platform of DSPs:

- *TMS320C55x™ DSP Functional Overview* (literature number SPRU312)
- Device-specific data sheets
- Complete user's guides
- Development support tools
- Hardware and software application reports

TMS320C55x reference documentation that includes, but is not limited to, the following:

- *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371)
- *TMS320C55x DSP Mnemonic Instruction Set Reference Guide* (literature number SPRU374)
- *TMS320C55x DSP Algebraic Instruction Set Reference Guide* (literature number SPRU375)
- *TMS320C55x DSP Programmer's Guide* (literature number SPRU376)
- *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317)

The reference guides describe in detail the TMS320C55x™ DSP products currently available and the hardware and software applications, including algorithms, for fixed-point TMS320 DSP family of devices.

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support digital signal processing research and education. The TMS320 DSP newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320 DSP customers on product information.

Information regarding TI DSP products is also available on the Worldwide Web at *http://www.ti.com* uniform resource locator (URL).

5 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320™ DSP devices and support tools. Each TMS320™ DSP commercial family member has one of three prefixes: TMX, TMP, or TMS. Texas Instruments recommends two of three possible prefix designators for support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

TMX Experimental device that is not necessarily representative of the final device's electrical specifications

TMP Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

TMS Fully qualified production device

Support tool development evolutionary flow:

TMDX Development-support product that has not yet completed Texas Instruments internal qualification testing.

TMDS Fully qualified development-support product

TMX and TMP devices and TMDX development-support tools are shipped with appropriate disclaimers describing their limitations and intended uses. Experimental devices (TMX) may not be representative of a final product and Texas Instruments reserves the right to change or discontinue these products without notice.

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

6 Electrical Specifications

This section provides the absolute maximum ratings and the recommended operating conditions for the TMS320VC5501 DSP.

All electrical and switching characteristics in this data manual are valid over the recommended operating conditions unless otherwise specified.

6.1 Absolute Maximum Ratings

The list of absolute maximum ratings are specified over operating case temperature. Stresses beyond those listed under Section 6.2, Electrical Specifications, may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under Section 6.3, Recommended Operating Conditions, is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability. All supply voltage values (core and I/O) are with respect to V_{SS}. Figure 6–1 provides the test load circuit values for a 3.3-V device. Measured timing information contained in this data manual is based on the test load setup and conditions shown in Figure 6–1.

6.2 Electrical Specifications

This section provides the absolute maximum ratings for the TMS320VC5501 DSP.

Supply voltage I/O range, DV_{DD} . . . . . – 0.3 V to 4.0 V
Supply voltage core range, CV_{DD} . . . . . – 0.3 V to 2.0 V
Input voltage range, V_I . . . . . – 0.3 V to 4.5 V
Output voltage range, V_o . . . . . – 0.3 V to 4.5 V
Operating case temperature range, T_C . . . . . –40°C to 85°C
Storage temperature range T_{stg} . . . . . – 55°C to 150°C

6.3 Recommended Operating Conditions

This section provides the recommended operating conditions for the TMS320VC5501 DSP.

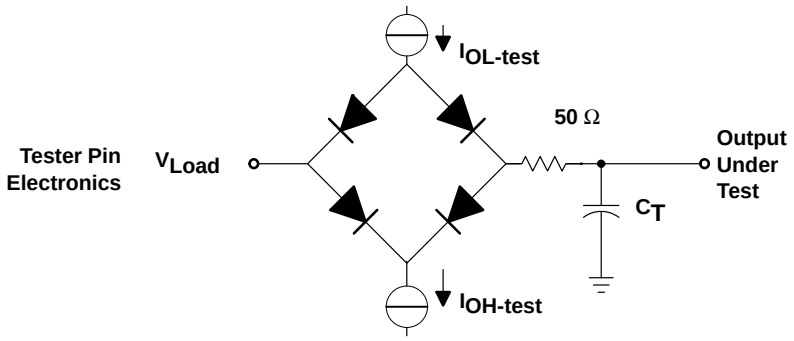
			MIN	NOM	MAX	UNIT
DV_{DD}	Device supply voltage, I/O		2.7	3.3	3.6	V
CV_{DD}	Device supply voltage, core		1.20	1.26	1.32	V
PV_{DD}	Device supply voltage, PLL		2.7	3.3	3.6	V
V_{SS}	Supply voltage, GND			0		V
V_{IH}	High-level input voltage, I/O	Hysteresis inputs DV_{DD} = 2.7 – 3.6 V	2.2		DV_{DD} + 0.3	V
		All other inputs	2		DV_{DD} + 0.3	
V_{IL}	Low-level input voltage, I/O	Hysteresis inputs DV_{DD} = 2.7 – 3.6 V	–0.3		0.8	V
		All other inputs	–0.3		0.8	
I_{OH}	High-level output current	All outputs			– 300	µA
I_{OL}	Low-level output current	All outputs			1.5	mA
T_C	Operating case temperature		–40		85	°C

6.4 Electrical Characteristics Over Recommended Operating Case Temperature Range (Unless Otherwise Noted)

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		DV_{DD} = 2.7 – 3.0 V, I_{OH} = MAX		2.2			V
			DV_{DD} = 3.3 ± 0.3 V, I_{OH} = MAX		2.4			
V_{OL}	Low-level output voltage		I_{OL} = MAX				0.4	V
I_{IZ}	Input current for outputs in high impedance	Output-only or input/output pins with bus holders	Bus holders enabled DV_{DD} = MAX, V_O = V_{SS} to DV_{DD}		– 300		300	µA
		All other output-only or input/output pins	DV_{DD} = MAX, –0.3 V < V_I < 4.5 V		– 5		5	
I_I	Input current	Input pins with internal pulldown	DV_{DD} = MAX, –0.3 V < V_I < 4.5 V		– 5		300	µA
		X2/CLKIN with oscillator on	DV_{DD} = MAX, –0.3 V < V_I < 4.5 V		TBD		TBD	
		X2/CLKIN with oscillator off	DV_{DD} = MAX, –0.3 V < V_I < 4.5 V		TBD		TBD	
		Input pins with internal pullup	Pullup enabled DV_{DD} = MAX, –0.3 V < V_I < 4.5 V		– 300		5	
		All other input-only pins	DV_{DD} = MAX, –0.3 V < V_I < 4.5 V		– 5		5	
I_{DDC}	CV_{DD} supply current, CPU + internal memory access†		CV_{DD} = Nominal CPU clock = 300 MHz T_C = 25°C			TBD		mA
I_{DDD}	DV_{DD} supply current, pins active‡		DV_{DD} = Nominal CPU clock = 300 MHz T_C = 25°C			TBD		mA
I_{DDP}	PV_{DD} supply current, standby Only CLKGEN domain enabled, PLL enabled		PV_{DD} = Nominal 30-MHz clock input, APLL mode = x10 T_C = 25°C			TBD		mA
I_{DDC}	CV_{DD} supply current, standby All domains idled		CV_{DD} = Nominal input clock stopped	25°C		TBD		mA
				85°C		TBD		
I_{DDD}	DV_{DD} supply current, standby All domains idled		DV_{DD} = Nominal no pin activity T_C = 25°C			TBD		µA
C_i	Input capacitance					3		pF
C_o	Output capacitance					3		pF

† Test Condition: CPU executing 75% Dual-MAC / 25% ADD with moderate data bus activity (table of sine values). CPU and CLKGEN domains are active. All other domains are idled. The APLL is enabled.

‡ Test Condition: One word of a table of 16-bit sine values is written to the EMIF each microsecond (16 Mbps). Each EMIF output pin is connected to a 10-pF load capacitance.



Where: $I_{OL-test}$ = 1.5 mA (all outputs)
$I_{OH-test}$ = 300 µA (all outputs)
V_{Load} = 1.8 V
C_T = 15 pF typical load circuit capacitance

Figure 6–1. 3.3-V Test Load Circuit

6.5 Package Thermal Resistance Characteristics

Table 6–1 provide the thermal resistance characteristics for the recommended package types used on the TMS320VC5501 DSP.

Table 6–1. Thermal Resistance Characteristics

PARAMETER	GGW PACKAGE	PGF PACKAGE	UNIT	BOARD TYPE†
$R_{\theta JC}$	TBD	TBD	°C/W	2s JEDEC Test Card

† Board types are as defined by JEDEC. Reference JEDEC Standard JESD51–9, Test Boards for Area Array Surface Mount Package Thermal Measurements.

6.6 Timing Parameter Symbology

Timing parameter symbols used in the timing requirements and switching characteristics tables are created in accordance with JEDEC Standard 100. To shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

Lowercase subscripts and their meanings:

a	access time
c	cycle time (period)
d	delay time
dis	disable time
en	enable time
f	fall time
h	hold time
r	rise time
su	setup time
t	transition time
v	valid time
w	pulse duration (width)
X	Unknown, changing, or don't care level

Letters and symbols and their meanings:

H	High
L	Low
V	Valid
Z	High impedance

6.7 Clock Options

This section provides the timing requirements and switching characteristics for the various clock options available on the 5501.

6.7.1 Clock Generation in Bypass Mode (APLL Disabled)

Table 6–2 and Table 6–3 assume testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 6–2).

Table 6–2. CLKIN in Bypass Mode Timing Requirements

NO.			MIN	MAX	UNIT
C7	$t_{c(CI)}$	Cycle time, CLKIN	3.33	†	ns
C8	$t_{f(CI)}$	Fall time, CLKIN		6	ns
C9	$t_{r(CI)}$	Rise time, CLKIN		6	ns

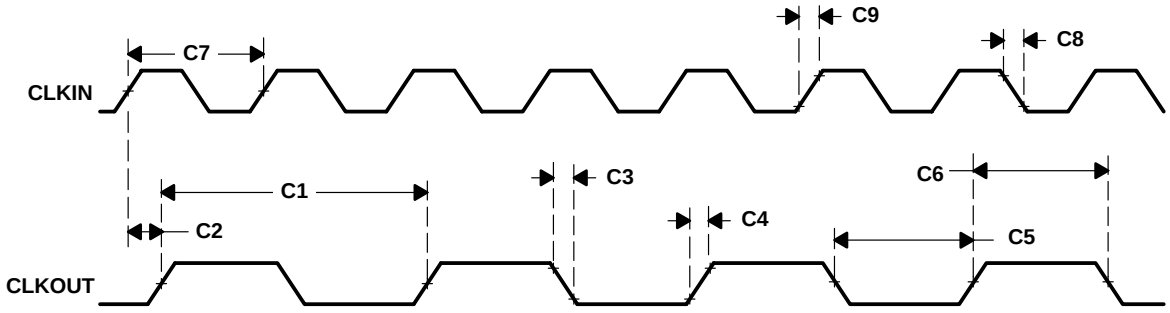
† This device utilizes a fully static design and therefore can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz.

Table 6–3. CLKOUT in Bypass Mode Switching Characteristics

NO.	PARAMETER		MIN	TYP	MAX	UNIT
C1	$t_{c(CO)}$	Cycle time, CLKOUT	3.33‡	$t_{c(CI)}$	†	ns
C2	$t_{d(CIH-CO)}$	Delay time, CLKIN high to CLKOUT high/low	1	2	4	ns
C3	$t_{f(CO)}$	Fall time, CLKOUT		1		ns
C4	$t_{r(CO)}$	Rise time, CLKOUT		1		ns
C5	$t_{w(COL)}$	Pulse duration, CLKOUT low	H–1	H	H+1	ns
C6	$t_{w(COH)}$	Pulse duration, CLKOUT high	H–1	H	H+1	ns

† This device utilizes a fully static design and therefore can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz.

‡ It is recommended that the lock mode (APLL synthesis enabled) clocking option be used for maximum frequency operation.



NOTE: The relationship of CLKIN to CLKOUT depends on the divide factor chosen. The waveform relationship shown in Figure 6–2 is intended to illustrate the timing parameters only and may differ based on configuration.

Figure 6–2. Bypass Mode Clock Timing

PRODUCT PREVIEW

6.7.2 Clock Generation in PLL Mode

The frequency of the reference clock provided at the CLKIN pin can be multiplied by a synthesis factor of N to generate the internal CPU clock cycle. The synthesis factor is determined by:

$$N = \frac{M}{D_0}$$

where: M = the multiply factor set in the PLLM field of the PLL Multiplier Control Register (PLLM)
D_0 = the divide factor set in the PLLDIV0 field of the PLL Divider 0 Register (PLLDIV0)

Valid values for M are (multiply by) 1 to 16. Valid values for D_0 are (divide by) 1 – 32.

For detailed information on clock generation configuration, see the *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317).

Table 6–4 and Table 6–5 assume testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 6–3).

Table 6–4. CLKIN in Lock Mode Timing Requirements

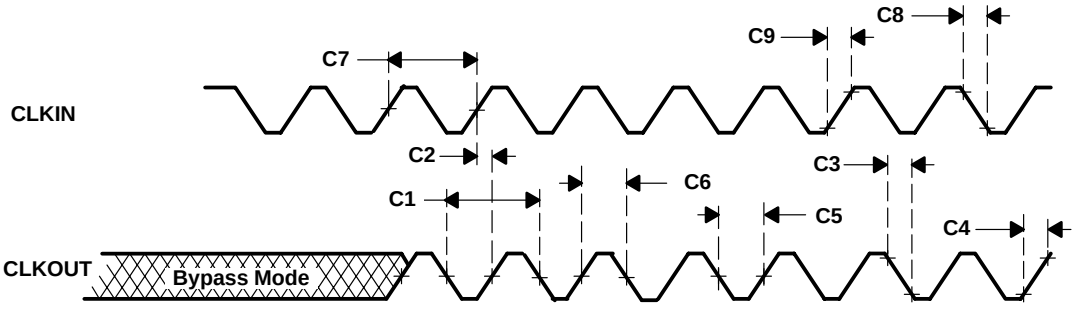
NO.					MIN	MAX	UNIT
C7	$t_{c(CI)}$	Cycle time, CLKIN	APLL synthesis enabled	Oscillator disabled	40†	400	ns
				Oscillator enabled	50	200	ns
C8	$t_{f(CI)}$	Fall time, CLKIN				6	ns
C9	$t_{r(CI)}$	Rise time, CLKIN				6	ns

† The clock frequency synthesis factor and minimum CLKIN cycle time should be chosen such that the resulting CLKOUT cycle time is within the specified range [$t_{c(CO)}$].

Table 6–5. CLKOUT in Lock Mode Switching Characteristics

NO.	PARAMETER		MIN	TYP	MAX	UNIT
C1	$t_{c(CO)}$	Cycle time, CLKOUT	3.33	$t_{c(CI)}/N$‡		ns
C2	$t_{d(CI-CO)}$	Delay time, CLKIN high/low to CLKOUT high/low	1	2	4	ns
C3	$t_{f(CO)}$	Fall time, CLKOUT		1		ns
C4	$t_{r(CO)}$	Rise time, CLKOUT		1		ns
C5	$t_{w(COL)}$	Pulse duration, CLKOUT low	H–1	H	H+1	ns
C6	$t_{w(COH)}$	Pulse duration, CLKOUT high	H–1	H	H+1	ns

‡ N = Clock frequency synthesis factor



NOTE: The waveform relationship of CLKIN to CLKOUT depends on the multiply and divide factors chosen. The waveform relationship shown in Figure 6–3 is intended to illustrate the timing parameters only and may differ based on configuration.

Figure 6–3. External Multiply-by-N Clock Timing

6.7.3 EMIF Clock Options

Table 6–6 through Table 6–8 assume testing over recommended operating conditions (see Figure 6–4 through Figure 6–6).

Table 6–6. EMIF Timing Requirements for ECLKIN†‡

NO.			MIN	MAX	UNIT
E1	$t_{c(EKI)}$	Cycle time, ECLKIN	10	16P	ns
E2	$t_{w(EKIH)}$	Pulse duration, ECLKIN high	3.38		ns
E3	$t_{w(EKIL)}$	Pulse duration, ECLKIN low	3.38		ns
E4	$t_{t(EKI)}$	Transition time, ECLKIN		2	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.
‡ The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.

Table 6–7. EMIF Switching Characteristics for ECLKOUT1§¶#

NO.	PARAMETER		MIN	MAX	UNIT
E11	$t_{c(EKO1)}$	Cycle time, ECLKOUT1	E – 0.7	E + 0.7	ns
E22	$t_{w(EKO1H)}$	Pulse duration, ECLKOUT1 high	EH – 0.7	EH + 0.7	ns
E33	$t_{w(EKO1L)}$	Pulse duration, ECLKOUT1 low	EL – 0.7	EL + 0.7	ns
E44	$t_{t(EKO1)}$	Transition time, ECLKOUT1		1	ns
E55	$t_{d(EKIH-EKO1H)}$	Delay time, ECLKIN high to ECLKOUT1 high	3	8	ns
E66	$t_{d(EKIL-EKO1L)}$	Delay time, ECLKIN low to ECLKOUT1 low	3	8	ns

§ The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
¶ E = the EMIF input clock (CPU clock, CPU/2 clock, or CPU/4 clock) period in ns for EMIF.
# EH is the high period of E (EMIF input clock period) in ns and EL is the low period of E (EMIF input clock period) in ns for EMIF.

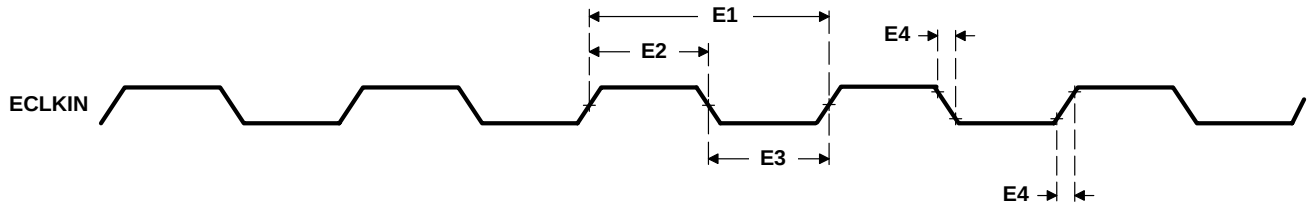


Figure 6–4. ECLKIN Timing for EMIF

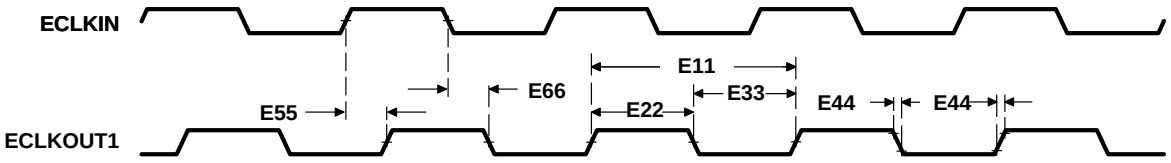


Figure 6–5. ECLKOUT1 Timing for EMIF Module

Table 6–8. EMIF Switching Characteristics for ECLKOUT2†‡

NO.	PARAMETER		MIN	MAX	UNIT
E1	$t_{c(EKO2)}$	Cycle time, ECLKOUT2	NE – 0.7	NE + 0.7	ns
E2	$t_{w(EKO2H)}$	Pulse duration, ECLKOUT2 high	0.5NE – 0.7	0.5NE + 0.7	ns
E3	$t_{w(EKO2L)}$	Pulse duration, ECLKOUT2 low	0.5NE – 0.7	0.5NE + 0.7	ns
E4	$t_{t(EKO2)}$	Transition time, ECLKOUT2		1	ns
E5	$t_{d(EKIH-EKO2H)}$	Delay time, ECLKIN high to ECLKOUT2 high	3	8	ns
E6	$t_{d(EKIH-EKO2L)}$	Delay time, ECLKIN high to ECLKOUT2 low	3	8	ns

† The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
‡ E = the EMIF input clock (CPU clock, CPU/2 clock, or CPU/4 clock) period in ns for EMIF.
N = the EMIF input clock divider; N = 1, 2, or 4.

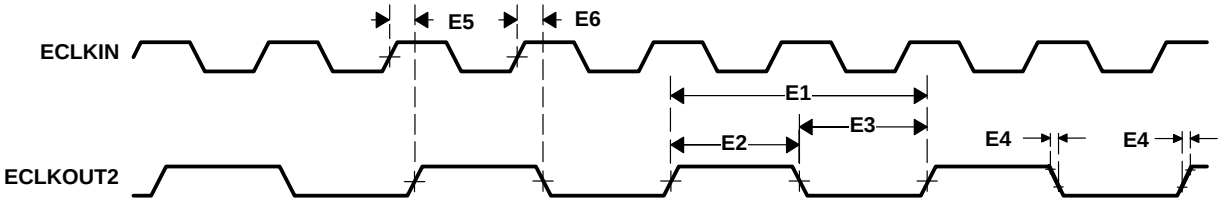


Figure 6–6. ECLKOUT2 Timing for the EMIF Module

PRODUCT PREVIEW

6.8 EMIF Asynchronous Memory Timing

Table 6–9 and Table 6–10 assume testing over recommended operating conditions (see Figure 6–7 and Figure 6–8).

Table 6–9. EMIF Asynchronous Memory Cycle Timing Requirements for ECLKIN†‡

NO.			MIN	MAX	UNIT
E3	$t_{su(EDV-AREH)}$	Setup time, Dx valid before $\overline{ARE}$ high	6		ns
E4	$t_{h(AREH-EDV)}$	Hold time, Dx valid after $\overline{ARE}$ high	1		ns
E6	$t_{su(ARDY-EKO1H)}$	Setup time, ARDY valid before ECLKOUT1 high	3		ns
E7	$t_{h(EKO1H-ARDY)}$	Hold time, ARDY valid after ECLKOUT1 high	1		ns

† To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. The ARDY signal is recognized in the cycle for which the setup and hold time is met. To use ARDY as an asynchronous input, the pulse width of the ARDY signal should be wide enough (e.g., pulse width = 2E) to ensure setup and hold time is met.

‡ RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed via the EMIF CE space control registers.

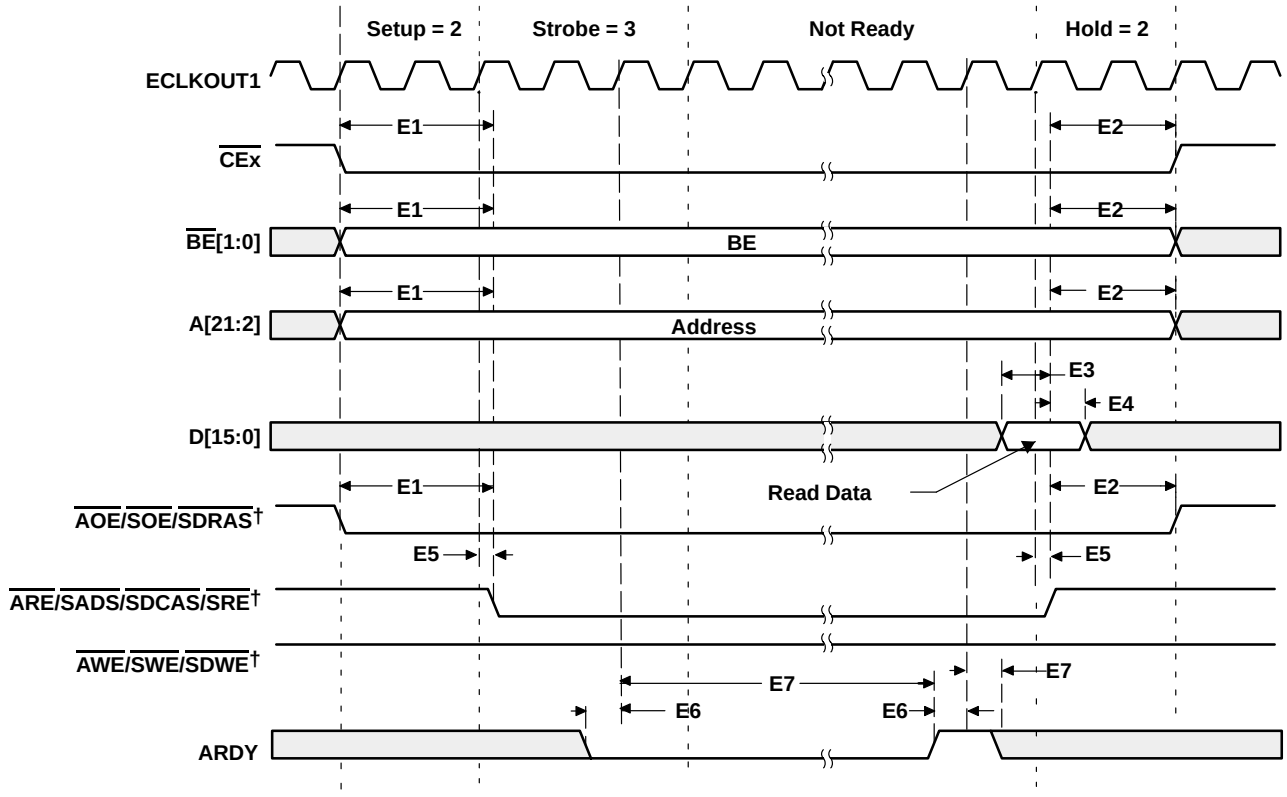
Table 6–10. EMIF Asynchronous Memory Cycle Switching Characteristics for ECLKOUT1‡§¶

NO.	PARAMETER		MIN	MAX	UNIT
E1	$t_{osu(SELV-AREL)}$	Output setup time, select signals valid to $\overline{ARE}$ low	RS * E – 1.5		ns
E2	$t_{oh(AREH-SELIV)}$	Output hold time, $\overline{ARE}$ high to select signals invalid	RH * E – 1.5		ns
E5	$t_{d(EKO1H-AREV)}$	Delay time, ECLKOUT1 high to $\overline{ARE}$ valid	1.5	5	ns
E8	$t_{osu(SELV-AWEL)}$	Output setup time, select signals valid to $\overline{AWE}$ low	WS * E – 1.5		ns
E9	$t_{oh(AWEH-SELIV)}$	Output hold time, $\overline{AWE}$ high to select signals invalid	WH * E – 1.5		ns
E10	$t_{d(EKO1H-AWEV)}$	Delay time, ECLKOUT1 high to $\overline{AWE}$ valid	1.5	5	ns

‡ RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed via the EMIF CE space control registers.

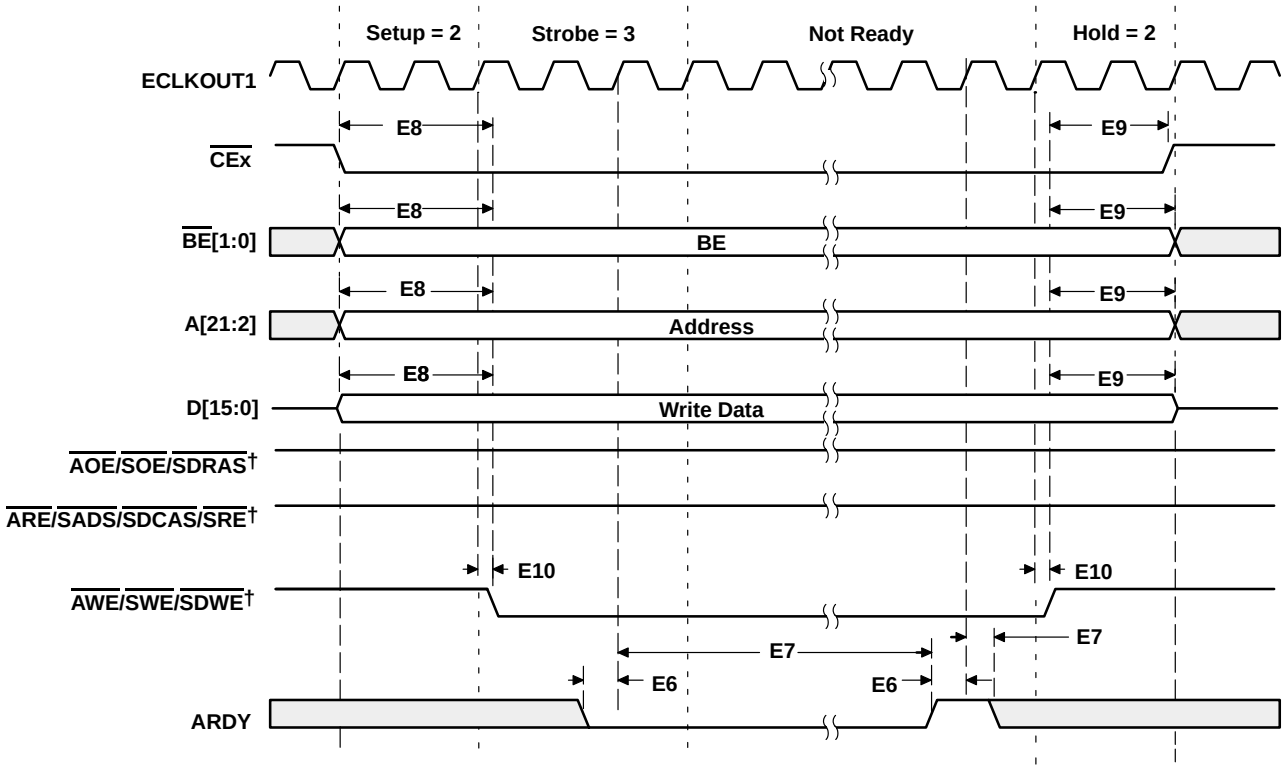
§ E = ECLKOUT1 period in ns for EMIF.

¶ Select signals for EMIF include: $\overline{CE}$x, $\overline{BE}$[1:0], A[21:2], and $\overline{AOE}$; and for EMIF writes, include D[15:0].



† AOE/SOE/SDRAS, ARE/SADS/SDCAS/SRE, and AWE/SWE/SDWE operate as AOE (identified under select signals), ARE, and AWE, respectively, during asynchronous memory accesses.

Figure 6–7. Asynchronous Memory Read Timing for EMIF†



† AOE/SOE/SDRAS, ARE/SADS/SDCAS/SRE, and AWE/SWE/SDWE operate as AOE (identified under select signals), ARE, and AWE, respectively, during asynchronous memory accesses.

Figure 6–8. Asynchronous Memory Write Timing for EMIF†

6.9 Programmable Synchronous Interface Timing

Table 6–11 and Table 6–12 assume testing over recommended operating conditions (see Figure 6–9 through Figure 6–11).

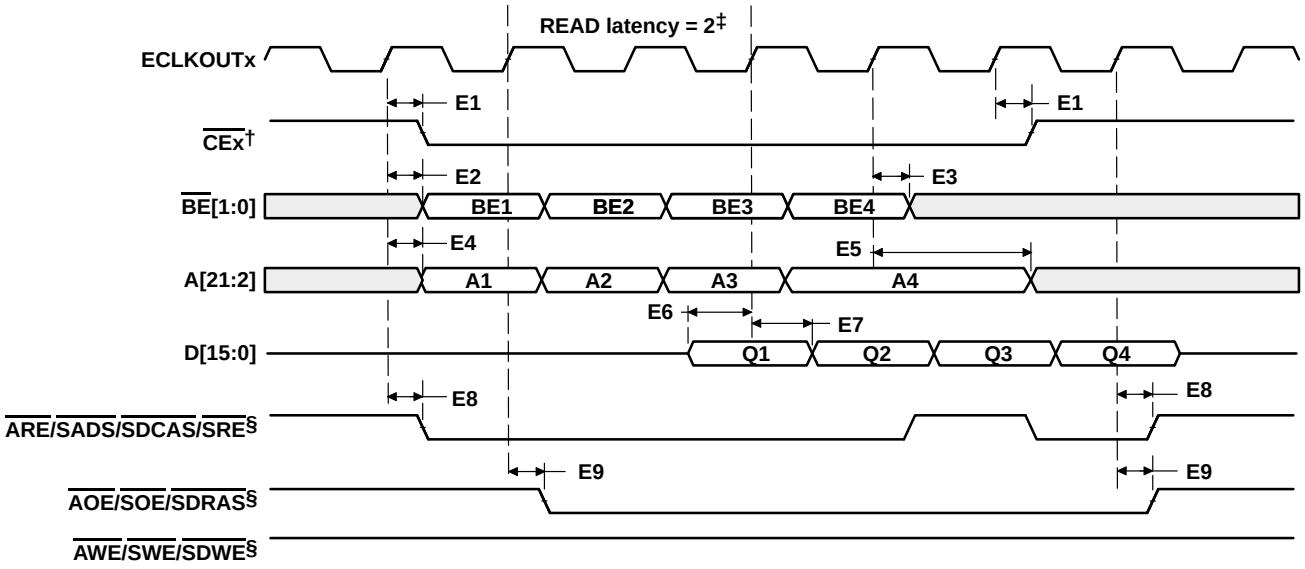
Table 6–11. EMIF Programmable Synchronous Interface Timing Requirements

NO.			MIN	MAX	UNIT
E6	$t_{su(EDV-EKOxH)}$	Setup time, read Dx valid before ECLKOUTx high	2		ns
E7	$t_{h(EKOxH-EDV)}$	Hold time, read Dx valid after ECLKOUTx high	1.5		ns

Table 6–12. EMIF Programmable Synchronous Interface Switching Characteristics†

NO.	PARAMETER		MIN	MAX	UNIT
E1	$t_{d(EKOxH-CEV)}$	Delay time, ECLKOUTx high to $\overline{CEx}$ valid	1	5	ns
E2	$t_{d(EKOxH-BEV)}$	Delay time, ECLKOUTx high to $\overline{BEx}$ valid		5	ns
E3	$t_{d(EKOxH-BEIV)}$	Delay time, ECLKOUTx high to $\overline{BEx}$ invalid	1		ns
E4	$t_{d(EKOxH-EAV)}$	Delay time, ECLKOUTx high to Ax valid		5	ns
E5	$t_{d(EKOxH-EAIV)}$	Delay time, ECLKOUTx high to Ax invalid	1		ns
E8	$t_{d(EKOxH-ADSV)}$	Delay time, ECLKOUTx high to $\overline{SADS}/\overline{SRE}$ valid	1	5	ns
E9	$t_{d(EKOxH-OEV)}$	Delay time, ECLKOUTx high to, $\overline{SOE}$ valid	1	5	ns
E10	$t_{d(EKOxH-EDV)}$	Delay time, ECLKOUTx high to Dx valid		5	ns
E11	$t_{d(EKOxH-EDIV)}$	Delay time, ECLKOUTx high to Dx invalid	1		ns
E12	$t_{d(EKOxH-WEV)}$	Delay time, ECLKOUTx high to $\overline{SWE}$ valid	1	5	ns

† The following parameters are programmable via the EMIF CE Secondary Control Registers (CEx_SC1, CEx_SC2):
- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{CEx}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{CEx}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, $\overline{CEx}$ is active when $\overline{SOE}$ is active (CEEXT = 1).
- Function of $\overline{SADS}/\overline{SRE}$ (RENEN): For standard SBSRAM or ZBT SRAM interface, $\overline{SADS}/\overline{SRE}$ acts as $\overline{SADS}$ with deselect cycles (RENEN = 0). For FIFO interface, $\overline{SADS}/\overline{SRE}$ acts as $\overline{SRE}$ with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2



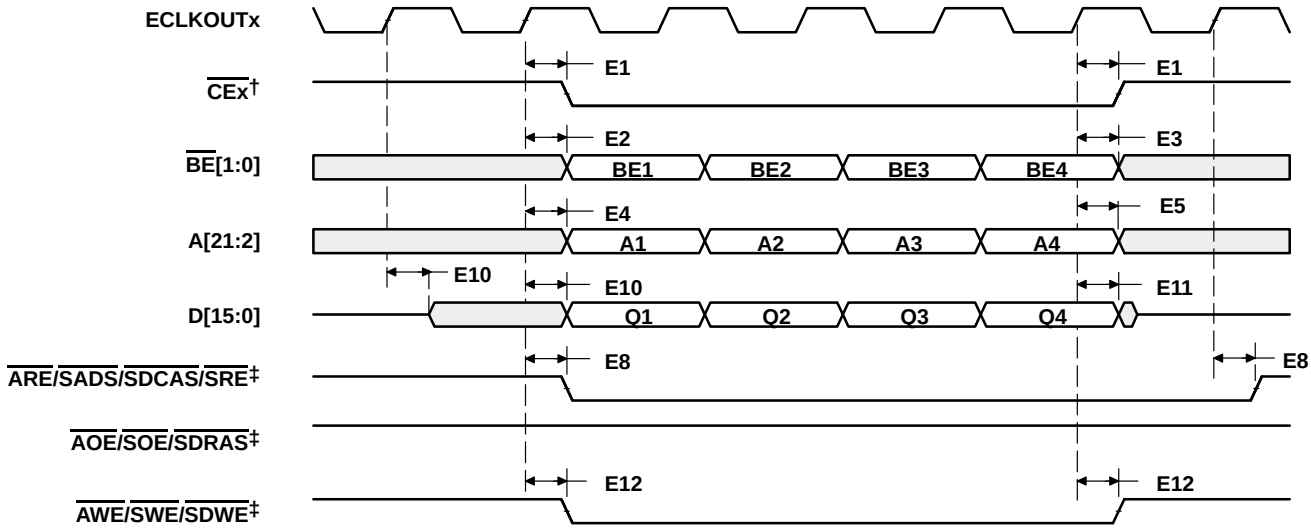
† The read latency and the length of $\overline{\text{CEx}}$ assertion are programmable via the SYNCRL and CEEXT fields, respectively, in the EMIF CE Secondary Control Registers (CEx_SC1, CEx_SC2). In the figure, SYNCRL = 2 and CEEXT = 0.

‡ The following parameters are programmable via the EMIF CE Secondary Control Registers (CEx_SC1, CEx_SC2):

- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{\text{CEx}}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{\text{CEx}}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, $\overline{\text{CEx}}$ is active when $\overline{\text{SOE}}$ is active (CEEXT = 1).
- Function of SADS/SRE (RENEN): For standard SBSRAM or ZBT SRAM interface, $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ acts as $\overline{\text{SADS}}$ with deselect cycles (RENEN = 0). For FIFO interface, $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ acts as $\overline{\text{SRE}}$ with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

§ $\overline{\text{ARE}}$/$\overline{\text{SADS}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SRE}}$, $\overline{\text{AOE}}$/$\overline{\text{SOE}}$/$\overline{\text{SDRAS}}$, and $\overline{\text{AWE}}$/$\overline{\text{SWE}}$/$\overline{\text{SDWE}}$ operate as $\overline{\text{SADS}}$/$\overline{\text{SRE}}$, $\overline{\text{SOE}}$, and $\overline{\text{SWE}}$, respectively, during programmable synchronous interface accesses.

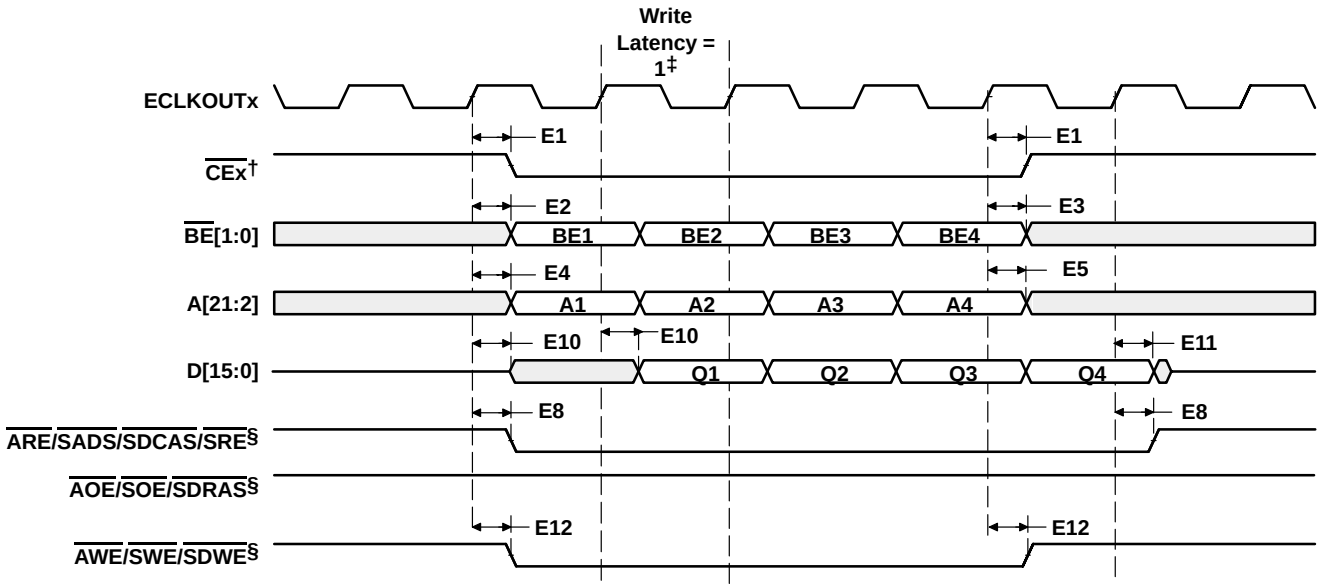
Figure 6–9. Programmable Synchronous Interface Read Timing for EMIF (With Read Latency = 2)



† The write latency and the length of CEx assertion are programmable via the SYNCWL and CEEXT fields, respectively, in the EMIF CE Secondary Control Registers (CEx_SC1, CEx_SC2). In this figure, SYNCWL = 0 and CEEXT = 0.

‡ ARE/SADS/SDCAS/SRE, AOE/SOE/SDRAS, and AWE/SWE/SDWE operate as SADS/SRE, SOE, and SWE, respectively, during programmable synchronous interface accesses.

Figure 6–10. Programmable Synchronous Interface Write Timing for EMIF (With Write Latency = 0)



† The write latency and the length of $\overline{CEx}$ assertion are programmable via the SYNCWL and CEEXT fields, respectively, in the EMIF CE Secondary Control Registers (CEx_SC1, CEx_SC2). In this figure, SYNCWL = 1 and CEEXT = 0.

‡ The following parameters are programmable via the EMIF CE Secondary Control Registers (CEx_SC1, CEx_SC2):

- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{CEx}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{CEx}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, $\overline{CEx}$ is active when $\overline{SOE}$ is active (CEEXT = 1).
- Function of $\overline{SADS}$/$\overline{SRE}$ (RENEN): For standard SBSRAM or ZBT SRAM interface, $\overline{SADS}$/$\overline{SRE}$ acts as $\overline{SADS}$ with deselect cycles (RENEN = 0). For FIFO interface, $\overline{SADS}$/$\overline{SRE}$ acts as $\overline{SRE}$ with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

§ $\overline{ARE}$/$\overline{SADS}$/$\overline{SDCAS}$/$\overline{SRE}$, $\overline{AOE}$/$\overline{SOE}$/$\overline{SDRAS}$, and $\overline{AWE}$/$\overline{SWE}$/$\overline{SDWE}$ operate as $\overline{SADS}$/$\overline{SRE}$, $\overline{SOE}$, and $\overline{SWE}$, respectively, during programmable synchronous interface accesses.

Figure 6–11. Programmable Synchronous Interface Write Timing for EMIF (With Write Latency = 1)

6.10 Synchronous DRAM Timing

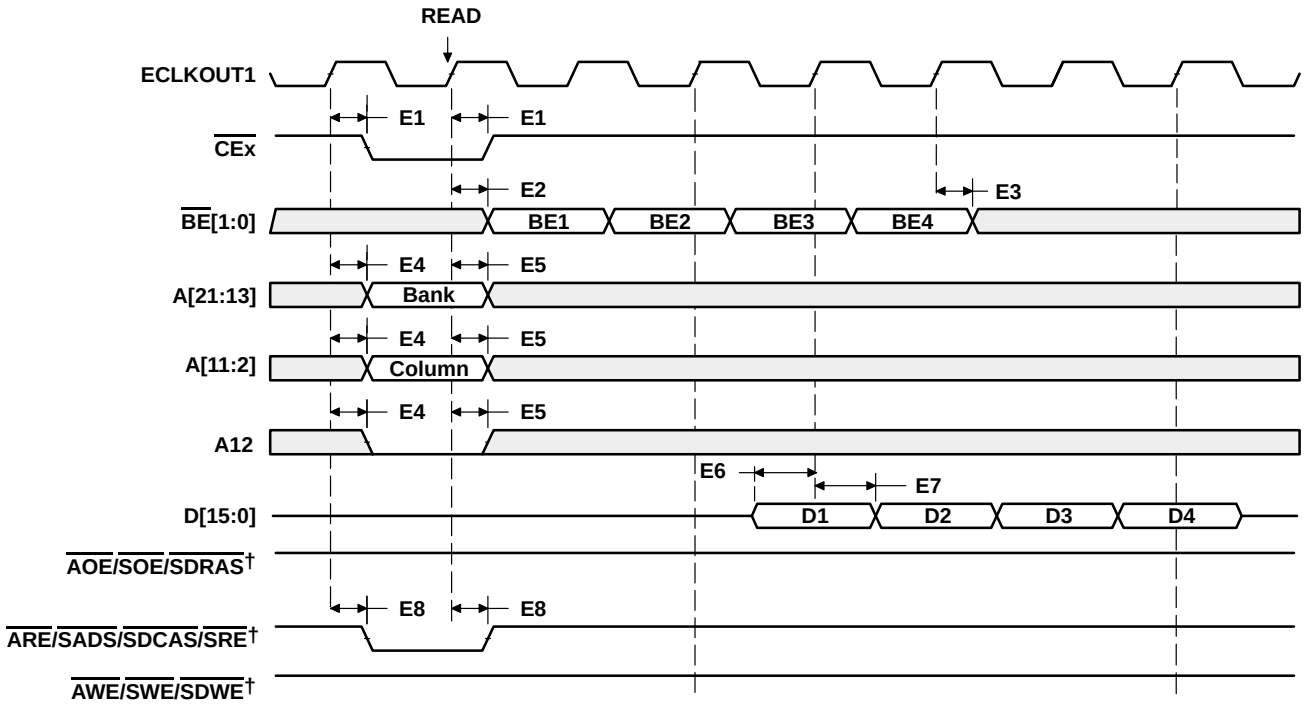
Table 6–13 and Table 6–14 assume testing over recommended operating conditions (see Figure 6–12 through Figure 6–19).

Table 6–13. EMIF Synchronous DRAM Cycle Timing Requirements

NO.			MIN	MAX	UNIT
E6	$t_{su(EDV-EKO1H)}$	Setup time, read Dx valid before ECLKOUT1 high	0.5		ns
E7	$t_{h(EKO1H-EDV)}$	Hold time, read Dx valid after ECLKOUT1 high	2		ns

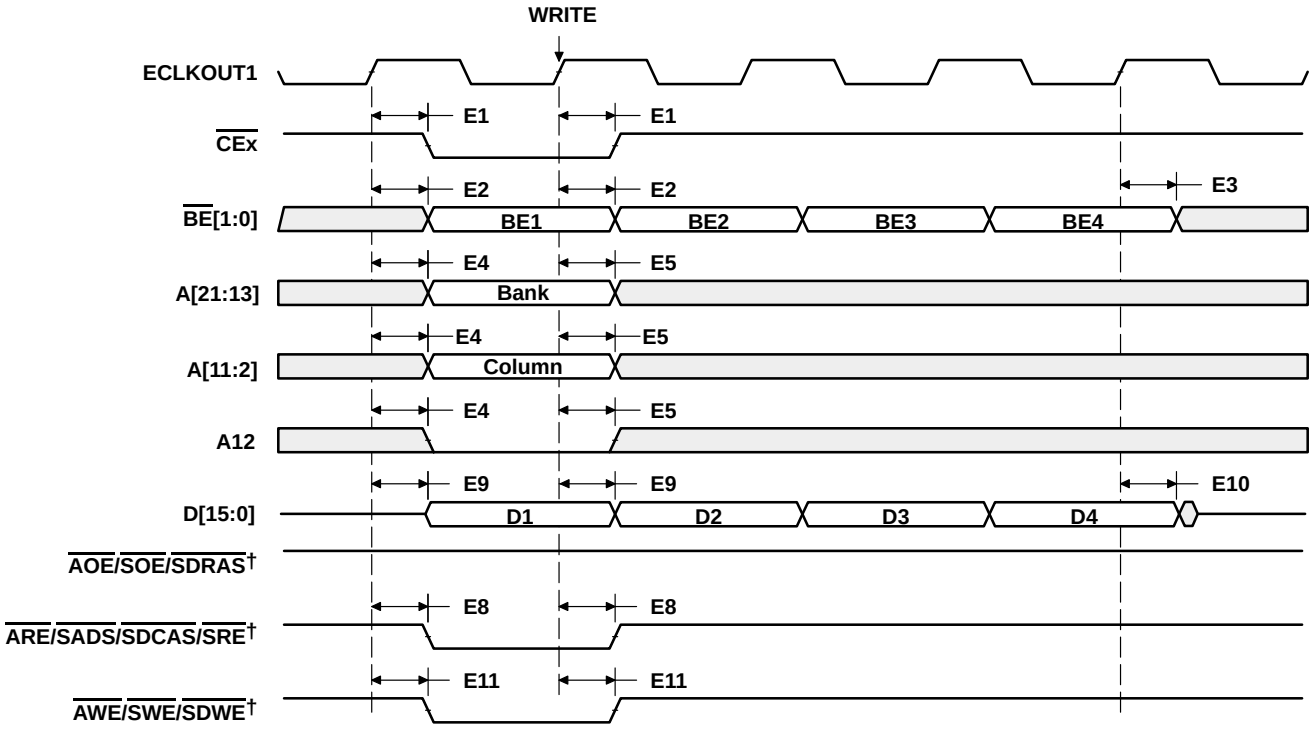
Table 6–14. EMIF Synchronous DRAM Cycle Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
E1	$t_{d(EKO1H-CEV)}$	Delay time, ECLKOUT1 high to $\overline{CEx}$ valid/invalid	1	5	ns
E2	$t_{d(EKO1H-BEV)}$	Delay time, ECLKOUT1 high to $\overline{BEx}$ valid		5	ns
E3	$t_{d(EKO1H-BEIV)}$	Delay time, ECLKOUT1 high to $\overline{BEx}$ invalid	1		ns
E4	$t_{d(EKO1H-EAV)}$	Delay time, ECLKOUT1 high to Ax valid		5	ns
E5	$t_{d(EKO1H-EAIV)}$	Delay time, ECLKOUT1 high to Ax invalid	1		ns
E8	$t_{d(EKO1H-CASV)}$	Delay time, ECLKOUT1 high to $\overline{SDCAS}$ valid	1	5	ns
E9	$t_{d(EKO1H-EDV)}$	Delay time, ECLKOUT1 high to Dx valid		5	ns
E10	$t_{d(EKO1H-EDIV)}$	Delay time, ECLKOUT1 high to Dx invalid	1		ns
E11	$t_{d(EKO1H-WEV)}$	Delay time, ECLKOUT1 high to $\overline{SDWE}$ valid	1	5	ns
E12	$t_{d(EKO1H-RASV)}$	Delay time, ECLKOUT1 high to $\overline{SDRAS}$ valid	1	5	ns
E13	$t_{d(EKO1H-CKEV)}$	Delay time, ECLKOUT1 high to SDCKE valid	1	5	ns



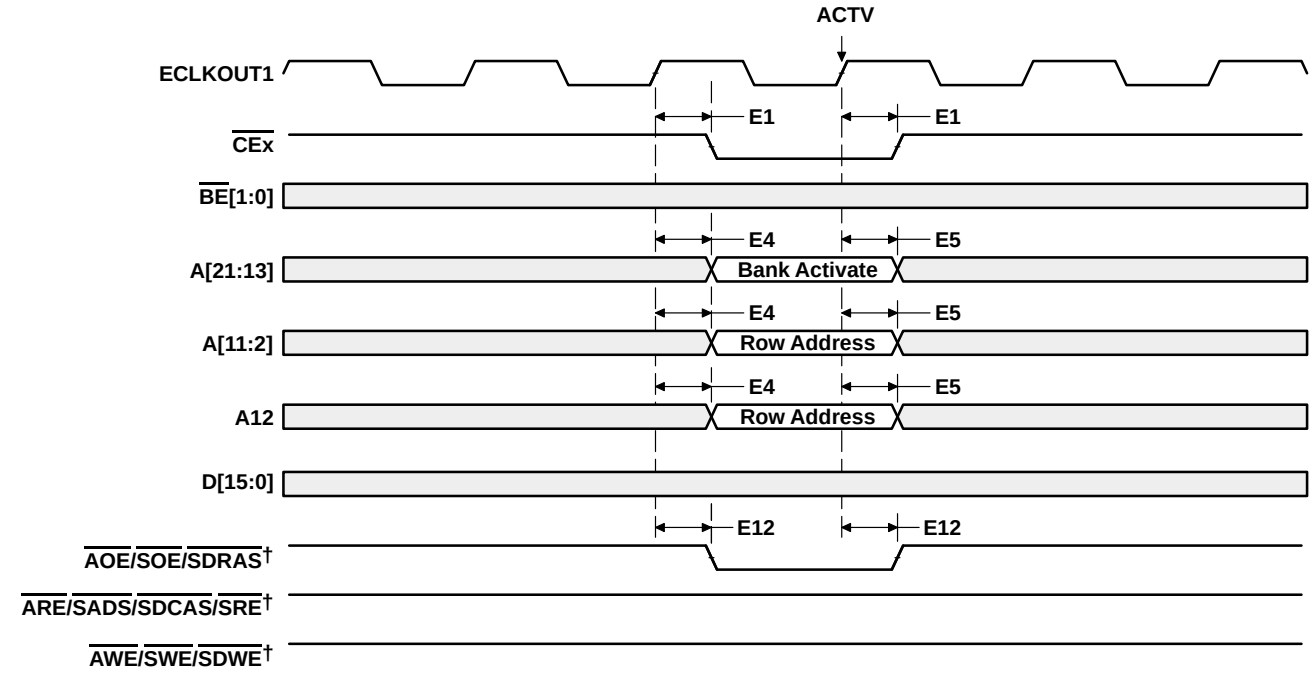
† $\overline{ARE}/\overline{SADS}/\overline{SDCAS}/\overline{SRE}$, $\overline{AWE}/\overline{SWE}/\overline{SDWE}$, and $\overline{AOE}/\overline{SOE}/\overline{SDRAS}$ operate as $\overline{SDCAS}$, $\overline{SDWE}$, and $\overline{SDRAS}$, respectively, during SDRAM accesses.

Figure 6–12. SDRAM Read Command (CAS Latency 3) for EMIF



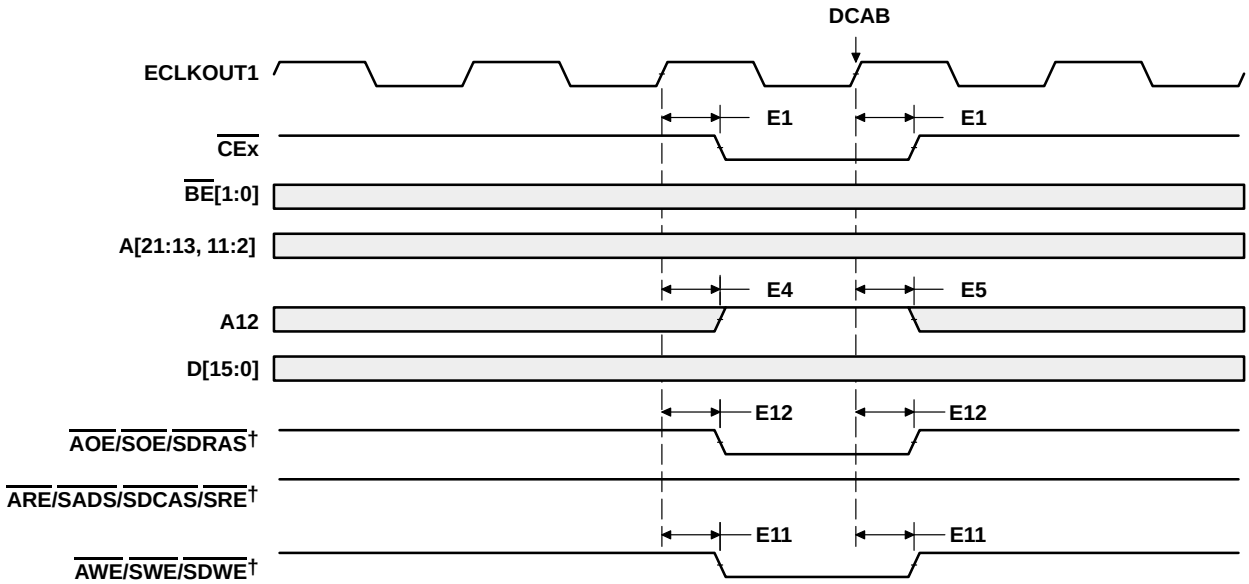
† ARE/SADS/SDCAS/SRE, AWE/SWE/SDWE, and AOE/SOE/SDRAS operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 6–13. SDRAM Write Command for EMIF



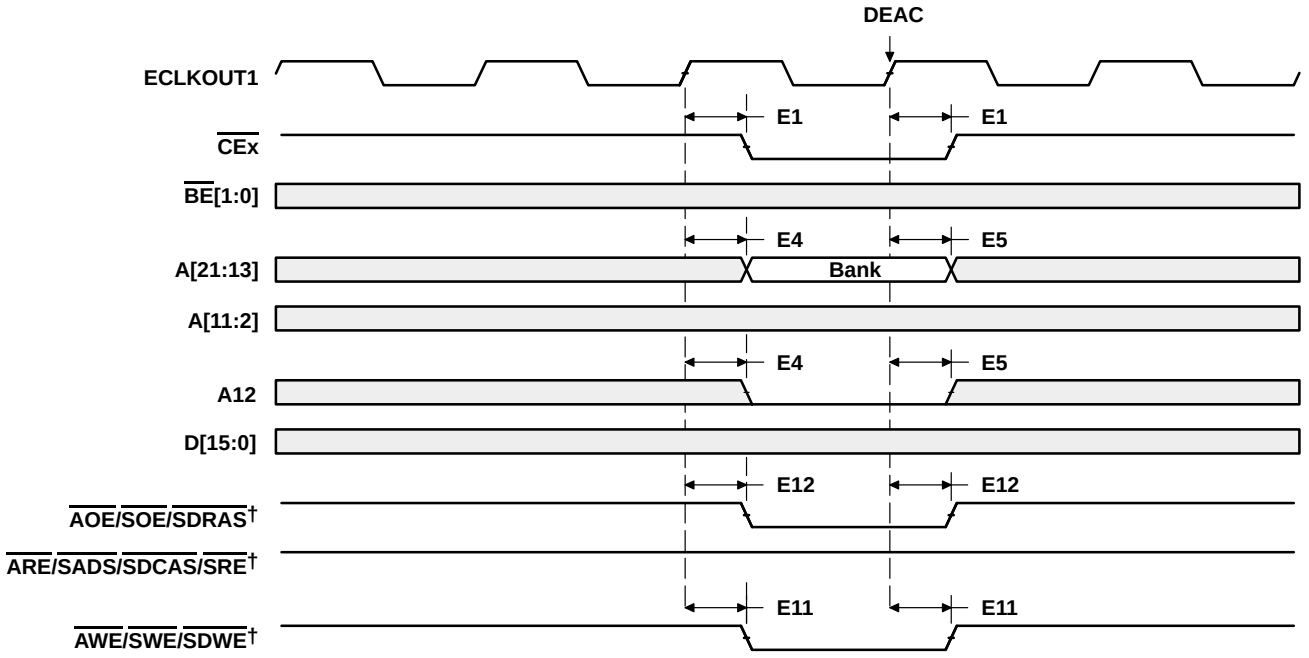
† ARE/SADS/SDCAS/SRE, AWE/SWE/SDWE, and AOE/SOE/SDRAS operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 6–14. SDRAM ACTV Command for EMIF



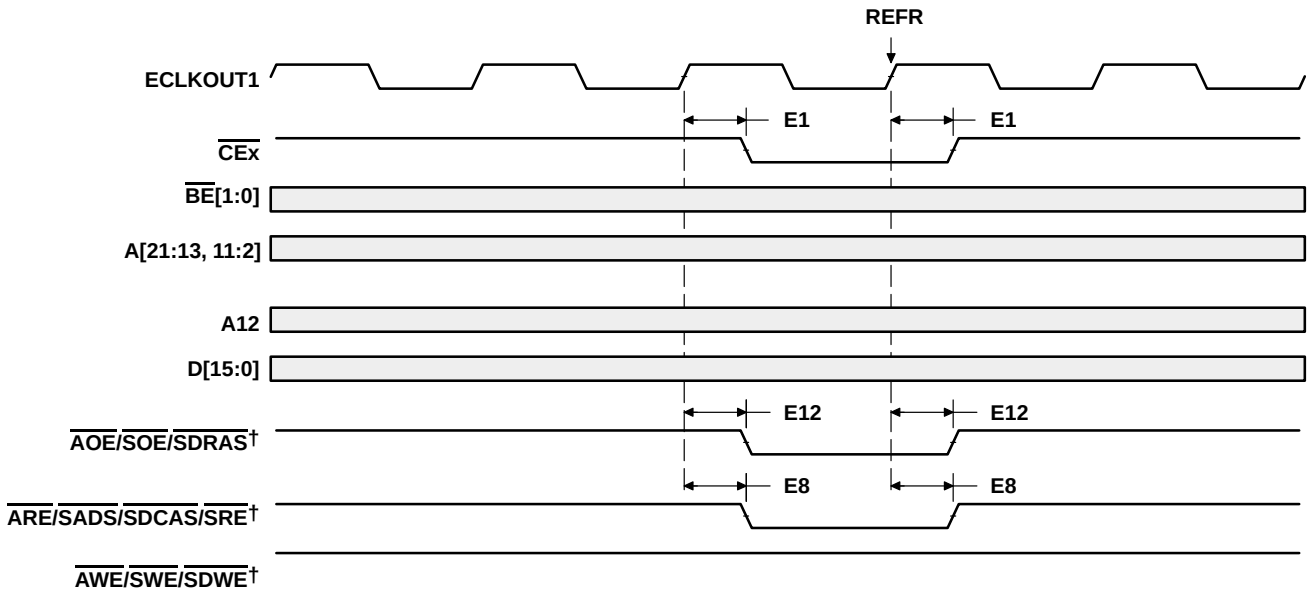
† ARE/SADS/SDCAS/SRE, AWE/SWE/SDWE, and AOE/SOE/SDRAS operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 6–15. SDRAM DCAB Command for EMIF



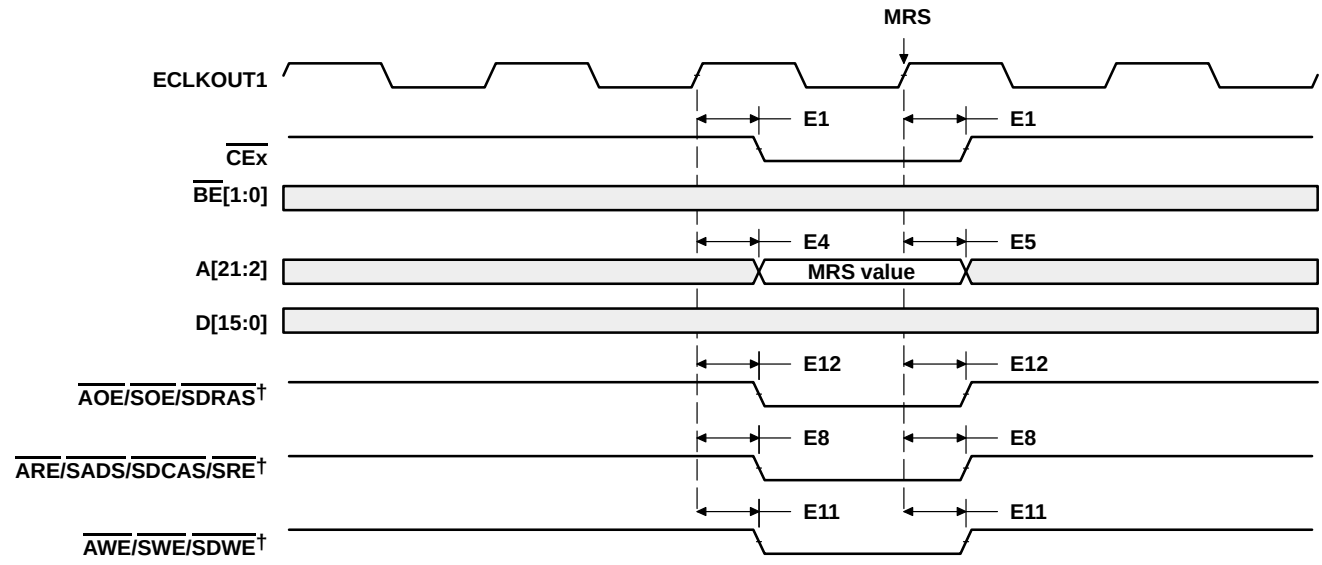
† ARE/SADS/SDCAS/SRE, AWE/SWE/SDWE, and AOE/SOE/SDRAS operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 6–16. SDRAM DEAC Command for EMIF



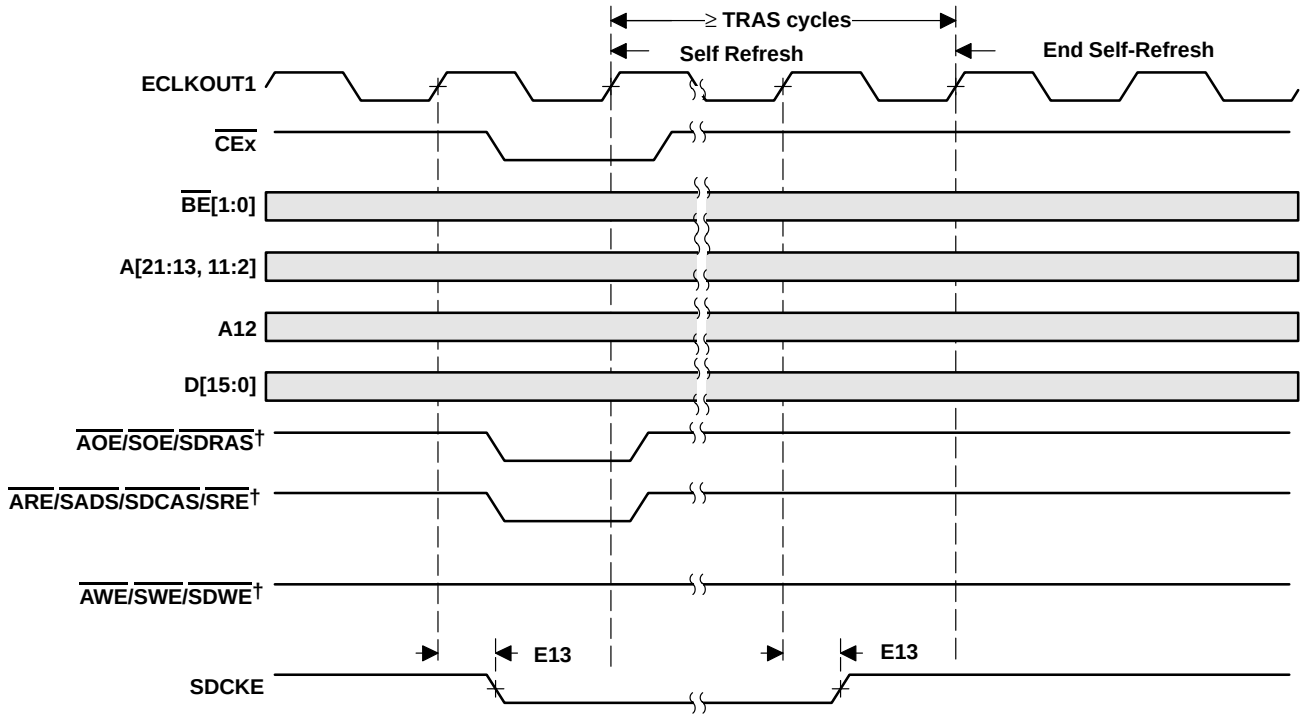
† ARE/SADS/SDCAS/SRE, AWE/SWE/SDWE, and AOE/SOE/SDRAS operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 6–17. SDRAM REFR Command for EMIF



† ARE/SADS/SDCAS/SRE, AWE/SWE/SDWE, and AOE/SOE/SDRAS operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 6–18. SDRAM MRS Command for EMIF



† ARE/SADS/SDCAS/SRE, AWE/SWE/SDWE, and AOE/SOE/SDRAS operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 6–19. SDRAM Self-Refresh Timing for EMIF

6.11 HOLD/HOLDA Timing

Table 6–15 and Table 6–16 assume testing over recommended operating conditions (see Figure 6–20).

Table 6–15. EMIF HOLD/HOLDA Timing Requirements†

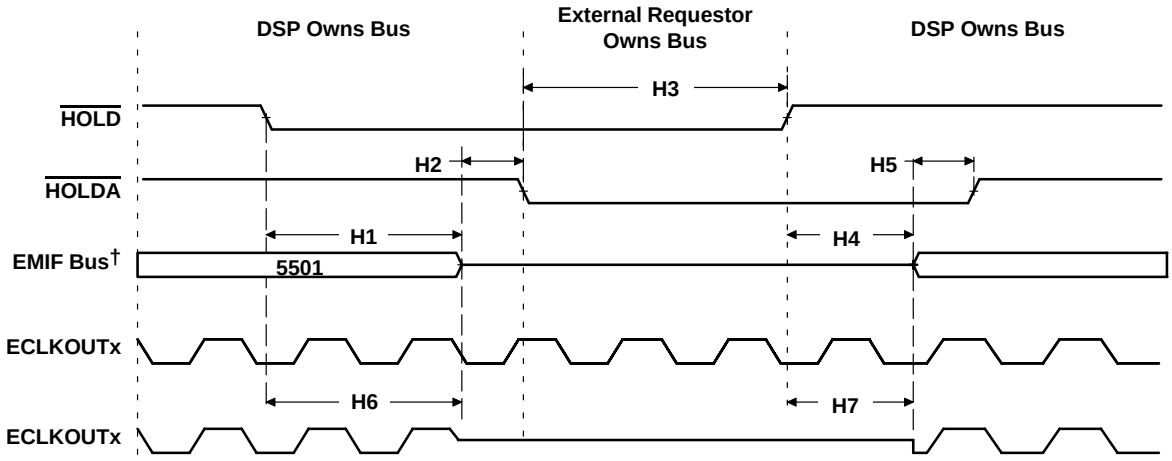
NO.			MIN	MAX	UNIT
H3	$t_{oh(HOLDAL-HOLDL)}$	Hold time, HOLD low after HOLDA low	E		ns

† E = the EMIF input clock (ECLKIN, CPU/1 clock, CPU1/2 clock, or CPU1/4 clock) period in ns for EMIF.

Table 6–16. EMIF HOLD/HOLDA Switching Characteristics†‡§

NO.	PARAMETER		MIN	MAX	UNIT
H1	$t_{d(HOLDL-EMHZ)}$	Delay time, HOLD low to EMIF Bus high impedance	2E	¶	ns
H2	$t_{d(EMHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to HOLDA low	0	2E	ns
H4	$t_{d(HOLDH-EMLZ)}$	Delay time, HOLD high to EMIF Bus low impedance	2E	7E	ns
H5	$t_{d(EMLZ-HOLDAH)}$	Delay time, EMIF Bus low impedance to HOLDA high	0	2E	ns
H6	$t_{d(HOLDL-EKOHZ)}$	Delay time, HOLD low to ECLKOUTx high impedance	2E	¶	ns
H7	$t_{d(HOLDH-EKOLZ)}$	Delay time, HOLD high to ECLKOUTx low impedance	2E	7E	ns

† E = the EMIF input clock (ECLKIN, CPU/1 clock, CPU1/2 clock, or CPU1/4 clock) period in ns for EMIF.
‡ EMIF Bus consists of: CE[3:0], BE[1:0], D[15:0], A[21:2], ARE/SADS/SDCAS/SRE, AOE/SOE/SDRAS, and AWE/SWE/SDWE, SDCKE, and SOE3.
§ The EKxHZ bits in the EMIF Global Control Registers (EGCR1, EGCR2) determine the state of the ECLKOUTx signals during HOLDA. If EKxHZ = 0, ECLKOUTx continues clocking during Hold mode. If EKxHZ = 1, ECLKOUTx goes to high impedance during Hold mode, as shown in Figure 6–20.
¶ All pending EMIF transactions are allowed to complete before HOLDA is asserted. If no bus transactions are occurring, then the minimum delay time can be achieved. Also, bus hold can be indefinitely delayed by setting NOHOLD = 1.



† EMIF Bus consists of: CE[3:0], BE[1:0], D[15:0], A[21:2], ARE/SADS/SDCAS/SRE, AOE/SOE/SDRAS, and AWE/SWE/SDWE, SDCKE, and SOE3.

Figure 6–20. HOLD/HOLDA Timing for EMIF

PRODUCT PREVIEW

6.12 Reset Timings

Table 6–17 and Table 6–18 assume testing over recommended operating conditions (see Figure 6–21).

Table 6–17. Reset Timing Requirements †

NO.			MIN	MAX	UNIT
R1	$t_{w(RSL)}$	Pulse width, $\overline{\text{RESET}}$ low	2P + 5		ns

† P = the period of the clock on the X2/CLKIN pin in ns. For example, when using 20 MHz as the input clock, use P = 50 ns.

Table 6–18. Reset Switching Characteristics †

NO.	PARAMETER			MIN	MAX	UNIT
R3	$t_{d(RSL-EMIFHZ)}$	Delay time, $\overline{\text{RESET}}$ low to EMIF group high impedance‡			6	ns
R4	$t_{d(RSH-EMIFV)}$	Delay time, $\overline{\text{RESET}}$ high to EMIF group valid‡	GPIO4 = 0 (CLKMOD = 0)		41102P + 6	ns
			GPIO4 = 1 (CLKMOD = 1)		6	
R5	$t_{d(RSL-LOWIV)}$	Delay time, $\overline{\text{RESET}}$ low to low group invalid§			6	ns
R6	$t_{d(RSH-LOWV)}$	Delay time, $\overline{\text{RESET}}$ high to low group valid§	GPIO4 = 0 (CLKMOD = 0)		41102P + 6	ns
			GPIO4 = 1 (CLKMOD = 1)		6	
R7	$t_{d(RSL-HIGHIV)}$	Delay time, $\overline{\text{RESET}}$ low to high group invalid§			6	ns
R8	$t_{d(RSH-HIGHV)}$	Delay time, $\overline{\text{RESET}}$ high to high group valid§	GPIO4 = 0 (CLKMOD = 0)		41102P + 6	ns
			GPIO4 = 1 (CLKMOD = 1)		6	
R9	$t_{d(RSL-ZHZ)}$	Delay time, $\overline{\text{RESET}}$ low to Z group high impedance¶			6	ns
R10	$t_{d(RSH-ZV)}$	Delay time, $\overline{\text{RESET}}$ high to Z group invalid¶	GPIO4 = 0 (CLKMOD = 0)		41102P + 6	ns
			GPIO4 = 1 (CLKMOD = 1)		6	

† P = the period of the clock on the X2/CLKIN pin in ns. For example, when using 20 MHz as the input clock, use P = 50 ns.
‡ EMIF group: $\overline{\text{CE}}$[3:0], $\overline{\text{BE}}$[1:0], ECLKOUTx, $\overline{\text{ARE}}$, $\overline{\text{AOE}}$, $\overline{\text{AWE}}$, $\overline{\text{SADS}}$, $\overline{\text{SOE}}$, $\overline{\text{SWE}}$, $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$
§ High group: $\overline{\text{HINT}}$, $\overline{\text{IACK}}$ Low group: $\overline{\text{HOLDA}}$
¶ Z group: A[21:2], D[15:0], CLKR[2:0], CLKX[2:0], FSR[2:0], FSX[2:0], DX[2:0], GPIO[7:0], XF, TIM1, and TIM0.

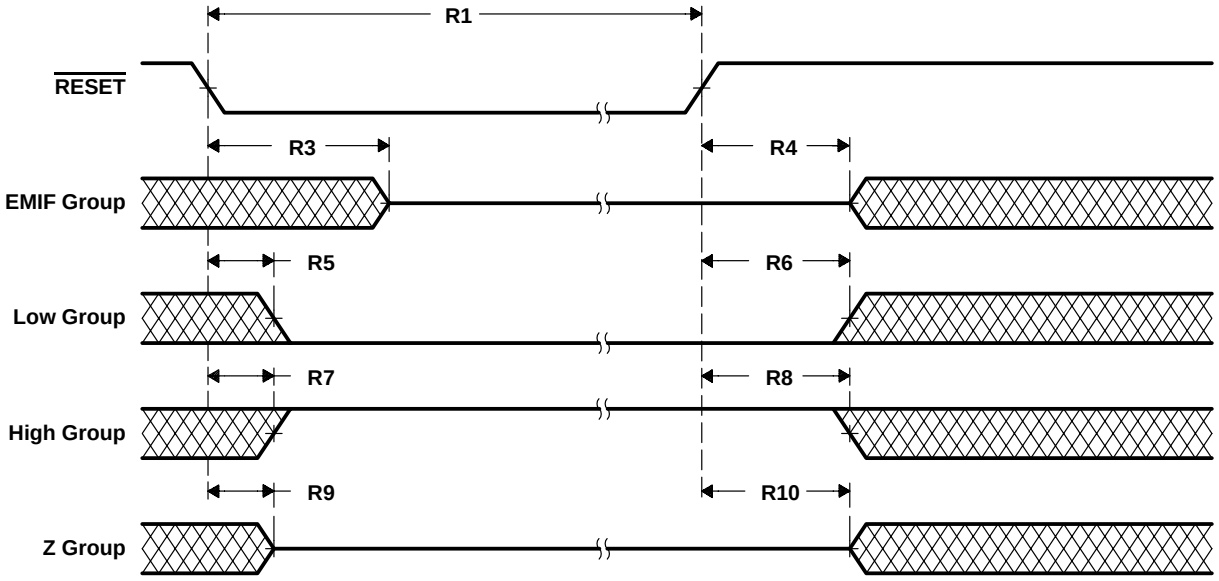


Figure 6–21. Reset Timing

6.13 External Interrupt and Interrupt Acknowledge ($\overline{\text{IACK}}$) Timings

Table 6–19 and Table 6–20 assume testing over recommended operating conditions (see Figure 6–22 and Figure 6–23).

Table 6–19. External Interrupt and Interrupt Acknowledge Timing Requirements

NO.			MIN	MAX	UNIT
I1	$t_{w(INTL)A}$	Pulse width, interrupt low, CPU active	3P†		ns
I2	$t_{w(INTH)A}$	Pulse width, interrupt high, CPU active	2P†		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

Table 6–20. External Interrupt and Interrupt Acknowledge Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
I3	$t_{d(COH-IACKV)}$	Delay time, CLKOUT high to $\overline{\text{IACK}}$ valid	0	3	ns

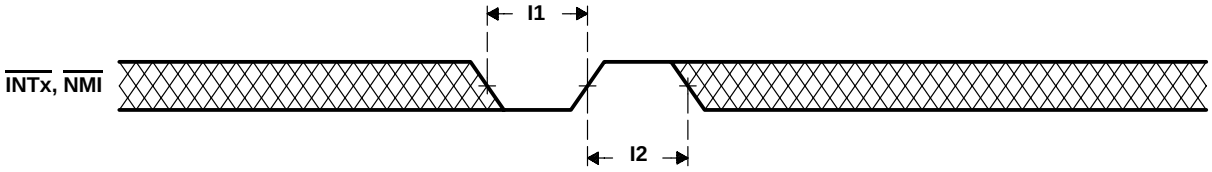


Figure 6–22. External Interrupt Timing

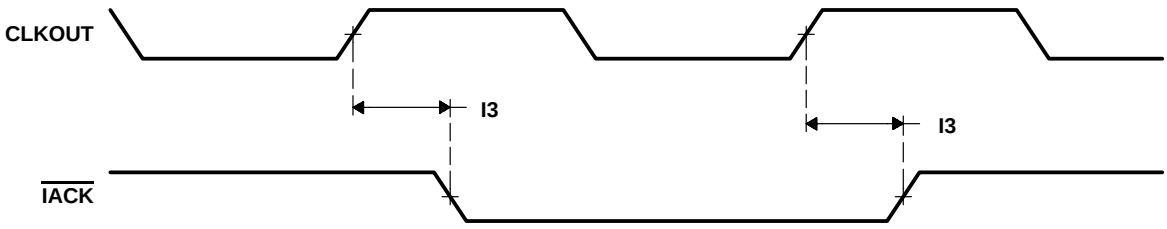


Figure 6–23. External Interrupt Acknowledge Timing

6.14 XF Timing

Table 6–21 assumes testing over recommended operating conditions (see Figure 6–24).

Table 6–21. XF Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
X1	$t_{d(XF)}$	Delay time, CLKOUT high to XF high	0	3	ns
		Delay time, CLKOUT high to XF low	0	3	

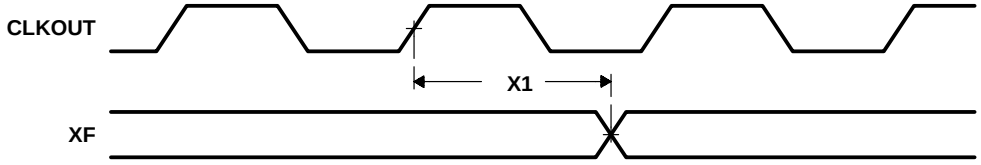


Figure 6–24. XF Timing

6.15 General-Purpose Input/Output (GPIOx)Timing

Table 6–22 and Table 6–23 assume testing over recommended operating conditions (see Figure 6–25).

Table 6–22. GPIO Pins Configured as Inputs Timing Requirements

NO.			MIN	MAX	UNIT
G2	$t_{su(GPIO–COH)}$	Setup time, GPIOx input valid before CLKOUT high	4		ns
G3	$t_{h(COH–GPIO)}$	Hold time, GPIOx input valid after CLKOUT high	0		ns

Table 6–23. GPIO Pins Configured as Inputs Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
G1	$t_{d(COH–GPIO)}$	Delay time, CLKOUT high to GPIOx output change	0	3	ns

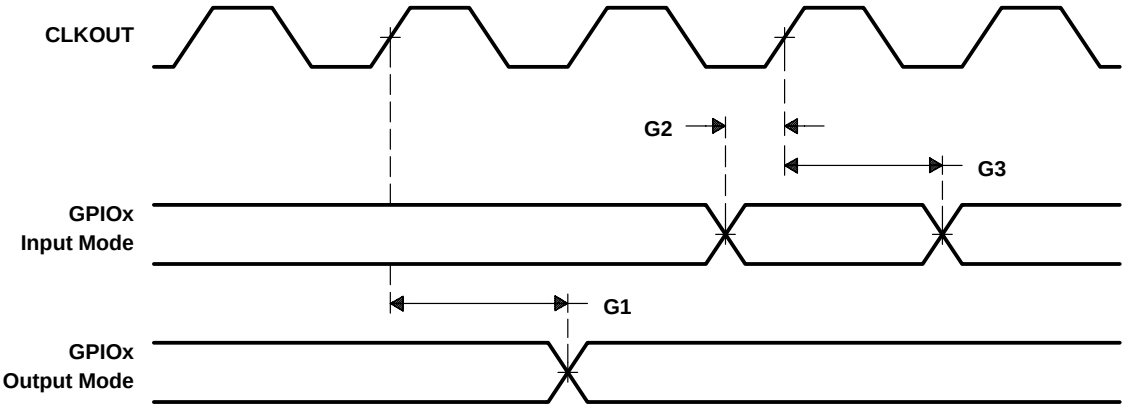


Figure 6–25. General-Purpose Input/Output (GPIOx) Signal Timings

6.15.1 TIM0/TIM1/WDTOUT Timings

Table 6–24 and Table 6–25 assume testing over recommended operating conditions (see Figure 6–26 and Figure 6–27).

Table 6–24. TIM0/TIM1/WDTOUT Pins Configured as Inputs Timing Requirements†

NO.			MIN	MAX	UNIT
T4	$t_{w(TIML)}$	Pulse width, TIM0/TIM1/WDTOUT low	P/4		ns
T5	$t_{w(TIMH)}$	Pulse width, TIM0/TIM1/WDTOUT high	P/4		ns

† P = (Divider1 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the fast peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

Table 6–25. TIM0/TIM1/WDTOUT Pins Configured as Outputs Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
T1	$t_{d(COH–TIMH)}$	Delay time, CLKOUT high to TIM0/TIM1/WDTOUT high	0	3	ns
T2	$t_{d(COH–TIML)}$	Delay time, CLKOUT high to TIM0/TIM1/WDTOUT low	0	3	ns
T3	$t_{w(TIM)}$	Pulse duration, TIM0/TIM1/WDTOUT	P†		ns

† P = (Divider1 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the fast peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

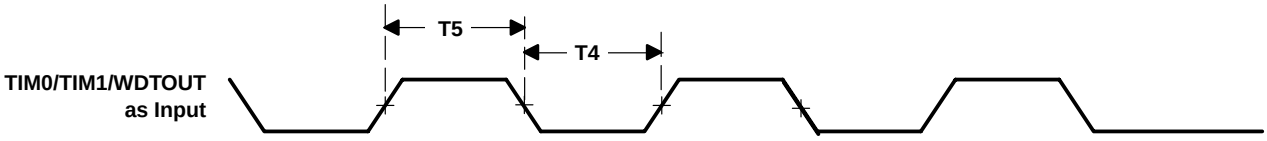


Figure 6–26. TIM0/TIM1/WDTOUT Timings When Configured as Inputs

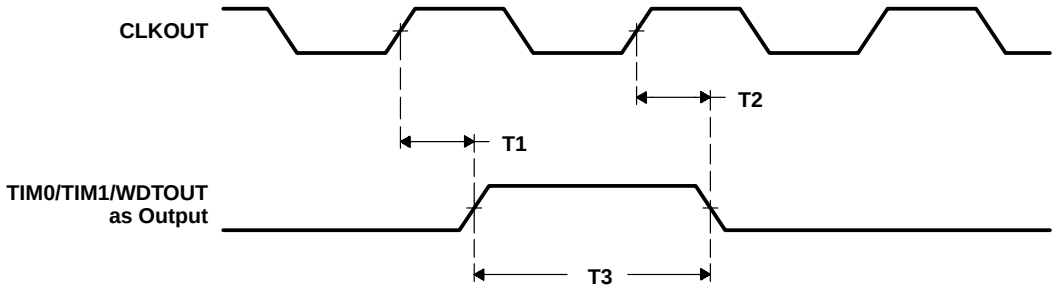


Figure 6–27. TIM0/TIM1/WDTOUT Timings When Configured as Outputs

6.16 Multichannel Buffered Serial Port (McBSP) Timing

6.16.1 McBSP Transmit and Receive Timings

Table 6–26 and Table 6–27 assume testing over recommended operating conditions (see Figure 6–28 and Figure 6–29).

Table 6–26. McBSP Transmit and Receive Timing Requirements†‡

NO.				MIN	MAX	UNIT
M11	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	2P		ns
M12	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	P – 1		ns
M13	$t_{r(CKRX)}$	Rise time, CLKR/X	CLKR/X ext		5	ns
M14	$t_{f(CKRX)}$	Fall time, CLKR/X	CLKR/X ext		5	ns
M15	$t_{su(FRH–CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	5		ns
			CLKR ext	1		
M16	$t_{h(CKRL–FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	0		ns
			CLKR ext	2		
M17	$t_{su(DRV–CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	3		ns
			CLKR ext	1		
M18	$t_{h(CKRL–DRV)}$	Hold time, DR valid after CLKR low	CLKR int	0		ns
			CLKR ext	2		
M19	$t_{su(FXH–CKXL)}$	Setup time, external FSX high before CLKX low	CLKX int	5		ns
			CLKX ext	1		
M20	$t_{h(CKXL–FXH)}$	Hold time, external FSX high after CLKX low	CLKX int	0		ns
			CLKX ext	2		

† Polarity bits CLKRP = CLKXP = FSRP = FSXP = 0. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

Table 6–27. McBSP Transmit and Receive Switching Characteristics†‡

NO.	PARAMETER				MIN	MAX	UNIT
M1	$t_{c(CKRX)}$	Cycle time, CLKR/X		CLKR/X int	2P		ns
M2	$t_{w(CKRXH)}$	Pulse duration, CLKR/X high		CLKR/X int	D–1§	D+1§	ns
M3	$t_{w(CKRXL)}$	Pulse duration, CLKR/X low		CLKR/X int	C–1§	C+1§	ns
M4	$t_{d(CKRH–FRV)}$	Delay time, CLKR high to internal FSR valid		CLKR int	–2	2	ns
				CLKR ext	4	8	
M5	$t_{d(CKXH–FXV)}$	Delay time, CLKX high to internal FSX valid		CLKX int	–2	2	ns
				CLKX ext	4	8	
M6	$t_{dis(CKXH–DXHZ)}$	Disable time, CLKX high to DX high impedance following last data bit		CLKX int	–5	5	ns
				CLKX ext	1	11	
M7	$t_{d(CKXH–DXV)}$	Delay time, CLKX high to DX valid. This applies to all bits except the first bit transmitted.	DXENA = 0	CLKX int		2	ns
				CLKX ext		8	
		Delay time, CLKX high to DX valid¶ Only applies to first bit transmitted when in Data Delay 1 or 2 (XDATDLY=01b or 10b) modes	DXENA = 0	CLKX int		2	
				CLKX ext		8	
			DXENA = 1	CLKX int		2P+2	
				CLKX ext		2P+8	
M8	$t_{en(CKXH–DX)}$	Enable time, CLKX high to DX driven¶ Only applies to first bit transmitted when in Data Delay 1 or 2 (XDATDLY=01b or 10b) modes	DXENA = 0	CLKX int	0		ns
				CLKX ext	6		
			DXENA = 1	CLKX int	2P		
				CLKX ext	2P+6		
M9	$t_{d(FXH–DXV)}$	Delay time, FSX high to DX valid¶ Only applies to first bit transmitted when in Data Delay 0 (XDATDLY=00b) mode.	DXENA = 0	FSX int		1	ns
				FSX ext		7	
			DXENA = 1	FSX int		2P+1	
				FSX ext		2P+7	
M10	$t_{en(FXH–DX)}$	Enable time, FSX high to DX driven¶ Only applies to first bit transmitted when in Data Delay 0 (XDATDLY=00b) mode	DXENA = 0	FSX int	0		ns
				FSX ext	6		
			DXENA = 1	FSX int	2P		
				FSX ext	P+6		

† Polarity bits CLKRP = CLKXP = FSRP = FSXP = 0. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

§ T=CLKRX period = (1 + CLKGDV) * P
C=CLKRX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * P when CLKGDV is even
D=CLKRX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * P when CLKGDV is even

¶ See the *TMS320C55x DSP Peripherals Reference Guide* (literature number SPRU317) for a description of the DX enable (DXENA) and data delay features of the McBSP.

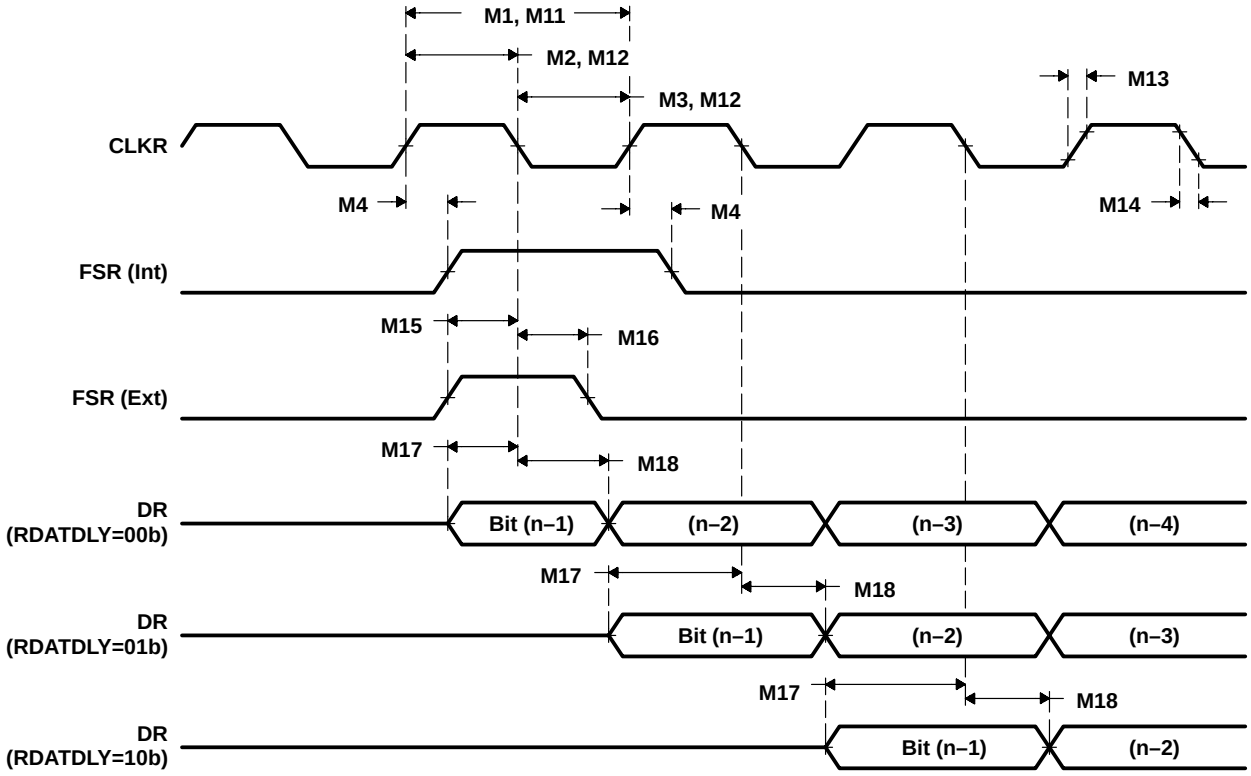
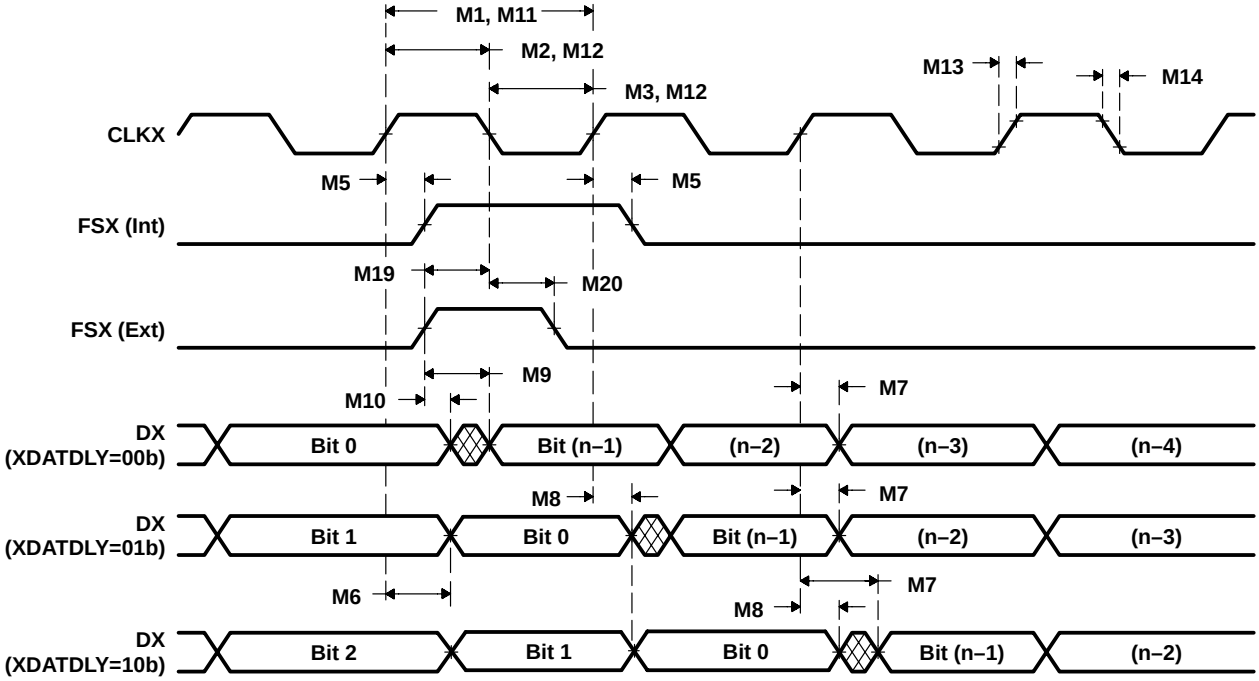


Figure 6–28. McBSP Receive Timings



NOTE A: This figure does not include first or last frames. For first frame, no data will be present before frame synchronization. For last frame, no data will be present after frame synchronization.

Figure 6–29. McBSP Transmit Timings

6.16.2 McBSP General-Purpose I/O Timing

Table 6–28 and Table 6–29 assume testing over recommended operating conditions (see Figure 6–30).

Table 6–28. McBSP General-Purpose I/O Timing Requirements

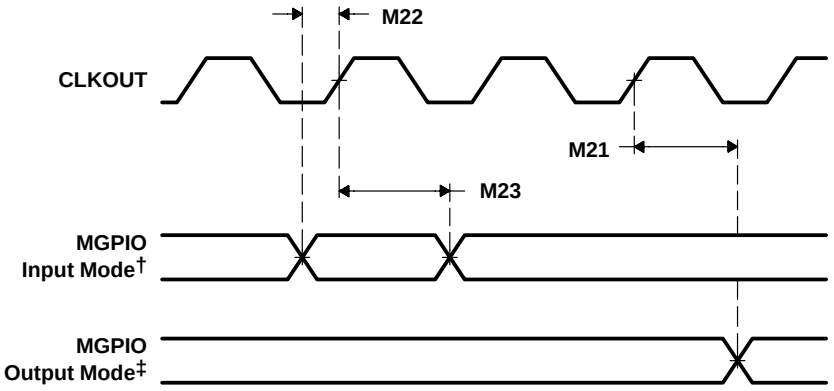
NO.			MIN	MAX	UNIT
M22	$t_{su(MGPIO–COH)}$	Setup time, MGPIOx input mode before CLKOUT high†	4		ns
M23	$t_{h(COH–MGPIO)}$	Hold time, MGPIOx input mode after CLKOUT high†	0		ns

† MGPIOx refers to CLKRx, FSRx, DRx, CLKXx, or FSXx when configured as a general-purpose input.

Table 6–29. McBSP General-Purpose I/O Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
M21	$t_{d(COH–MGPIO)}$	Delay time, CLKOUT high to MGPIOx output mode‡	0	4	ns

‡ MGPIOx refers to CLKRx, FSRx, CLKXx, FSXx, or DXx when configured as a general-purpose output.



† MGPIOx refers to CLKRx, FSRx, DRx, CLKXx, or FSXx when configured as a general-purpose input.
‡ MGPIOx refers to CLKRx, FSRx, CLKXx, FSXx, or DXx when configured as a general-purpose output.

Figure 6–30. McBSP General-Purpose I/O Timings

6.16.3 McBSP as SPI Master or Slave Timing

Table 6–30 to Table 6–37 assume testing over recommended operating conditions (see Figure 6–31 through Figure 6–34).

Table 6–30. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 10b, CLKXP = 0)†‡

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M30	$t_{su(DRV–CKXL)}$	Setup time, DR valid before CLKX low	3		2 – 6P		ns
M31	$t_{h(CKXL–DRV)}$	Hold time, DR valid after CLKX low	1		6 + 6P		ns
M32	$t_{su(FXL–CKXH)}$	Setup time, FSX low before CLKX high			10		ns
M33	$t_{c(CKX)}$	Cycle time, CLKX	2P		16P		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

Table 6–31. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 10b, CLKXP = 0)†‡§

NO.	PARAMETER		MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M24	$t_{d(CKXL–FXL)}$	Delay time, CLKX low to FSX low¶	T – 1	T + 1			ns
M25	$t_{d(FXL–CKXH)}$	Delay time, FSX low to CLKX high#	C – 1	C + 1			ns
M26	$t_{d(CKXH–DXV)}$	Delay time, CLKX high to DX valid	–2	2	3P + 4	5P+ 10	ns
M27	$t_{dis(CKXL–DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	C – 2	C +10			ns
M28	$t_{dis(FXH–DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			2P+ 4	4P + 10	ns
M29	$t_{d(FXL–DXV)}$	Delay time, FSX low to DX valid			2P + 4	4P + 10	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.
§ T = BCLKX period = (1 + CLKGDV) * 2P
C = BCLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * 2P when CLKGDV is even
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

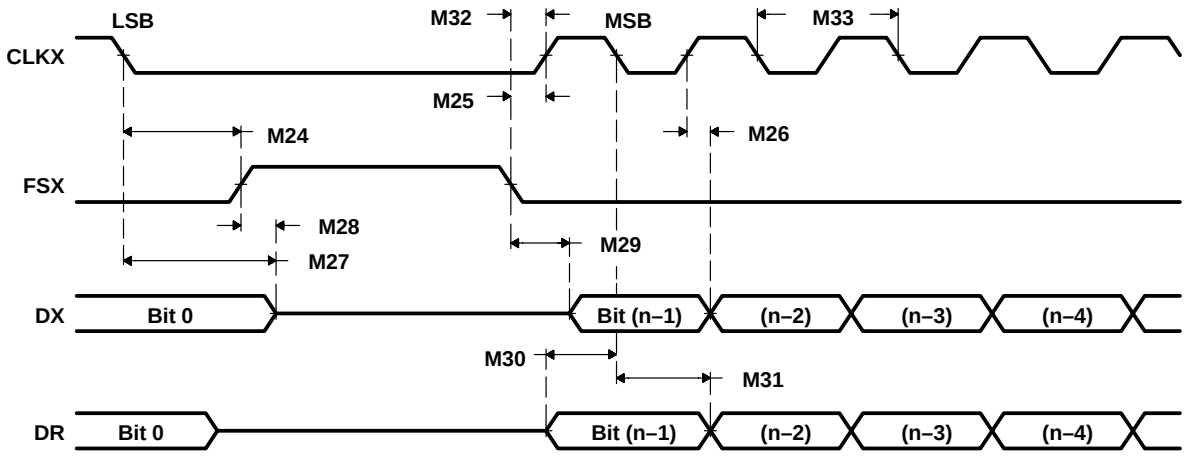


Figure 6–31. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

Table 6–32. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 11b, CLKXP = 0)†‡

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M39	$t_{su(DRV–CKXH)}$	Setup time, DR valid before CLKX high	3		2 – 6P		ns
M40	$t_{h(CKXH–DRV)}$	Hold time, DR valid after CLKX high	1		6 +6P		ns
M41	$t_{su(FXL–CKXH)}$	Setup time, FSX low before CLKX high			10		ns
M42	$t_{c(CKX)}$	Cycle time, CLKX	2P		16P		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

Table 6–33. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 11b, CLKXP = 0)†‡§

NO.	PARAMETER		MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M34	$t_{d(CKXL–FXL)}$	Delay time, CLKX low to FSX low¶	C – 1	C + 1			ns
M35	$t_{d(FXL–CKXH)}$	Delay time, FSX low to CLKX high#	T – 1	T + 1			ns
M36	$t_{d(CKXH–DXV)}$	Delay time, CLKX high to DX valid	–2	2	3P + 4	5P + 10	ns
M37	$t_{dis(CKXL–DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	–2	10	3P + 4	5P + 10	ns
M38	$t_{d(FXL–DXV)}$	Delay time, FSX low to DX valid	D – 2	D +10	2P – 4	4P + 10	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.
§ T = CLKX period = (1 + CLKGDV) * P
C = CLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * P when CLKGDV is even
D = CLKX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * P when CLKGDV is even
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

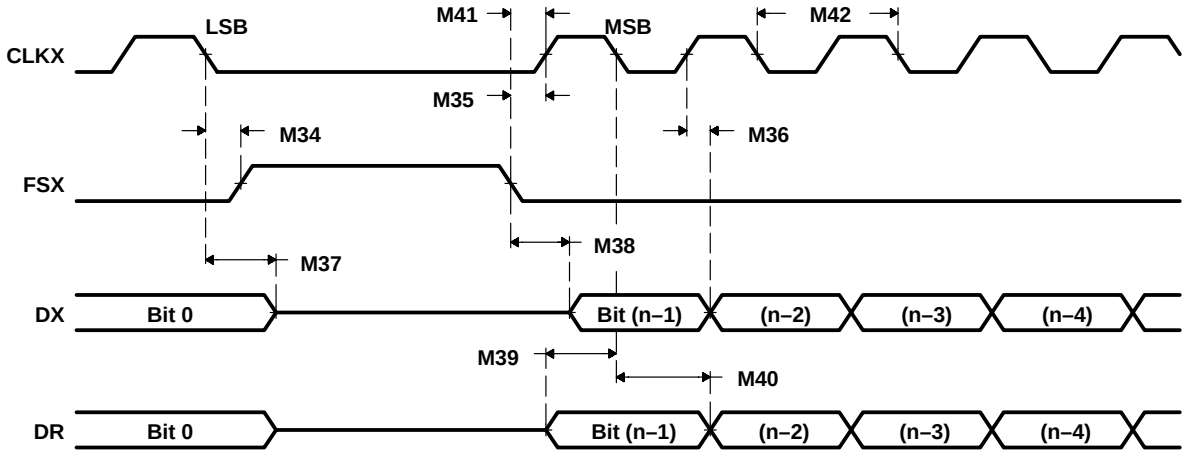


Figure 6–32. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

PRODUCT PREVIEW

Table 6–34. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 10b, CLKXP = 1)†‡

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M49	$t_{su(DRV–CKXH)}$	Setup time, DR valid before CLKX high	3		2 – 2P		ns
M50	$t_{h(CKXH–DRV)}$	Hold time, DR valid after CLKX high	1		6 + 6P		ns
M51	$t_{su(FXL–CKXL)}$	Setup time, FSX low before CLKX low			10		ns
M52	$t_{c(CKX)}$	Cycle time, CLKX	2P		16P		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

Table 6–35. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 10b, CLKXP = 1)†‡§

NO.	PARAMETER		MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M43	$t_{d(CKXH–FXL)}$	Delay time, CLKX high to FSX low¶	T – 1	T + 1			ns
M44	$t_{d(FXL–CKXL)}$	Delay time, FSX low to CLKX low#	D – 1	D + 1			ns
M45	$t_{d(CKXL–DXV)}$	Delay time, CLKX low to DX valid	–2	2	3P + 4	5P + 10	ns
M46	$t_{dis(CKXH–DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	D – 2	D +10			ns
M47	$t_{dis(FXH–DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			2P + 4	4P + 10	ns
M48	$t_{d(FXL–DXV)}$	Delay time, FSX low to DX valid			2P – 4	4P + 10	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.
§ T = CLKX period = (1 + CLKGDV) * P
D = CLKX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * P when CLKGDV is even
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

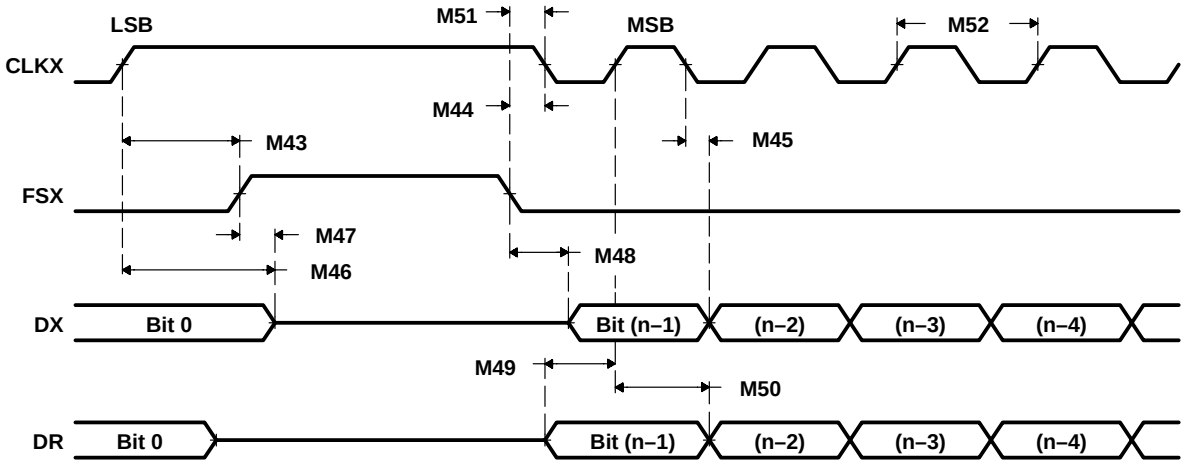


Figure 6–33. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

Table 6–36. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 11b, CLKXP = 1)†‡

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M58	$t_{su(DRV–CKXL)}$	Setup time, DR valid before CLKX low	3		2 – 6P		ns
M59	$t_{h(CKXL–DRV)}$	Hold time, DR valid after CLKX low	1		6 + 6P		ns
M60	$t_{su(FXL–CKXL)}$	Setup time, FSX low before CLKX low			10		ns
M61	$t_{c(CKX)}$	Cycle time, CLKX	2P		16P		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.

Table 6–37. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 11b, CLKXP = 1)†‡§

NO.	PARAMETER		MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
M53	$t_{d(CKXH–FXL)}$	Delay time, CLKX high to FSX low¶	D – 1	D + 1			ns
M54	$t_{d(FXL–CKXL)}$	Delay time, FSX low to CLKX low#	T – 1	T + 1			ns
M55	$t_{d(CKXH–DXV)}$	Delay time, CLKX high to DX valid	–2	2	3P + 4	5P + 10	ns
M56	$t_{dis(CKXH–DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	–2	10	3P + 4	5P + 10	ns
M57	$t_{d(FXL–DXV)}$	Delay time, FSX low to DX valid	C – 2	C +10	2P – 4	4P + 10	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ P = (Divider2 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the slow peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.
§ T = CLKX period = (1 + CLKGDV) * P
C = CLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * P when CLKGDV is even
D = CLKX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * P when CLKGDV is even
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

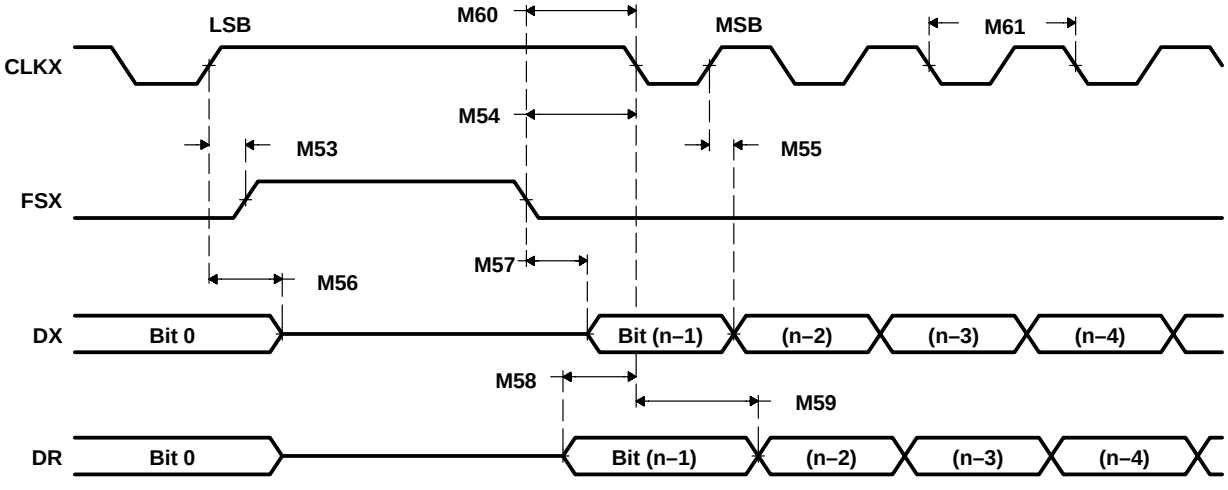


Figure 6–34. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

PRODUCT PREVIEW

6.17 HPI Timings

Table 6–38 and Table 6–39 assume testing over recommended operating conditions (see Figure 6–35 through Figure 6–39).

Table 6–38. HPI Timing Requirements†‡

NO.			MIN	MAX	UNIT
H9	$t_{su(HASL–HDSL)}$	Setup time, $\overline{HAS}$ low before DS falling edge	5		ns
H10	$t_{h(HDSL–HASL)}$	Hold time, $\overline{HAS}$ low after DS falling edge	2		ns
H11	$t_{su(HBV–HASL)}$	Setup time, HAD valid before $\overline{HAS}$ falling edge	5		ns
H12	$t_{h(HASL–HBV)}$	Hold time, HAD valid after $\overline{HAS}$ falling edge	5		ns
H13	$t_{w(HDSL)}$	Pulse duration, DS low	10		ns
H14	$t_{w(HDSH)}$	Pulse duration, DS high	2P		ns
H15	$t_{su(HBV–HDSL)}$	Setup time, HAD valid before DS falling edge	5		ns
H16	$t_{h(HDSL–HBV)}$	Hold time, HAD valid after DS falling edge	5		ns
H17	$t_{su(HDV–HDSH)W}$	Setup time, HD valid before DS rising edge	10		ns
H18	$t_{h(HDSH–HDV)W}$	Hold time, HD valid after DS rising edge	0		ns

† P = (Divider1 Ratio)/(CPU Clock Frequency) in ns. For example, when running parts at 300 MHz with the fast peripheral domain at 1/2 the CPU clock frequency, use P = 2/300 MHz = 6.66 ns.
‡ DS refers to logical OR of $\overline{HCS}$, $\overline{HDS1}$, and $\overline{HDS2}$. HD refers to HPI Data Bus. $\overline{HDS}$ refers to $\overline{HDS1}$ or $\overline{HDS2}$. HAD refers to HCNTL0, HCNTL1, and HR/$\overline{W}$.

Table 6–39. HPI Switching Characteristics‡§

NO.	PARAMETER			MIN	MAX	UNIT
H1	$t_{d(HDSL–HDV)}$	Delay time, DS low to HD valid	*Case 1.* Read Data present in FIFO.	3	20	ns
			Case 2. FIFO is Empty; No other DMA/MEM activity is present.		n*2H+20	ns
			Case 3. HPIC or Unobstructed HPIA Read.	3	20	ns
H2	$t_{d(HDSH–HDV)R}$	Delay time, DS rising edge to HD valid, read		0	10	ns
H3	$t_{d(HDSL–HDD)}$	Delay time, DS low to HD driven		3	20	ns
H4	$t_{d(HDSL–HRDYL)}$	Delay time, DS low to HRDY low			12	ns
H5	$t_{d(HDSH–HRDYL)}$	Delay time, DS high to HRDY low			12	ns
H6	$t_{d(HDSH–HRDYH)}$	Delay time, DS high to HRDY high	*Case 1.* FIFO is Empty; No other DMA/MEM activity present.		n*2H+20	ns
			Case 2. FIFO is Full; No other DMA/MEM activity is present.		n*2H+20	ns
H7	$t_{v(HRDYH–HDV)}$	Valid time, HD valid after HRDY high			7	ns
H8	$t_{d(COH–HINT)}$	Delay time, CLKOUT rising edge to $\overline{HINT}$ change			3	ns

‡ DS refers to logical OR of $\overline{HCS}$, $\overline{HDS1}$, and $\overline{HDS2}$. HD refers to HPI Data Bus. $\overline{HDS}$ refers to $\overline{HDS1}$ or $\overline{HDS2}$. HAD refers to HCNTL0, HCNTL1, and HR/$\overline{W}$.
§ H is half the CPU clock cycle.

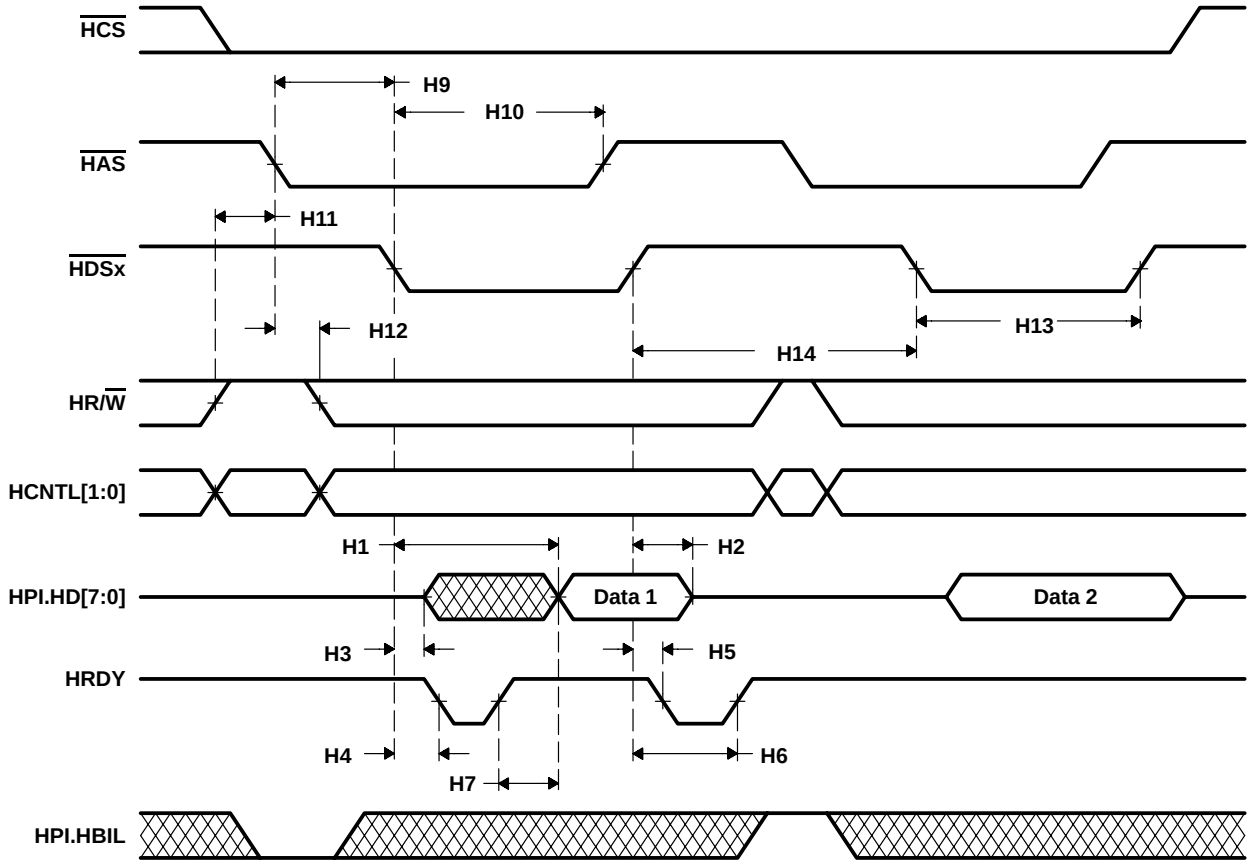


Figure 6–35. Multiplexed Read Timings Using $\overline{\text{HAS}}$

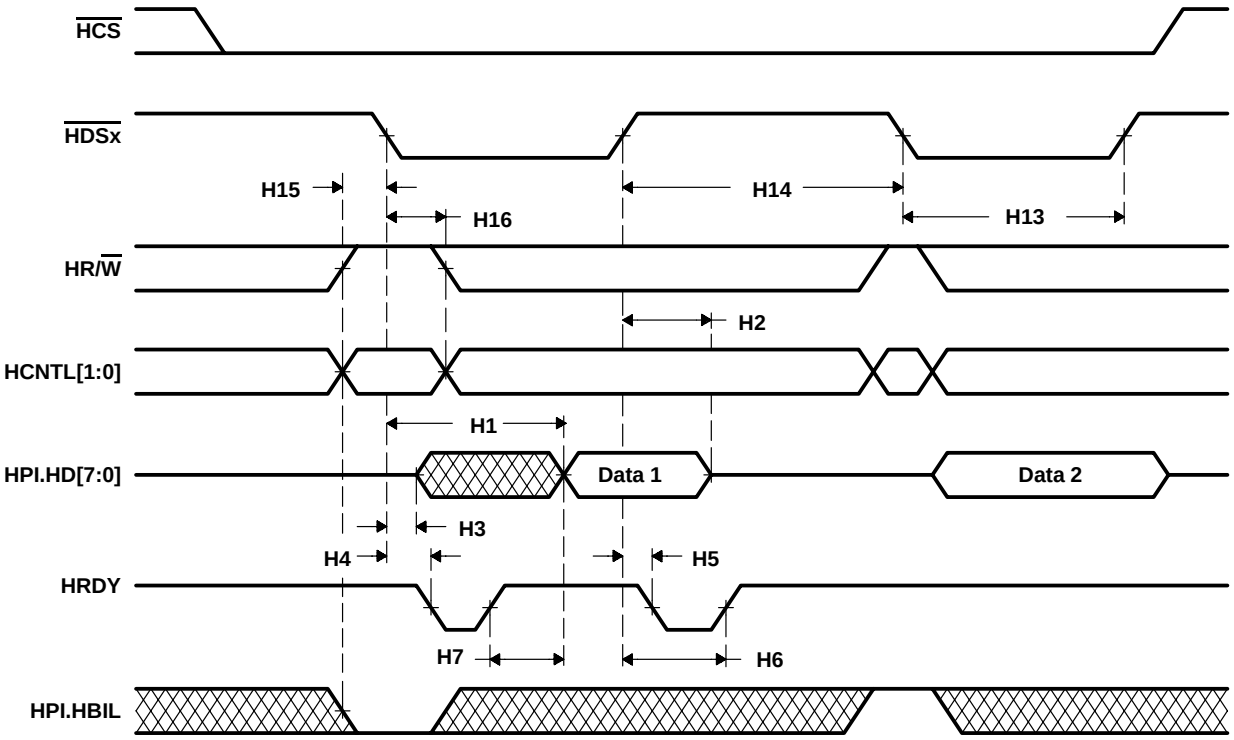


Figure 6–36. Multiplexed Read Timings With HAS Held High

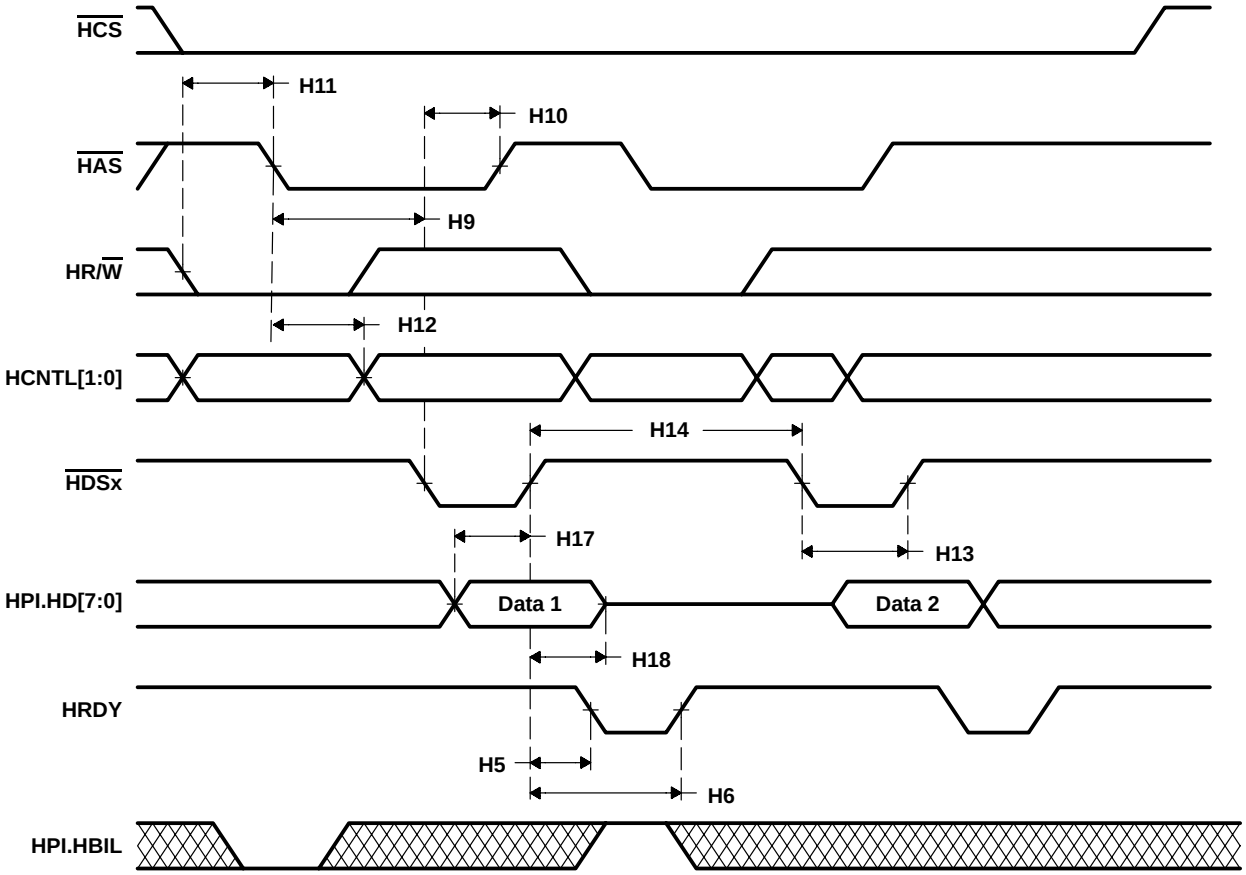


Figure 6–37. Multiplexed Write Timings Using $\overline{\text{HAS}}$

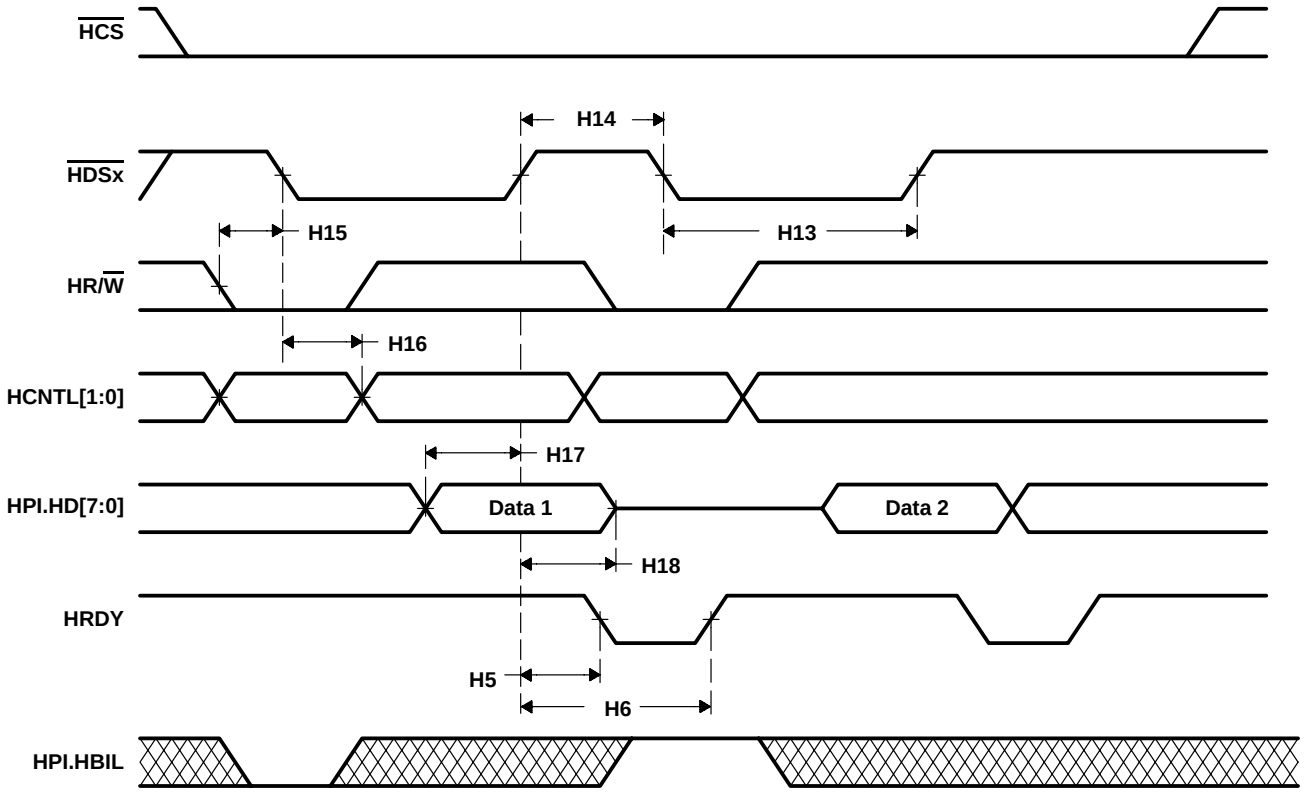


Figure 6–38. Multiplexed Write Timings With $\overline{\text{HAS}}$ Held High

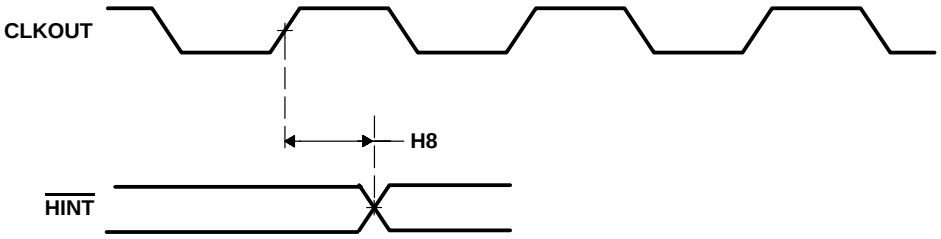


Figure 6–39. $\overline{\text{HINT}}$ Timing

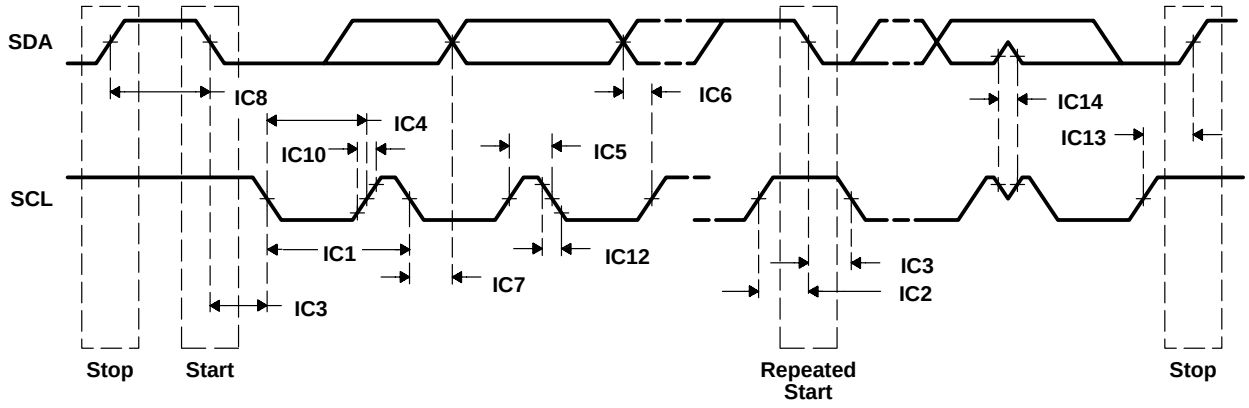
6.18 I²C Timing

Table 6–40 assumes testing over recommended operating conditions (see Figure 6–40).

Table 6–40. I²C Signals (SDA and SCL) Switching Characteristics

NO.	PARAMETER			STANDARD MODE		FAST MODE		UNIT
				MIN	MAX	MIN	MAX	
IC1	$t_{c(SCL)}$	Cycle time, SCL		10		2.5		µs
IC2	$t_{su(SCLH-SDAL)}$	Setup time, SCL high before SDA low for a START and a repeated START condition		4.7		0.6		µs
IC3	$t_{h(SCLL-SDAL)}$	Hold time, SCL low after SDA low for a START and a repeated START condition		4		0.6		µs
IC4	$t_{w(SCLL)}$	Pulse duration, SCL low		4.7		1.3		µs
IC5	$t_{w(SCLH)}$	Pulse duration, SCL high		4		0.6		µs
IC6	$t_{su(SDA-SCLH)}$	Setup time, SDA valid before SCL high		250	†	100		ns
IC7	$t_{h(SDA-SCLL)}$	Hold time, SDA valid after SCL low	For CBUS compatible masters	5				µs
			For I²C bus devices	0		0	0.9	
IC8	$t_{w(SDAH)}$	Pulse duration, SDA high between STOP and START conditions		4.7		1.3		µs
IC9	$t_{r(SDA)}$	Rise time, SDA			1000		300	ns
IC10	$t_{r(SCL)}$	Rise time, SCL			1000		300	
IC11	$t_{f(SDA)}$	Fall time, SDA			300		300	ns
IC12	$t_{f(SCL)}$	Fall time, SCL			300		300	
IC13	$t_{su(SCLH-SDAH)}$	Setup time, SCL high before SDA high (for STOP condition)		4.0		0.6		µs
IC14	$t_{w(SP)}$	Pulse duration, spike (must be suppressed)				0	50	µs
IC15	C_b‡	Capacitive load for each bus line			400		400	pF

† The maximum $t_{h(SCLL-SDAL)}$ has only to be met if the device does not stretch the low period ($t_{w(SCLL)}$) of the SCL signal.
‡ C_b = The total capacitance of one bus line in pF.



NOTES: A. A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHmin} of the SCL signal) to bridge the undefined region of the falling edge of SCL.
B. The maximum $t_{h(SCLL–SDAL)}$ has only to be met if the device does not stretch the LOW period [$t_{w(SCLL)}$] of the SCL signal.
C. A Fast-mode I²C-bus device can be used in a Standard-mode I²C-bus system, but the requirement $t_{su(SDA–SCLH)}$ • 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line t_r max + $t_{su(SDA–SCLH)}$ = 1000 + 250 = 1250 ns (according to the Standard-mode I²C-bus specification) before the SCL line is released.
D. C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

Figure 6–40. I²C Timings

6.19 UART Timing

Table 6–41 to Table 6–42 assume testing over recommended operating conditions (see Figure 6–41).

Table 6–41. UART Timing Requirements

NO.			MIN	MAX	UNIT
U1	$t_{w(UDB)R}$	Pulse width, receive data bit	0.99U†	1.01U†	ns
U2	$t_{w(USB)R}$	Pulse width, receive start bit	0.99U†	1.01U†	ns

† U = UART baud time = 1/programmed baud rate

Table 6–42. UART Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
U3	f_{baud}	Maximum programmable baud rate		5	MHz
U4	$t_{w(UDB)X}$	Pulse width, transmit data bit	U – 2†	U + 2†	ns
U5	$t_{w(USB)X}$	Pulse width, transmit start bit	U – 2†	U + 2†	ns

† U = UART baud time = 1/programmed baud rate

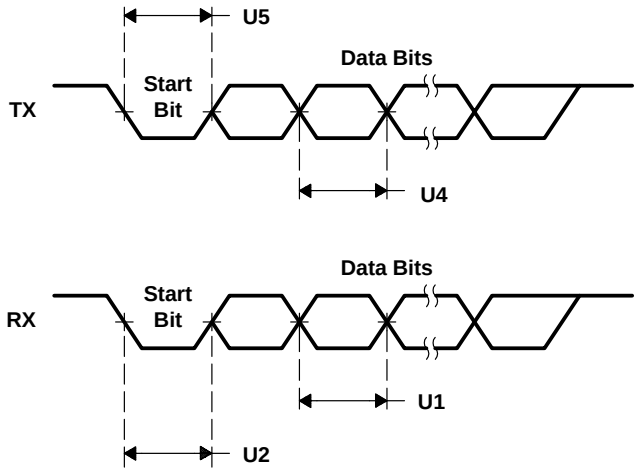
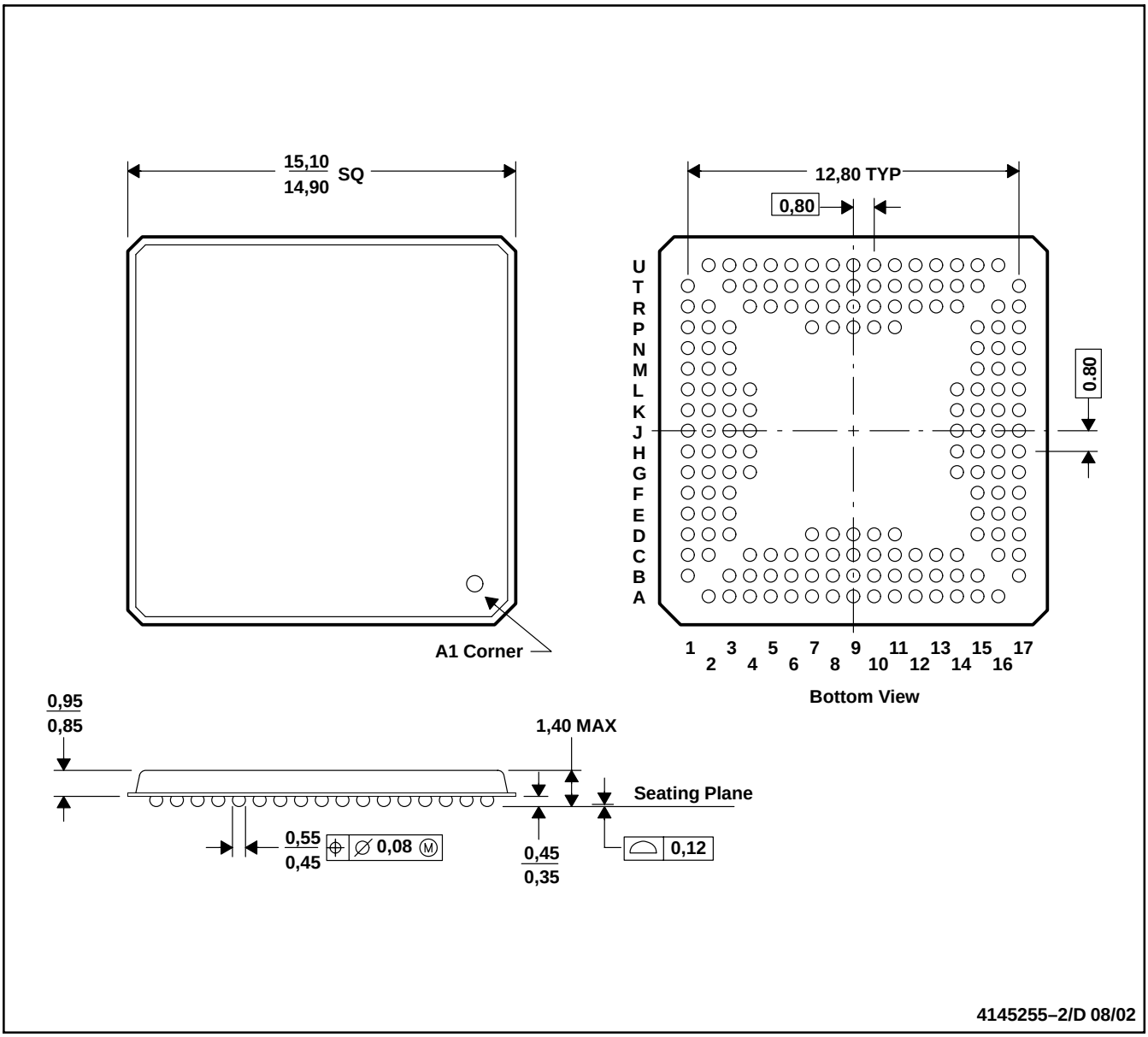


Figure 6–41. UART Timings

7 Mechanical Data

7.1 Ball Grid Array Mechanical Data

GGW (S–PBGA–N176) **PLASTIC BALL GRID ARRAY**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice
C. MicroStar BGA™ configuration.

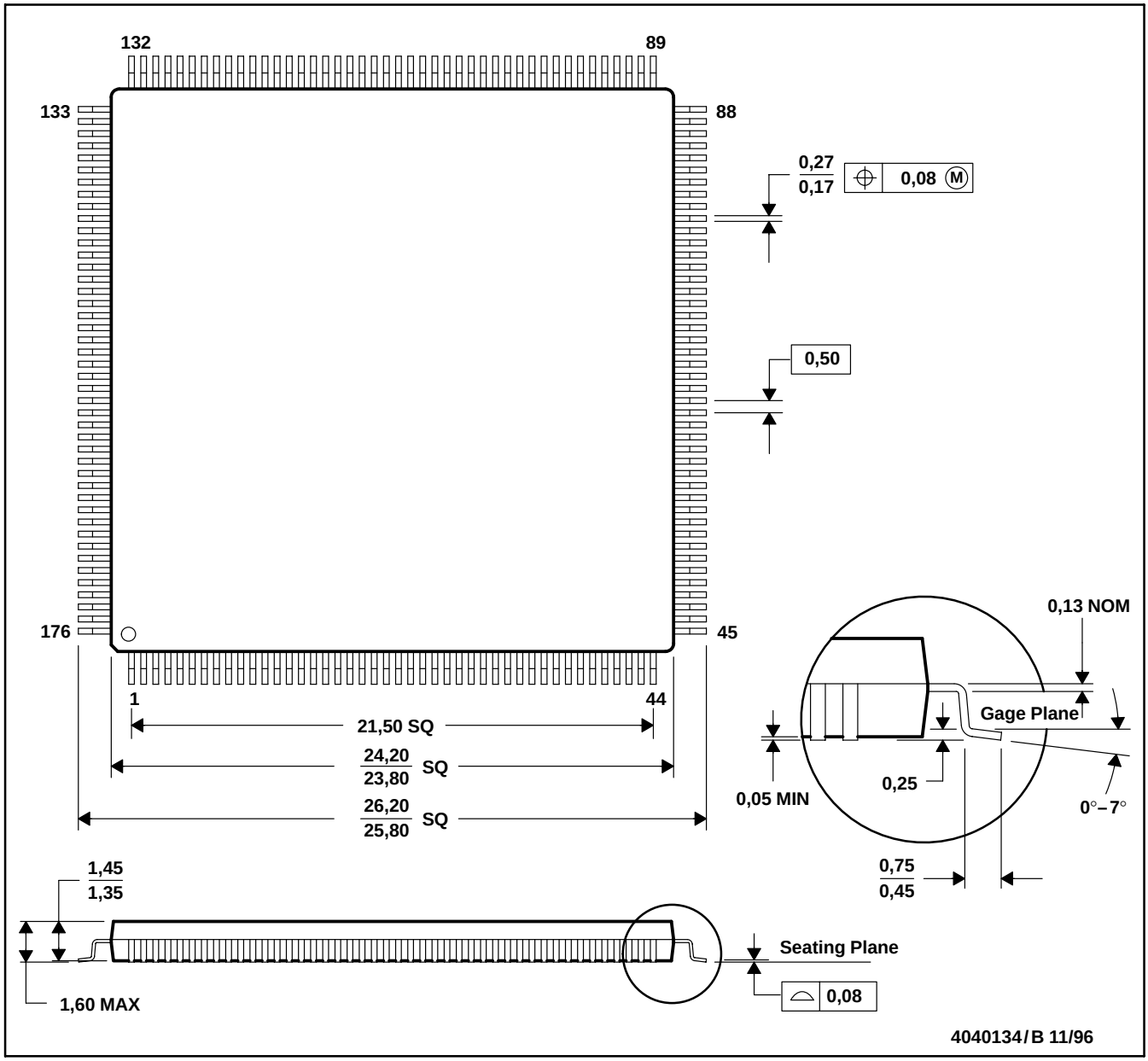
Figure 7–1. TMS320VC5501 176-Ball MicroStar BGA™ Plastic Ball Grid Array Package

PRODUCT PREVIEW

MicroStar BGA is a trademark of Texas Instruments.

7.2 Low-Profile Quad Flatpack Mechanical Data

PGF (S-PQFP-G176) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026

Figure 7–2. TMS320VC5501 176-Pin Low-Profile Quad Flatpack