

TMS320C6745/C6747 DSP System

Reference Guide



Literature Number: SPRUFK4D

March 2010

Preface	13
1 Overview	15
1.1 Introduction	16
1.2 Block Diagram	16
1.3 DSP Subsystem	16
2 DSP Subsystem	17
2.1 Introduction	18
2.2 TMS320C674x Megamodule	19
2.2.1 Internal Memory Controllers	19
2.2.2 Internal Peripherals	19
2.3 Memory Map	23
2.3.1 DSP Internal Memory	23
2.3.2 External Memory	23
2.4 Advanced Event Triggering (AET)	24
3 System Interconnect	25
3.1 Introduction	26
3.2 System Interconnect Block Diagram	27
4 System Memory	29
4.1 Introduction	30
4.2 DSP Memories	30
4.3 Shared RAM	30
4.4 External Memories	30
4.5 Internal Peripherals	30
4.6 Peripherals	31
5 Memory Protection Unit (MPU)	33
5.1 Introduction	34
5.1.1 Purpose of the MPU	34
5.1.2 Features	34
5.1.3 Block Diagram	34
5.1.4 MPU Default Configuration	35
5.2 Architecture	35
5.2.1 Privilege Levels	35
5.2.2 Memory Protection Ranges	36
5.2.3 Permission Structures	37
5.2.4 Protection Check	38
5.2.5 DSP L1/L2 Cache Controller Accesses	38
5.2.6 MPU Register Protection	38
5.2.7 Invalid Accesses and Exceptions	39
5.2.8 Reset Considerations	39
5.2.9 Interrupt Support	39
5.2.10 Emulation Considerations	39
5.3 MPU Registers	40
5.3.1 Revision Identification Register (REVID)	42
5.3.2 Configuration Register (CONFIG)	42

5.3.3	Interrupt Raw Status/Set Register (IRAWSTAT)	43
5.3.4	Interrupt Enable Status/Clear Register (IENSTAT)	44
5.3.5	Interrupt Enable Set Register (IENSET)	45
5.3.6	Interrupt Enable Clear Register (IENCLR)	45
5.3.7	Fixed Range Start Address Register (FXD_MPSAR)	46
5.3.8	Fixed Range End Address Register (FXD_MPEAR)	46
5.3.9	Fixed Range Memory Protection Page Attributes Register (FXD_MPPA)	47
5.3.10	Programmable Range <i>n</i> Start Address Registers (PROG _{<i>n</i>} _MPSAR)	48
5.3.11	Programmable Range <i>n</i> End Address Registers (PROG _{<i>n</i>} _MPEAR)	49
5.3.12	Programmable Range <i>n</i> Memory Protection Page Attributes Register (PROG _{<i>n</i>} _MPPA)	50
5.3.13	Fault Address Register (FLTADDRR)	51
5.3.14	Fault Status Register (FLTSTAT)	52
5.3.15	Fault Clear Register (FLTCLR)	53
6	Device Clocking	55
6.1	Overview	56
6.2	Frequency Flexibility	57
6.3	Peripheral Clocking	59
6.3.1	USB Clocking	59
6.3.2	EMIFB Clocking	61
6.3.3	EMIFA Clocking	63
6.3.4	EMAC Clocking	64
6.3.5	I/O Domains	66
7	Phase-Locked Loop Controller (PLL)	67
7.1	Introduction	68
7.2	PLL Control	68
7.2.1	Device Clock Generation	70
7.2.2	Steps for Changing PLL0 Domain Frequency	71
7.3	Locking/Unlocking PLL Register Access	72
7.4	PLL Registers	73
7.4.1	Revision Identification Register (REVID)	74
7.4.2	Reset Type Status Register (RSTYPE)	74
7.4.3	PLL Control Register (PLLCTL)	75
7.4.4	OBSCLK Select Register (OCSEL)	76
7.4.5	PLL Multiplier Control Register (PLLM)	77
7.4.6	PLL Pre-Divider Control Register (PREDIV)	77
7.4.7	PLL Controller Divider 1 Register (PLLDIV1)	78
7.4.8	PLL Controller Divider 2 Register (PLLDIV2)	78
7.4.9	PLL Controller Divider 3 Register (PLLDIV3)	79
7.4.10	PLL Controller Divider 4 Register (PLLDIV4)	79
7.4.11	PLL Controller Divider 5 Register (PLLDIV5)	80
7.4.12	PLL Controller Divider 6 Register (PLLDIV6)	80
7.4.13	PLL Controller Divider 7 Register (PLLDIV7)	81
7.4.14	Oscillator Divider 1 Register (OSCDIV)	82
7.4.15	PLL Post-Divider Control Register (POSTDIV)	83
7.4.16	PLL Controller Command Register (PLLCMD)	83
7.4.17	PLL Controller Status Register (PLLSTAT)	84
7.4.18	PLL Controller Clock Align Control Register (ALNCTL)	85
7.4.19	PLLDIV Ratio Change Status Register (DCHANGE)	86
7.4.20	Clock Enable Control Register (CKEN)	87
7.4.21	Clock Status Register (CKSTAT)	88
7.4.22	SYCLK Status Register (SYSTAT)	89
7.4.23	Emulation Performance Counter 0 Register (EMUCNT0)	90
7.4.24	Emulation Performance Counter 1 Register (EMUCNT1)	90

8	Power and Sleep Controller (PSC)	91
8.1	Introduction	92
8.2	Power Domain and Module Topology	92
8.2.1	Power Domain States	94
8.2.2	Module States	94
8.3	Executing State Transitions	96
8.3.1	Power Domain State Transitions	96
8.3.2	Module State Transitions	96
8.4	IcePick Emulation Support in the PSC	97
8.5	PSC Interrupts	97
8.5.1	Interrupt Events	97
8.5.2	Interrupt Registers	98
8.5.3	Interrupt Handling	99
8.6	PSC Registers	100
8.6.1	Revision Identification Register (REVID)	101
8.6.2	Interrupt Evaluation Register (INTEVAL)	101
8.6.3	PSC0 Module Error Pending Register 0 (modules 0-15) (MERRPR0)	102
8.6.4	PSC1 Module Error Pending Register 0 (modules 0-31) (MERRPR0)	102
8.6.5	PSC0 Module Error Clear Register 0 (modules 0-15) (MERRCR0)	103
8.6.6	PSC1 Module Error Clear Register 0 (modules 0-31) (MERRCR0)	103
8.6.7	Power Error Pending Register (PERRPR)	104
8.6.8	Power Error Clear Register (PERRCR)	104
8.6.9	Power Domain Transition Command Register (PTCMD)	105
8.6.10	Power Domain Transition Status Register (PTSTAT)	106
8.6.11	Power Domain 0 Status Register (PDSTAT0)	107
8.6.12	Power Domain 1 Status Register (PDSTAT1)	108
8.6.13	Power Domain 0 Control Register (PDCTL0)	109
8.6.14	Power Domain 1 Control Register (PDCTL1)	110
8.6.15	Power Domain 0 Configuration Register (PDCFG0)	111
8.6.16	Power Domain 1 Configuration Register (PDCFG1)	112
8.6.17	Module Status <i>n</i> Register (MDSTAT n)	113
8.6.18	PSC0 Module Control <i>n</i> Register (modules 0-15) (MDCTL n)	114
8.6.19	PSC1 Module Control <i>n</i> Register (modules 0-31) (MDCTL n)	115
9	Power Management	117
9.1	Introduction	118
9.2	Power Consumption Overview	118
9.3	Features	119
9.4	PSC and PLLC Overview	119
9.5	Clock Management	120
9.5.1	Module Clock ON/OFF	120
9.5.2	Module Clock Frequency Scaling	120
9.5.3	PLL Bypass and Power Down	120
9.6	DSP Sleep Mode Management	121
9.6.1	C674x DSP CPU Sleep Mode	121
9.6.2	C674x Megamodule Sleep Mode	121
9.7	RTC-Only Mode	121
9.8	Additional Peripheral Power Management Considerations	122
9.8.1	USB PHY Power Down Control	122
9.8.2	EMIFB Memory Clock Gating	122
10	System Configuration (SYSCFG) Module	123
10.1	Introduction	124
10.2	Protection	125

10.2.1	Requirements to Access SYSCFG Registers	126
10.3	Master Priority Control	127
10.4	Host-DSP Communication Interrupts	128
10.5	SYSCFG Registers	129
10.5.1	Revision Identification Register (REVID)	130
10.5.2	Device Identification Register 0 (DEVIDR0)	130
10.5.3	Boot Configuration Register (BOOTCFG)	131
10.5.4	Kick Registers (KICK0R-KICK1R)	132
10.5.5	Host 1 Configuration Register (HOST1CFG)	133
10.5.6	Interrupt Registers	134
10.5.7	Fault Registers	137
10.5.8	Master Priority Registers (MSTPRI0-MSTPRI2)	139
10.5.9	Pin Multiplexing Control Registers (PINMUX0-PINMUX19)	142
10.5.10	Suspend Source Register (SUSPSRC)	179
10.5.11	Chip Signal Register (CHIPSIG)	181
10.5.12	Chip Signal Clear Register (CHIPSIG_CLR)	182
10.5.13	Chip Configuration 0 Register (CFGCHIP0)	183
10.5.14	Chip Configuration 1 Register (CFGCHIP1)	184
10.5.15	Chip Configuration 2 Register (CFGCHIP2)	188
10.5.16	Chip Configuration 3 Register (CFGCHIP3)	190
10.5.17	Chip Configuration 4 Register (CFGCHIP4)	191
11	Boot Considerations	193
11.1	Introduction	194
A	Revision History	195

List of Figures

1-1.	TMS320C6745/C6747 DSP Block Diagram	16
2-1.	TMS320C674x Megamodule Block Diagram	18
3-1.	System Interconnect Block Diagram	27
5-1.	MPU Block Diagram.....	34
5-2.	Permission Fields.....	37
5-3.	Revision ID Register (REVID)	42
5-4.	Configuration Register (CONFIG)	42
5-5.	Interrupt Raw Status/Set Register (IRAWSTAT)	43
5-6.	Interrupt Enable Status/Clear Register (IENSTAT)	44
5-7.	Interrupt Enable Set Register (IENSET).....	45
5-8.	Interrupt Enable Clear Register (IENCLR)	45
5-9.	Fixed Range Start Address Register (FXD_MPSAR)	46
5-10.	Fixed Range End Address Register (FXD_MPEAR)	46
5-11.	Fixed Range Memory Protection Page Attributes Register (FXD_MPPA).....	47
5-12.	MPU1 Programmable Range <i>n</i> Start Address Register (PROG _{<i>n</i>} _MPSAR).....	48
5-13.	MPU2 Programmable Range <i>n</i> Start Address Register (PROG _{<i>n</i>} _MPSAR).....	48
5-14.	MPU1 Programmable Range <i>n</i> End Address Register (PROG _{<i>n</i>} _MPEAR).....	49
5-15.	MPU2 Programmable Range <i>n</i> End Address Register (PROG _{<i>n</i>} _MPEAR).....	49
5-16.	Programmable Range Memory Protection Page Attributes Register (PROG _{<i>n</i>} _MPPA).....	50
5-17.	Fault Address Register (FLTADDR)	51
5-18.	Fault Status Register (FLTSTAT)	52
5-19.	Fault Clear Register (FLTCLR)	53
6-1.	Overall Clocking Diagram	57
6-2.	USB Clocking Diagram	59
6-3.	EMIFB Clocking Diagram.....	62
6-4.	EMIFA Clocking Diagram.....	63
6-5.	EMAC Clocking Diagram	64
7-1.	PLL0 Structure	69
7-2.	Revision Identification Register (REVID)	74
7-3.	Reset Type Status Register (RSTYPE)	74
7-4.	PLL Control Register (PLLCTL)	75
7-5.	OBSCLK Select Register (OCSEL)	76
7-6.	PLL Multiplier Control Register (PLLM).....	77
7-7.	PLL Pre-Divider Control Register (PREDIV)	77
7-8.	PLL Controller Divider 1 Register (PLLDIV1)	78
7-9.	PLL Controller Divider 2 Register (PLLDIV2)	78
7-10.	PLL Controller Divider 3 Register (PLLDIV3)	79
7-11.	PLL Controller Divider 4 Register (PLLDIV4)	79
7-12.	PLL Controller Divider 5 Register (PLLDIV5)	80
7-13.	PLL Controller Divider 6 Register (PLLDIV6).....	80
7-14.	PLL Controller Divider 7 Register (PLLDIV7)	81
7-15.	Oscillator Divider 1 Register (OSCDIV)	82
7-16.	PLL Post-Divider Control Register (POSTDIV).....	83
7-17.	PLL Controller Command Register (PLLCMD)	83
7-18.	PLL Controller Status Register (PLLSTAT)	84
7-19.	PLL Controller Clock Align Control Register (ALNCTL)	85
7-20.	PLLDIV Ratio Change Status Register (DCHANGE).....	86

7-21.	Clock Enable Control Register (CKEN).....	87
7-22.	Clock Status Register (CKSTAT).....	88
7-23.	SYSCCLK Status Register (SYSTAT).....	89
7-24.	Emulation Performance Counter 0 Register (EMUCNT0)	90
7-25.	Emulation Performance Counter 1 Register (EMUCNT1)	90
8-1.	Revision Identification Register (REVID)	101
8-2.	Interrupt Evaluation Register (INTEVAL)	101
8-3.	PSC0 Module Error Pending Register 0 (MERRPR0)	102
8-4.	PSC1 Module Error Pending Register 0 (MERRPR0)	102
8-5.	PSC0 Module Error Clear Register 0 (MERRCR0).....	103
8-6.	PSC1 Module Error Clear Register 0 (MERRCR0).....	103
8-7.	Power Error Pending Register (PERRPR).....	104
8-8.	Power Error Clear Register (PERRCR)	104
8-9.	Power Domain Transition Command Register (PTCMD).....	105
8-10.	Power Domain Transition Status Register (PTSTAT).....	106
8-11.	Power Domain 0 Status Register (PDSTAT0)	107
8-12.	Power Domain 1 Status Register (PDSTAT1)	108
8-13.	Power Domain 0 Control Register (PDCTL0)	109
8-14.	Power Domain 1 Control Register (PDCTL1)	110
8-15.	Power Domain 0 Configuration Register (PDCFG0)	111
8-16.	Power Domain 1 Configuration Register (PDCFG1)	112
8-17.	Module Status <i>n</i> Register (MDSTAT <i>n</i>).....	113
8-18.	PSC0 Module Control <i>n</i> Register (MDCTL <i>n</i>).....	114
8-19.	PSC1 Module Control <i>n</i> Register (MDCTL <i>n</i>).....	115
10-1.	Revision Identification Register (REVID)	130
10-2.	Device Identification Register 0 (DEVIDR0).....	130
10-3.	Boot Configuration Register (BOOTCFG)	131
10-4.	Kick 0 Register (KICK0R).....	132
10-5.	Kick 1 Register (KICK1R).....	132
10-6.	Host 1 Configuration Register (HOST1CFG).....	133
10-7.	Interrupt Raw Status/Set Register (IRAWSTAT).....	134
10-8.	Interrupt Enable Status/Clear Register (IENSTAT).....	135
10-9.	Interrupt Enable Register (IENSET)	136
10-10.	Interrupt Enable Clear Register (IENCLR).....	136
10-11.	End of Interrupt Register (EOI).....	137
10-12.	Fault Address Register (FLTADDR)	137
10-13.	Fault Status Register (FLTSTAT).....	138
10-14.	Master Priority 0 Register (MSTPRI0).....	139
10-15.	Master Priority 1 Register (MSTPRI1).....	140
10-16.	Master Priority 2 Register (MSTPRI2).....	141
10-17.	Pin Multiplexing Control 0 Register (PINMUX0)	142
10-18.	Pin Multiplexing Control 1 Register (PINMUX1)	144
10-19.	Pin Multiplexing Control 2 Register (PINMUX2)	146
10-20.	Pin Multiplexing Control 3 Register (PINMUX3)	148
10-21.	Pin Multiplexing Control 4 Register (PINMUX4)	149
10-22.	Pin Multiplexing Control 5 Register (PINMUX5)	150
10-23.	Pin Multiplexing Control 6 Register (PINMUX6)	152
10-24.	Pin Multiplexing Control 7 Register (PINMUX7)	154
10-25.	Pin Multiplexing Control 8 Register (PINMUX8)	156

10-26. Pin Multiplexing Control 9 Register (PINMUX9)	158
10-27. Pin Multiplexing Control 10 Register (PINMUX10)	160
10-28. Pin Multiplexing Control 11 Register (PINMUX11)	162
10-29. Pin Multiplexing Control 12 Register (PINMUX12)	164
10-30. Pin Multiplexing Control 13 Register (PINMUX13)	166
10-31. Pin Multiplexing Control 14 Register (PINMUX14)	168
10-32. Pin Multiplexing Control 15 Register (PINMUX15)	170
10-33. Pin Multiplexing Control 16 Register (PINMUX16)	172
10-34. Pin Multiplexing Control 17 Register (PINMUX17)	174
10-35. Pin Multiplexing Control 18 Register (PINMUX18)	176
10-36. Pin Multiplexing Control 19 Register (PINMUX19)	178
10-37. Suspend Source Register (SUSPSRC)	179
10-38. Chip Signal Register (CHIPSIG)	181
10-39. Chip Signal Clear Register (CHIPSIG_CLR)	182
10-40. Chip Configuration 0 Register (CFGCHIP0).....	183
10-41. Chip Configuration 1 Register (CFGCHIP1).....	184
10-42. Chip Configuration 2 Register (CFGCHIP2).....	188
10-43. Chip Configuration 3 Register (CFGCHIP3).....	190
10-44. Chip Configuration 4 Register (CFGCHIP4).....	191

List of Tables

2-1.	DSP Interrupt Map	19
3-1.	TMS320C6745/C6747 DSP System Interconnect Matrix	26
5-1.	MPU Memory Regions.....	35
5-2.	MPU Default Configuration.....	35
5-3.	Device Master Settings	36
5-4.	Request Type Access Controls.....	37
5-5.	MPU_BOOTCFG_ERR Interrupt Sources	39
5-6.	Memory Protection Unit 1 (MPU1) Registers	40
5-7.	Memory Protection Unit 2 (MPU2) Registers	40
5-8.	Revision ID Register (REVID) Field Descriptions	42
5-9.	Configuration Register (CONFIG) Field Descriptions.....	42
5-10.	Interrupt Raw Status/Set Register (IRAWSTAT) Field Descriptions.....	43
5-11.	Interrupt Enable Status/Clear Register (IENSTAT) Field Descriptions.....	44
5-12.	Interrupt Enable Set Register (IENSET) Field Descriptions	45
5-13.	Interrupt Enable Clear Register (IENCLR) Field Descriptions.....	45
5-14.	Fixed Range Memory Protection Page Attributes Register (FXD_MPPA) Field Descriptions	47
5-15.	MPU1 Programmable Range <i>n</i> Start Address Register (PROG _{<i>n</i>} _MPSAR) Field Descriptions	48
5-16.	MPU2 Programmable Range <i>n</i> Start Address Register (PROG _{<i>n</i>} _MPSAR) Field Descriptions	48
5-17.	MPU1 Programmable Range <i>n</i> End Address Register (PROG _{<i>n</i>} _MPEAR) Field Descriptions	49
5-18.	MPU2 Programmable Range <i>n</i> End Address Register (PROG _{<i>n</i>} _MPEAR) Field Descriptions	49
5-19.	Programmable Range Memory Protection Page Attributes Register (PROG _{<i>n</i>} _MPPA) Field Descriptions ...	50
5-20.	Fault Address Register (FLTADDRR) Field Descriptions	51
5-21.	Fault Status Register (FLTSTAT) Field Descriptions	52
5-22.	Fault Clear Register (FLTCLR) Field Descriptions.....	53
6-1.	Device Clock Inputs	56
6-2.	System Clock Domains	56
6-3.	Example PLL Frequencies	58
6-4.	USB Clock Multiplexing Options.....	60
6-5.	EMIFB MCLK Frequencies.....	62
6-6.	EMIFA Frequencies	63
6-7.	EMAC Reference Clock Frequencies.....	65
6-8.	Peripherals	66
7-1.	System PLLC0 Output Clocks	70
7-2.	PLL Controller (PLLC) Registers	73
7-3.	Revision Identification Register (REVID) Field Descriptions.....	74
7-4.	Reset Type Status Register (RSTYPE) Field Descriptions	74
7-5.	PLL Control Register (PLLCTL) Field Descriptions	75
7-6.	OBSCLK Select Register (OCSEL) Field Descriptions.....	76
7-7.	PLL Multiplier Control Register (PLLM) Field Descriptions	77
7-8.	PLL Pre-Divider Control Register (PREDIV) Field Descriptions.....	77
7-9.	PLL Controller Divider 1 Register (PLLDIV1) Field Descriptions.....	78
7-10.	PLL Controller Divider 2 Register (PLLDIV2) Field Descriptions.....	78
7-11.	PLL Controller Divider 3 Register (PLLDIV3) Field Descriptions.....	79
7-12.	PLL Controller Divider 4 Register (PLLDIV4) Field Descriptions.....	79
7-13.	PLL Controller Divider 5 Register (PLLDIV5) Field Descriptions.....	80
7-14.	PLL Controller Divider 6 Register (PLLDIV6) Field Descriptions.....	80
7-15.	PLL Controller Divider 7 Register (PLLDIV7) Field Descriptions.....	81

7-16.	Oscillator Divider 1 Register (OSCDIV) Field Descriptions	82
7-17.	PLL Post-Divider Control Register (POSTDIV) Field Descriptions.....	83
7-18.	PLL Controller Command Register (PLLCMD) Field Descriptions.....	83
7-19.	PLL Controller Status Register (PLLSTAT) Field Descriptions.....	84
7-20.	PLL Controller Clock Align Control Register (ALNCTL) Field Descriptions.....	85
7-21.	PLLDIV Ratio Change Status Register (DCHANGE) Field Descriptions.....	86
7-22.	Clock Enable Control Register (CKEN) Field Descriptions	87
7-23.	Clock Status Register (CKSTAT) Field Descriptions	88
7-24.	SYSCCLK Status Register (SYSTAT) Field Descriptions.....	89
7-25.	Emulation Performance Counter 0 Register (EMUCNT0) Field Descriptions	90
7-26.	Emulation Performance Counter 1 Register (EMUCNT1) Field Descriptions	90
8-1.	PSC0 Default Module Configuration	92
8-2.	PSC1 Default Module Configuration	93
8-3.	Module States.....	95
8-4.	IcePick Emulation Commands	97
8-5.	PSC Interrupt Events	97
8-6.	Power and Sleep Controller 0 (PSC0) Registers	100
8-7.	Power and Sleep Controller 1 (PSC1) Registers	100
8-8.	Revision Identification Register (REVID) Field Descriptions	101
8-9.	Interrupt Evaluation Register (INTEVAL) Field Descriptions	101
8-10.	PSC0 Module Error Pending Register 0 (MERRPR0) Field Descriptions	102
8-11.	PSC0 Module Error Clear Register 0 (MERRCR0) Field Descriptions	103
8-12.	Power Error Pending Register (PERRPR) Field Descriptions	104
8-13.	Power Error Clear Register (PERRCR) Field Descriptions.....	104
8-14.	Power Domain Transition Command Register (PTCMD) Field Descriptions	105
8-15.	Power Domain Transition Status Register (PTSTAT) Field Descriptions	106
8-16.	Power Domain 0 Status Register (PDSTAT0) Field Descriptions	107
8-17.	Power Domain 1 Status Register (PDSTAT1) Field Descriptions	108
8-18.	Power Domain 0 Control Register (PDCTL0) Field Descriptions	109
8-19.	Power Domain 1 Control Register (PDCTL1) Field Descriptions	110
8-20.	Power Domain 0 Configuration Register (PDCFG0) Field Descriptions	111
8-21.	Power Domain 1 Configuration Register (PDCFG1) Field Descriptions	112
8-22.	Module Status <i>n</i> Register (MDSTAT <i>n</i>) Field Descriptions	113
8-23.	PSC0 Module Control <i>n</i> Register (MDCTL <i>n</i>) Field Descriptions	114
8-24.	PSC1 Module Control <i>n</i> Register (MDCTL <i>n</i>) Field Descriptions	115
9-1.	Power Management Features.....	119
10-1.	System Configuration (SYSCFG) Module Register Access	125
10-2.	Master IDs	127
10-3.	Default Master Priority.....	128
10-4.	System Configuration Module (SYSCFG) Registers	129
10-5.	Revision Identification Register (REVID) Field Descriptions	130
10-6.	Device Identification Register 0 (DEVIDR0) Field Descriptions	130
10-7.	Boot Configuration Register (BOOTCFG) Field Descriptions	131
10-8.	Kick 0 Register (KICK0R) Field Descriptions.....	132
10-9.	Kick 1 Register (KICK1R) Field Descriptions.....	132
10-10.	Host 1 Configuration Register (HOST1CFG) Field Descriptions	133
10-11.	Interrupt Raw Status/Set Register (IRAWSTAT) Field Descriptions	134
10-12.	Interrupt Enable Status/Clear Register (IENSTAT) Field Descriptions	135
10-13.	Interrupt Enable Register (IENSET) Field Descriptions.....	136

10-14. Interrupt Enable Clear Register (IENCLR) Field Descriptions	136
10-15. End of Interrupt Register (EOI) Field Descriptions	137
10-16. Fault Address Register (FLTADDR) Field Descriptions.....	137
10-17. Fault Status Register (FLTSTAT) Field Descriptions.....	138
10-18. Master Priority 0 Register (MSTPRI0) Field Descriptions	139
10-19. Master Priority 1 Register (MSTPRI1) Field Descriptions	140
10-20. Master Priority 2 Register (MSTPRI2) Field Descriptions	141
10-21. Pin Multiplexing Control 0 Register (PINMUX0) Field Descriptions	142
10-22. Pin Multiplexing Control 1 Register (PINMUX1) Field Descriptions	144
10-23. Pin Multiplexing Control 2 Register (PINMUX2) Field Descriptions	146
10-24. Pin Multiplexing Control 3 Register (PINMUX3) Field Descriptions	148
10-25. Pin Multiplexing Control 4 Register (PINMUX4) Field Descriptions	149
10-26. Pin Multiplexing Control 5 Register (PINMUX5) Field Descriptions	150
10-27. Pin Multiplexing Control 6 Register (PINMUX6) Field Descriptions	152
10-28. Pin Multiplexing Control 7 Register (PINMUX7) Field Descriptions	154
10-29. Pin Multiplexing Control 8 Register (PINMUX8) Field Descriptions	156
10-30. Pin Multiplexing Control 9 Register (PINMUX9) Field Descriptions	158
10-31. Pin Multiplexing Control 10 Register (PINMUX10) Field Descriptions	160
10-32. Pin Multiplexing Control 11 Register (PINMUX11) Field Descriptions	162
10-33. Pin Multiplexing Control 12 Register (PINMUX12) Field Descriptions	164
10-34. Pin Multiplexing Control 13 Register (PINMUX13) Field Descriptions	166
10-35. Pin Multiplexing Control 14 Register (PINMUX14) Field Descriptions	168
10-36. Pin Multiplexing Control 15 Register (PINMUX15) Field Descriptions	170
10-37. Pin Multiplexing Control 16 Register (PINMUX16) Field Descriptions	172
10-38. Pin Multiplexing Control 17 Register (PINMUX17) Field Descriptions	174
10-39. Pin Multiplexing Control 18 Register (PINMUX18) Field Descriptions	176
10-40. Pin Multiplexing Control 19 Register (PINMUX19) Field Descriptions	178
10-41. Suspend Source Register (SUSPSRC) Field Descriptions.....	179
10-42. Chip Signal Register (CHIPSIG) Field Descriptions.....	181
10-43. Chip Signal Clear Register (CHIPSIG_CLR) Field Descriptions.....	182
10-44. Chip Configuration 0 Register (CFGCHIP0) Field Descriptions	183
10-45. Chip Configuration 1 Register (CFGCHIP1) Field Descriptions	185
10-46. Chip Configuration 2 Register (CFGCHIP2) Field Descriptions	188
10-47. Chip Configuration 3 Register (CFGCHIP3) Field Descriptions	190
10-48. Chip Configuration 4 Register (CFGCHIP4) Field Descriptions	191
A-1. Document Revision History	195

Read This First

About This Manual

Describes the System-on-Chip (SoC) system. The SoC system includes TI's standard TMS320C674x Megamodule and several blocks of internal memory (L1P, L1D, and L2). This document provides an overview of the system and the following considerations associated with it:

- DSP subsystem
- System interconnect
- System memory
- Memory protection unit (MPU)
- Device clocking
- Phase-locked loop controller (PLL)
- Power and sleep controller (PSC)
- Power management
- System configuration (SYSCFG) module
- Boot considerations

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
 - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
 - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documentation From Texas Instruments

Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

The current documentation that describes related peripherals and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

[SPRUFK9](#)— *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide*. Provides an overview and briefly describes the peripherals available on the TMS320C674x Digital Signal Processors (DSPs) and OMAP-L1x Applications Processors.

[SPRUFK5](#)— *TMS320C674x DSP Megamodule Reference Guide*. Describes the TMS320C674x digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

[SPRUFEB](#)— *TMS320C674x DSP CPU and Instruction Set Reference Guide*. Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C674x digital signal processors (DSPs). The C674x DSP is an enhancement of the C64x+ and C67x+ DSPs with added functionality and an expanded instruction set.

[SPRUG82](#)— ***TMS320C674x DSP Cache User's Guide***. Explains the fundamentals of memory caches and describes how the two-level cache-based internal memory architecture in the TMS320C674x digital signal processor (DSP) can be efficiently used in DSP applications. Shows how to maintain coherence with external memory, how to use DMA to reduce memory latencies, and how to optimize your code to improve cache efficiency. The internal memory architecture in the C674x DSP is organized in a two-level hierarchy consisting of a dedicated program cache (L1P) and a dedicated data cache (L1D) on the first level. Accesses by the CPU to these first level caches can complete without CPU pipeline stalls. If the data requested by the CPU is not contained in cache, it is fetched from the next lower memory level, L2 or external memory.

Overview

Topic	Page
1.1 Introduction	16
1.2 Block Diagram	16
1.3 DSP Subsystem	16

1.1 Introduction

The C6745/C6747 DSP efficiently handles communication and audio processing tasks. The C6745/C6747 DSP consists of the following primary components:

- DSP and associated memories
- A set of I/O peripherals
- A powerful DMA subsystem and SDRAM EMIF interface

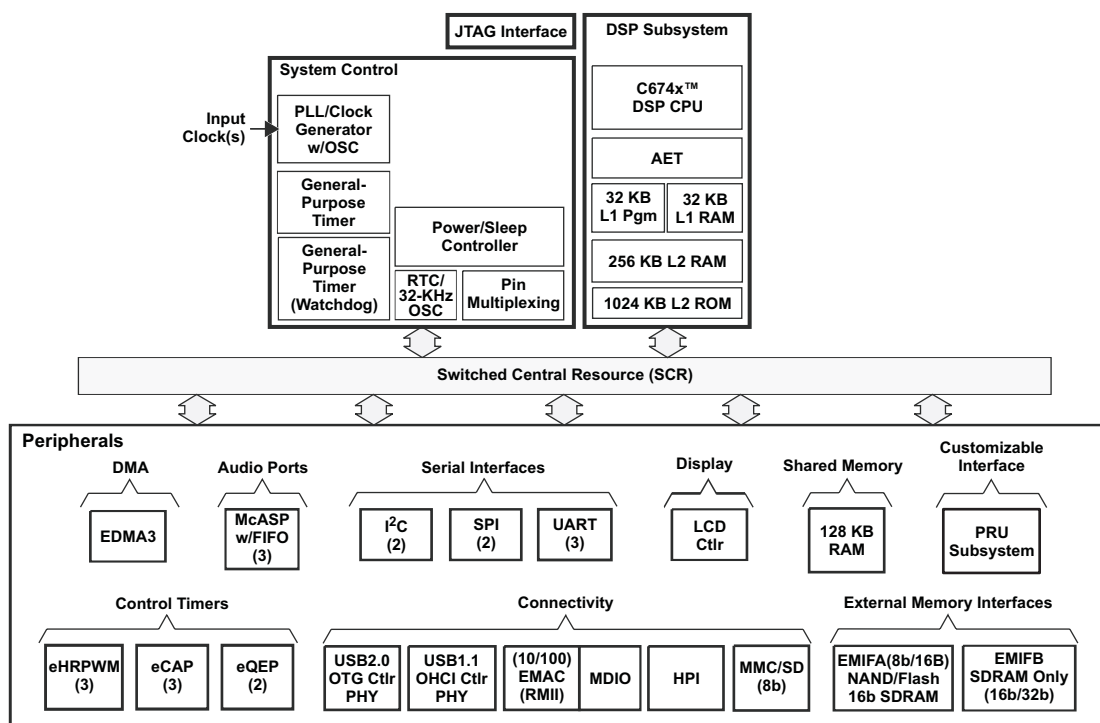
1.2 Block Diagram

A block diagram for the C6745/C6747 DSP is shown in [Figure 1-1](#).

1.3 DSP Subsystem

The DSP subsystem (DSPSS) includes TI's standard TMS320C674x megamodule and several blocks of internal memory (L1P, L1D, and L2). [Chapter 2](#) describes the DSPSS components.

Figure 1-1. TMS320C6745/C6747 DSP Block Diagram



Note: Not all peripherals are available at the same time due to multiplexing. The RTC, McASP2, LCDC, USB1.1, HPI, and 128 KB RAM are not supported on the C6745 DSP.

DSP Subsystem

Topic	Page
2.1 Introduction	18
2.2 TMS320C674x Megamodule	19
2.3 Memory Map	23
2.4 Advanced Event Triggering (AET)	24

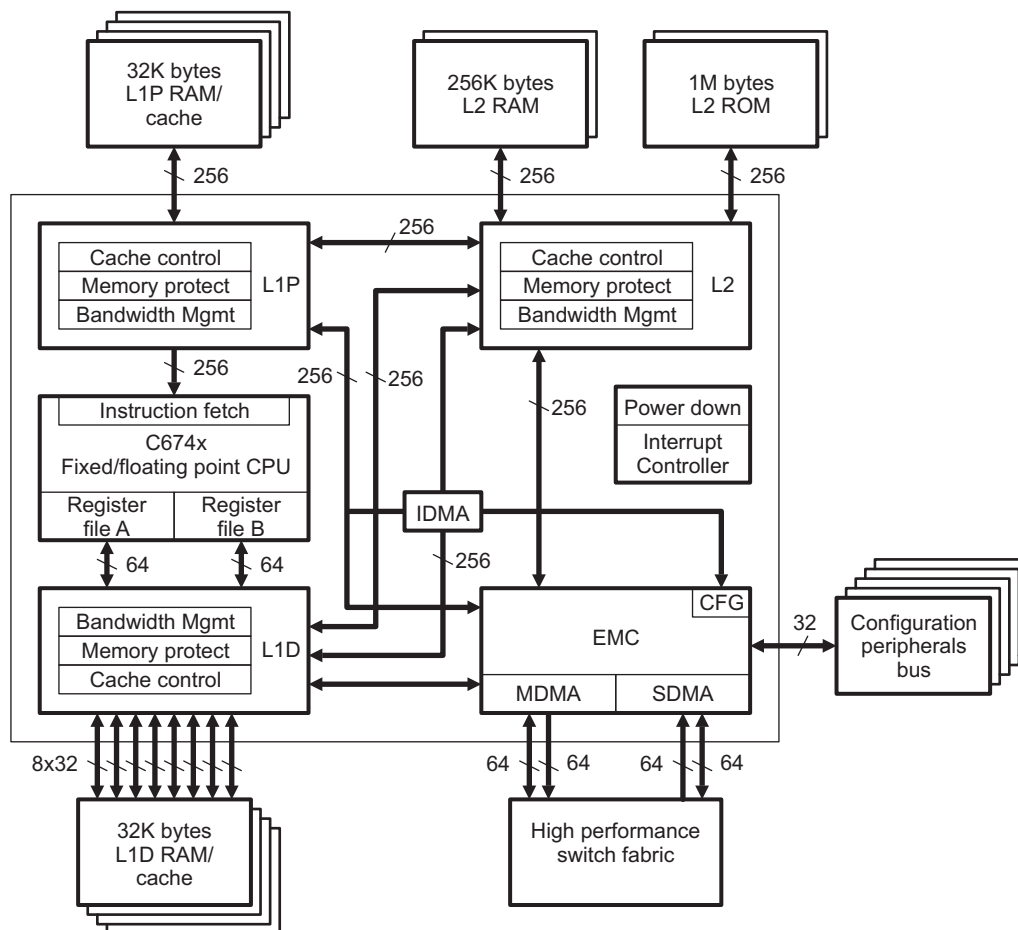
2.1 Introduction

The DSP subsystem (Figure 2-1) includes TI's standard TMS320C674x megamodule and several blocks of internal memory (L1P, L1D, and L2). This document provides an overview of the DSP subsystem and the following considerations associated with it:

- Memory mapping
- Interrupts
- Power management

For more information, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5), the *TMS320C674x DSP CPU and Instruction Set Reference Guide* (SPRUF8), and the *TMS320C674x DSP Cache User's Guide* (SPRUG82).

Figure 2-1. TMS320C674x Megamodule Block Diagram



2.2 TMS320C674x Megamodule

The C674x megamodule (Figure 2-1) consists of the following components:

- TMS320C674x CPU
- Internal memory controllers:
 - Program memory controller (PMC)
 - Data memory controller (DMC)
 - Unified memory controller (UMC)
 - External memory controller (EMC)
 - Internal direct memory access (IDMA) controller
- Internal peripherals:
 - Interrupt controller (INTC)
 - Power-down controller (PDC)
 - Bandwidth manager (BWM)
- Advanced event triggering (AET)

2.2.1 Internal Memory Controllers

The C674x megamodule implements a two-level internal cache-based memory architecture with external memory support. Level 1 memory (L1) is split into separate program memory (L1P memory) and data memory (L1D memory). L1 memory is accessible to the CPU without stalls. Level 2 memory (L2) can also be split into L2 RAM (normal addressable on-chip memory) and L2 cache for caching external memory locations. The internal direct memory access controller (IDMA) manages DMA among the L1P, L1D, and L2 memories.

For more information about each of these controllers, see the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

2.2.2 Internal Peripherals

The C674x megamodule includes the following internal peripherals:

- DSP interrupt controller (INTC)
- DSP power-down controller (PDC)
- Bandwidth manager (BWM)
- Internal DMA (IDMA) controller

This section briefly describes the INTC, PDC, BWM, and IDMA controller. For more information on these internal peripherals, see the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

2.2.2.1 Interrupt Controller (INTC)

The C674x megamodule includes an interrupt controller (INTC) to manage CPU interrupts. The INTC maps DSP device events to 12 CPU interrupts. All DSP device events are listed in [Table 2-1](#). The INTC is fully described in the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

Table 2-1. DSP Interrupt Map

Event	Interrupt Name	Source
0	EVT0	C674x Interrupt Control 0
1	EVT1	C674x Interrupt Control 1
2	EVT2	C674x Interrupt Control 2
3	EVT3	C674x Interrupt Control 3
4	T64P0_TINT12	Timer64P0 - TINT12
5	SYSCFG_CHIPINT2	SYSCFG CHIPSIG Register
6	—	Reserved

Table 2-1. DSP Interrupt Map (continued)

Event	Interrupt Name	Source
7	EHRPWM0	HiResTimer/PWM0 Interrupt
8	TPCC0_INT1	TPCC0 Region 1 Interrupt
9	EMU-DTDMA	C674x-ECM
10	EHRPWM0TZ	HiResTimer/PWM0 Trip Zone Interrupt
11	EMU-RTDXRX	C674x-RTDX
12	EMU-RTDXTX	C674x-RTDX
13	IDMAINT0	C674x-EMC
14	IDMAINT1	C674x-EMC
15	MMCS0_INT0	MMCS0 MMC/SD Interrupt
16	MMCS0_INT1	MMCS0 SDIO Interrupt
17	—	Reserved
18	EHRPWM1	HiResTimer/PWM1 Interrupt
19	USB0_INT	USB0 (USB2.0) Interrupt
20	USB1_HCINT ⁽¹⁾	USB1 (USB1.1) OHCI Host Controller Interrupt
21	USB1_R/WAKEUP ⁽¹⁾	USB1 (USB1.1) Remote Wakeup Interrupt
22	—	Reserved
23	EHRPWM1TZ	HiResTimer/PWM1 Trip Zone Interrupt
24	EHRPWM2	HiResTimer/PWM2 Interrupt
25	EHRPWM2TZ	HiResTimer/PWM2 Trip Zone Interrupt
26	EMAC_C0RXTHRESH	EMAC - Core 0 Receive Threshold Interrupt
27	EMAC_C0RX	EMAC - Core 0 Receive Interrupt
28	EMAC_C0TX	EMAC - Core 0 Transmit Interrupt
29	EMAC_C0MISC	EMAC - Core 0 Miscellaneous Interrupt
30	EMAC_C1RXTHRESH	EMAC - Core 1 Receive Threshold Interrupt
31	EMAC_C1RX	EMAC - Core 1 Receive Interrupt
32	EMAC_C1TX	EMAC - Core 1 Transmit Interrupt
33	EMAC_C1MISC	EMAC - Core 1 Miscellaneous Interrupt
34	UHPI_DSPINT ⁽¹⁾	HPI DSP Interrupt
35	—	Reserved
36	IIC0_INT	I2C0
37	SPI0_INT	SPI0
38	UART0_INT	UART0
39	—	Reserved
40	T64P1_TINT12	Timer64P1 Interrupt 12
41	GPIO_B1INT	GPIO Bank 1 Interrupt
42	IIC1_INT	I2C1
43	SPI1_INT	SPI1
44	—	Reserved
45	ECAP0	ECAP0
46	UART_INT1	UART1
47	ECAP1	ECAP1
48	T64P1_TINT34	Timer64P1 Interrupt 34
49	GPIO_B2INT	GPIO Bank 2 Interrupt
50	—	Reserved
51	ECAP2	ECAP2
52	GPIO_B3INT	GPIO Bank 3 Interrupt

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

Table 2-1. DSP Interrupt Map (continued)

Event	Interrupt Name	Source
53	EQEP1	EQEP1
54	GPIO_B4INT	GPIO Bank 4 Interrupt
55	EMIFA_INT	EMIFA
56	EDMA3_CC0_ERRINT	EDMA3 Channel Controller 0
57	EDMA3_TC0_ERRINT	EDMA3 Transfer Controller 0
58	EDMA3_TC1_ERRINT	EDMA3 Transfer Controller 1
59	GPIO_B5INT	GPIO Bank 5 Interrupt
60	EMIFB_INT	EMIFB Memory Error Interrupt
61	MCASP_INT	McASP0,1,2 Combined RX/TX Interrupts
62	GPIO_B6INT	GPIO Bank 6 Interrupt
63	RTC_IRQS ⁽¹⁾	RTC Combined
64	T64P0_TINT34	Timer64P0 Interrupt 34
65	GPIO_B0INT	GPIO Bank 0 Interrupt
66	—	Reserved
67	SYSCFG_CHIPINT3	SYSCFG CHIPSIG Register
68	EQEP0	EQEP0
69	UART2_INT	UART2
70	PSC0_ALLINT	PSC0
71	PSC1_ALLINT	PSC1
72	GPIO_B7INT	GPIO Bank 7 Interrupt
73	LCDC_INT ⁽²⁾	LCD Controller
74	MPU_BOOTCFG_ERR	MPU Shared Interrupt
75-77	—	Reserved
78	T64P0_CMPINT0	Timer64P0 - Compare 0
79	T64P0_CMPINT1	Timer64P0 - Compare 1
80	T64P0_CMPINT2	Timer64P0 - Compare 2
81	T64P0_CMPINT3	Timer64P0 - Compare 3
82	T64P0_CMPINT4	Timer64P0 - Compare 4
83	T64P0_CMPINT5	Timer64P0 - Compare 5
84	T64P0_CMPINT6	Timer64P0 - Compare 6
85	T64P0_CMPINT7	Timer64P0 - Compare 7
86	T64P1_CMPINT0	Timer64P1 - Compare 0
87	T64P1_CMPINT1	Timer64P1 - Compare 1
88	T64P1_CMPINT2	Timer64P1 - Compare 2
89	T64P1_CMPINT3	Timer64P1 - Compare 3
90	T64P1_CMPINT4	Timer64P1 - Compare 4
91	T64P1_CMPINT5	Timer64P1 - Compare 5
92	T64P1_CMPINT6	Timer64P1 - Compare 6
93	T64P1_CMPINT7	Timer64P1 - Compare 7
94-95	—	Reserved
96	INTERR	C674x-Interrupt Control
97	EMC_IDMAERR	C674x-EMC
98-112	—	Reserved
113	PMC_ED	C674x-PMC
114-115	—	Reserved
116	UMC_ED1	C674x-UMC

⁽²⁾ This peripheral is not supported on the C6745 DSP.

Table 2-1. DSP Interrupt Map (continued)

Event	Interrupt Name	Source
117	UMC_ED2	C674x-UMC
118	PDC_INT	C674x-PDC
119	SYS_CMPA	C674x-SYS
120	PMC_CMPA	C674x-PMC
121	PMC_CMPA	C674x-PMC
122	DMC_CMPA	C674x-DMC
123	DMC_CMPA	C674x-DMC
124	UMC_CMPA	C674x-UMC
125	UMC_CMPA	C674x-UMC
126	EMC_CMPA	C674x-EMC
127	EMC_BUSERR	C674x-EMC

2.2.2.1.1 Interrupt Controller Registers

For more information on the DSP interrupt controller (INTC) registers, see the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

2.2.2.1.2 NMI Interrupt

In addition to the interrupts listed in [Table 2-1](#), the DSP also supports a special interrupt that behaves more like an exception, non-maskable interrupt (NMI). The NMI interrupt is controlled by two registers in the System Configuration Module, the chip signal register (CHIPSIG) and the chip signal clear register (CHIPSIG_CLR).

The NMI interrupt is asserted by writing a 1 to the CHIPSIG4 bit in CHIPSIG. The NMI interrupt is cleared by writing a 1 to the CHIPSIG4 bit in CHIPSIG_CLR. For more information on the System Configuration Module, CHIPSIG, and CHIPSIG_CLR, see [Chapter 10](#).

2.2.2.2 Power-Down Controller (PDC)

The C674x megamodule includes a power-down controller (PDC). The PDC can power-down all of the following components of the C674x megamodule and internal memories of the DSP subsystem:

- C674x CPU
- Program memory controller (PMC)
- Data memory controller (DMC)
- Unified memory controller (UMC)
- Extended memory controller (EMC)
- Internal Direct Memory Access controller (IDMA)
- L1P memory
- L1D memory
- L2 memory

This device supports the static power-down feature from the C674x megamodule. The *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)) describes the power-down control in more detail.

- Static power-down: The PDC initiates power-down (clock gating) of the entire C674x megamodule and all internal memories immediately upon command from software.

Static power-down (clock gating) affects all components of the C674x megamodule and all internal memories. Software can initiate static power-down by way of a register bit in the power-down controller command register (PDCCMD) of the PDC. For more information on the PDC, see the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

2.2.2.3 Bandwidth Manager (BWM)

The bandwidth manager (BWM) provides a programmable interface for optimizing bandwidth among the requesters for resources, which include the following:

- EDMA-initiated DMA transfers (and resulting coherency operations)
- IDMA-initiated transfers (and resulting coherency operations)
- Programmable cache coherency operations
 - Block based coherency operations
 - Global coherency operations
- CPU direct-initiated transfers
 - Data access (load/store)
 - Program access

The resources include the following:

- L1P memory
- L1D memory
- L2 memory
- Resources outside of the C674x megamodule: external memory, on-chip peripherals, registers

Since any given requester could potentially block a resource for extended periods of time, the bandwidth manager is implemented to assure fairness for all requesters.

The bandwidth manager implements a weighted-priority-driven bandwidth allocation. Each requester (EDMA, IDMA, CPU, etc.) is assigned a priority level on a per-transfer basis. The programmable priority level has a single meaning throughout the system. There are a total of nine priority levels, where priority zero is the highest priority and priority eight is the lowest priority. When requests for a single resource contend, access is granted to the highest-priority requester. When the contention occurs for multiple successive cycles, a contention counter assures that the lower-priority requester gets access to the resource every 1 out of n arbitration cycles, where n is programmable. A priority level of -1 represents a transfer whose priority has been increased due to expiration of the contention counter or a transfer that is fixed as the highest-priority transfer to a given resource.

2.2.2.4 Internal DMA (IDMA) Controller

The IDMA controller performs fast block transfers between any two memory locations local to the C674x megamodule. Local memory locations are defined as those in Level 1 program (L1P), Level 1 data (L1D), and Level 2 (L2) memories, or in the external peripheral configuration (CFG) memory. The IDMA cannot transfer data to or from the internal MMR space. The IDMA is fully described in the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

2.3 Memory Map

Refer to your device-specific data manual for memory-map information.

2.3.1 DSP Internal Memory

See [Section 4.2](#) for a description of the DSP internal memory.

2.3.2 External Memory

See [Chapter 3](#) and [Chapter 4](#) for a description of the additional system memory and peripherals that the DSP has access to.

2.4 Advanced Event Triggering (AET)

The C674x megamodule supports advanced event triggering (AET). This capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- **Hardware Program Breakpoints:** specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- **Data Watchpoints:** specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- **Counters:** count the occurrence of an event or cycles for performance monitoring.
- **State Sequencing:** allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

System Interconnect

Topic	Page
3.1 Introduction	26
3.2 System Interconnect Block Diagram	27

3.1 Introduction

The DSP, the EDMA3 transfer controllers, and the device peripherals are interconnected through a switch fabric architecture (see [Section 3.2](#)). The switch fabric is composed of multiple switched central resources (SCRs) and multiple bridges. The SCRs establish low-latency connectivity between master peripherals and slave peripherals. Additionally, the SCRs provide priority-based arbitration and facilitate concurrent data movement between master and slave peripherals. Through SCR, the DSP can send data to the EMIF without affecting a data transfer between a device peripheral and internal shared memory. Bridges are mainly used to perform bus-width conversion as well as bus operating frequency conversion.

The DSP, the EDMA3 transfer controllers, and the various device peripherals can be classified into two categories: master peripherals and slave peripherals. Master peripherals are typically capable of initiating read and write transfers in the system and do not rely on the EDMA3 or on a CPU to perform transfers to and from them. The system master peripherals include the DSP, the EDMA3 transfer controllers, EMAC, HPI, LCDC, and USB. Not all master peripherals may connect to all slave peripherals. The supported connections are designated by an X in [Table 3-1](#).

Table 3-1. TMS320C6745/C6747 DSP System Interconnect Matrix

Masters		Slaves					
Master	Default Priority	DSP SDMA	EMIFA	EMIFB	128K RAM	EDMA3TC Group ⁽¹⁾	Peripheral Group ⁽²⁾
EDMA3CC0	0					X	
EDMA3TC0	0	X	X	X	X	X	X
EDMA3TC1	0	X	X	X	X	X	X
PRU0	0	X	X	X	X	X	X
PRU1	0	X	X	X	X	X	X
DSP CFG	2					X	X
DSP MDMA	2		X	X	X		
EMAC	4	X	X	X	X		
USB2.0	4	X	X	X	X		
USB1.1 ⁽³⁾	4	X	X	X	X		
LCDC ⁽³⁾	5			X			
HPI ⁽³⁾	6	X		X	X		X ⁽⁴⁾

⁽¹⁾ EDMA3TC group: EDMA3TC0, EDMA3TC1

⁽²⁾ Peripheral group: SYSCFG, EMAC, eCAP0, eCAP1, eCAP2, eHRPWM0, eHRPWM1, eHRPWM2, GPIO, I2C0, I2C1, LCDC, McASP0, McASP1, McASP2, MDIO, MMC/SD, PLLC, PRU RAM0, PRU RAM1, PRU Config, PSC0, PSC1, RTC, SPI0, SPI1, TIMER64P0, TIMER64P1, EDMA3CC0, UART0, UART1, UART2, HPI, USB0 (USB2.0), USB1 (USB1.1). The LCDC and McASP2 are not supported on the C6745 DSP.

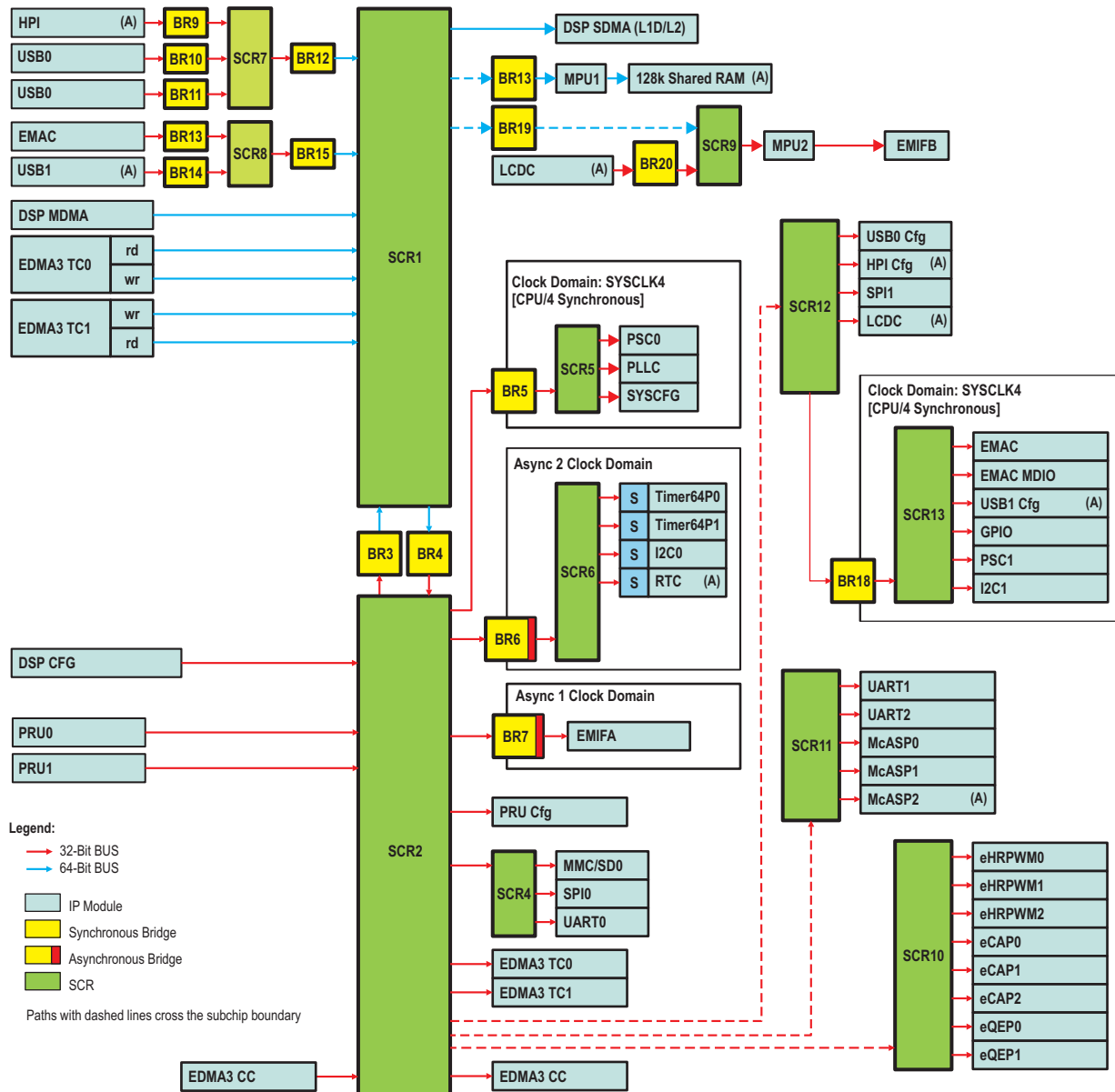
⁽³⁾ This peripheral is not supported on the C6745 DSP.

⁽⁴⁾ The HPI does not have access to all registers in the SYSCFG module because it operates with the User Privilege Level.

3.2 System Interconnect Block Diagram

Figure 3-1 shows a system interconnect block diagram.

Figure 3-1. System Interconnect Block Diagram



A This peripheral is not supported on the C6745 DSP.

System Memory

Topic	Page
4.1 Introduction	30
4.2 DSP Memories	30
4.3 Shared RAM	30
4.4 External Memories	30
4.5 Internal Peripherals	30
4.6 Peripherals	31

4.1 Introduction

This device has multiple on-chip/off-chip memories and several external device interfaces associated with the DSP and various subsystems. To help simplify software development, a unified memory-map is used wherever possible to maintain a consistent view of device resources across all masters (CPU and master peripherals).

For details on the memory addresses, actual memory supported and accessibility by various bus masters, see the detailed memory-map information in the device-specific data manual.

4.2 DSP Memories

The DSP internal memories are accessible by the DSP and other master peripherals (as dictated by the connectivity matrix) via the system interconnect through the DSP SDMA port. The accesses by the DSP to its internal memory are internal to the DSP subsystem and do not go out on the system interconnect.

The DSP internal memory consists of L1P, L1D, and L2. The DSP internal memory configuration is:

- L1P memory includes 32 KB of RAM. The DSP program memory controller (PMC) allows you to configure part or all of the L1P RAM as normal program RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, 16 KB, or 32 KB of the 32 KB of RAM. The default configuration is 32 KB cache.
- L1D memory includes 32 KB of RAM. The DSP data memory controller (DMC) allows you to configure part of the L1D RAM as normal data RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, 16 KB, or 32 KB of the 32 KB of RAM. The default configuration is 32 KB cache.
- L2 memory includes 256 KB of RAM. The DSP unified memory controller (UMC) allows you to configure part or all of the L2 RAM as normal RAM or as cache. You can configure cache sizes of 0 KB, 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, or 256 KB of the 256 KB of RAM. The default configuration is 256 KB normal RAM.
- L2 memory also includes 1024 KB of ROM.

4.3 Shared RAM

This device also offers an on-chip 128-KB shared RAM, apart from the DSP level 1 and level 2 internal memories. This shared RAM is accessible by the DSP and also is accessible by several master peripherals. The 128-KB shared RAM is not supported on the C6745 DSP.

4.4 External Memories

This device has two external memory interfaces that provide multiple external memory options accessible by the CPU and master peripherals:

- EMIFA:
 - 8/16-bit wide (package dependent) asynchronous EMIF module that supports asynchronous devices such as ASRAM, NAND Flash, and NOR Flash (up to 4 devices)
 - 8/16-bit wide (package dependent) NAND Flash with 4-bit ECC (up to 4 devices)
 - 16-bit SDRAM with 128-MB address space (package dependent)
- EMIFB: 32/16-bit SDRAM (package dependent) with up to 256-MB SDRAM address space

4.5 Internal Peripherals

The following peripherals are internal to the DSP subsystem and are only accessible to the DSP:

- DSP interrupt controller (INTC)
- DSP power down controller (PDC)
- Bandwidth manager (BWM)
- Internal DMA (IDMA)

For more information on these internal peripherals, see the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

4.6 Peripherals

The DSP has access to all peripherals on the device. See the device-specific data manual for the complete list of peripherals supported on your device.

Memory Protection Unit (MPU)

Topic	Page
5.1 Introduction	34
5.2 Architecture	35
5.3 MPU Registers	40

5.1 Introduction

This device supports two memory protection units (MPU1 and MPU2). MPU1 supports the 128KB shared RAM and MPU2 supports the EMIFB. The MPU1 is not supported on the C6745 DSP.

5.1.1 Purpose of the MPU

The memory protection unit (MPU) is provided to manage access to memory. The MPU allows you to define multiple ranges and limit access to system masters based on their privilege ID. The MPU can record a detected fault, or invalid access, and notify the system through an interrupt.

5.1.2 Features

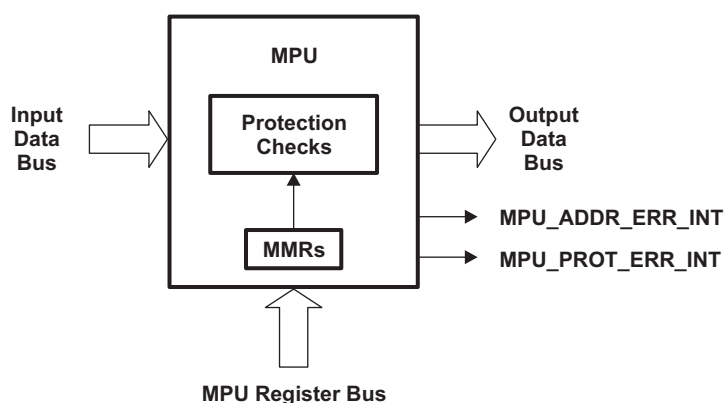
The MPU supports the following features:

- Supports multiple programmable address ranges
- Supports 0 or 1 fixed range
- Supports read, write, and execute access privileges
- Supports privilege ID associations with ranges
- Generates an interrupt when there is a protection violation, and saves violating transfer parameters
- Supports L1/L2 cache accesses
- Supports protection of its own registers

5.1.3 Block Diagram

Figure 5-1 shows a block diagram of the MPU. An access to a protected memory must pass through the MPU. During an access, the MPU checks the memory address on the input data bus against fixed and programmable ranges. If allowed, the transfer is passed unmodified to the output data bus. If the transfer fails the protection check then the MPU does not pass the transfer to the output bus but rather services the transfer internally back to the input bus (to prevent a hang) returning the fault status to the requestor as well as generating an interrupt about the fault. The MPU generates two interrupts: an address error interrupt (MPU_ADDR_ERR_INT) and a protection interrupt (MPU_PROT_ERR_INT).

Figure 5-1. MPU Block Diagram



5.1.4 MPU Default Configuration

Two MPUs are supported on the device, one for the 128KB shared RAM and one for the EMIFB. [Table 5-1](#) shows the memory regions protected by each MPU. [Table 5-2](#) shows the configuration of each MPU.

Table 5-1. MPU Memory Regions

Unit	Memory Protection	Memory Region	
		Start Address	End Address
MPU1	128KB Shared RAM ⁽¹⁾	8000 0000h	8001 FFFFh
MPU2	EMIFB	C000 0000h	DFFF FFFFh

⁽¹⁾ The 128KB shared RAM is not supported on the C6745 DSP; therefore, the MPU1 is not supported.

Table 5-2. MPU Default Configuration

Setting	MPU1 ⁽¹⁾	MPU2
Default permission	Assume allowed	Assume allowed
Number of allowed IDs supported	12	12
Number of fixed ranges supported	1	0
Number of programmable ranges supported	6	12
Compare width	1 KB granularity	64 KB granularity

⁽¹⁾ MPU1 is not supported on the C6745 DSP.

5.2 Architecture

5.2.1 Privilege Levels

The privilege level of a memory access determines what level of permissions the originator of the memory access might have. Two privilege levels are supported: supervisor and user.

Supervisor level is generally granted access to peripheral registers and the memory protection configuration. User level is generally confined to the memory spaces that the OS specifically designates for its use.

DSP CPU instruction and data accesses have a privilege level associated with them. The privilege level is inherited from the code running on the CPU. See the *TMS320C674x DSP CPU and Instruction Set Reference Guide* ([SPRUFE8](#)) for more details on privilege levels of the DSP CPU.

Although master peripherals like the EMAC do not execute code, they still have a privilege level associated with them. Unlike the DSP CPU, the privilege level of this peripheral is fixed.

[Table 5-3](#) shows the privilege ID of the CPU and every mastering peripheral. [Table 5-3](#) also shows the privilege level (supervisor vs. user) and access type (instruction read vs. data/DMA read or write) of each master on the device. In some cases, a particular setting depends on software being executed at the time of the access or the configuration of the master peripheral.

Table 5-3. Device Master Settings

Master	Privilege ID	Privilege Level	Access Type
EDMA3CC	Inherited	Inherited	DMA
EDMA3TC0 and TC1	Inherited	Inherited	DMA
DSP	1	Software dependant	Software dependant
PRU0/PRU1	2	Supervisor	DMA
HPI ⁽¹⁾	3	User	DMA
EMAC	4	Supervisor	Data/DMA
USB1.1 ⁽¹⁾	5	Supervisor	DMA
USB2.0	6	Supervisor	DMA
LCD Controller ⁽¹⁾	7	Supervisor	DMA

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

5.2.2 Memory Protection Ranges

NOTE: In some cases the amount of physical memory in actual use may be less than the maximum amount of memory supported by the device. For example, the device may support a total of 512 Mbytes of SDRAM memory, but your design may only populate 128 Mbytes. In such cases, the “unpopulated” memory range must be protected in order to prevent unintended/disallowed “aliased” access to protected memory. One of the programmable address ranges could be used to detect accesses to this “unpopulated” memory.

The MPU divides its assigned memory into address ranges. Each MPU can support one fixed address range and multiple programmable address ranges. The fixed address range is configured to an exact address. The programmable address range allows software to program the start and end addresses.

Each address range has the following set of registers:

- Range start and end address registers (MPSAR and MPEAR): Specifies the starting and ending address of the address range.
- Memory protection page attribute register (MPPA): Use to program the permission settings of the address range.

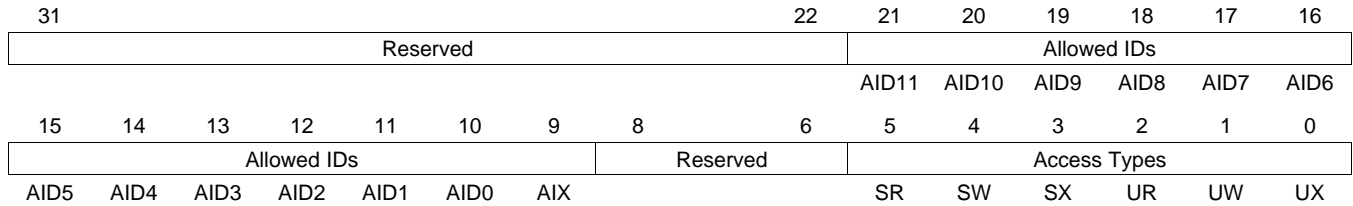
It is allowed to configure ranges such that they overlap each other. In this case, all the overlapped ranges must allow the access, otherwise the access is not allowed. The final permissions given to the access are the lowest of each type of permission from any hit range.

Addresses not covered by a range are either allowed or disallowed based on the configuration of the MPU. The MPU can be configured for “assumed allowed” or “assumed disallowed” mode as dictated by the ASSUME_ALLOWED bit in the configuration register (CONFIG).

5.2.3 Permission Structures

The MPU defines a per-range permission structure with three permission fields in a 32-bit permission entry. [Figure 5-2](#) shows the structure of a permission entry.

Figure 5-2. Permission Fields



5.2.3.1 Requestor-ID Based Access Controls

Each master on the device has an N-bit code associated with it that identifies it for privilege purposes. This privilege ID accompanies all memory accesses made on behalf of that master. That is, when a master triggers a memory access command, the privilege ID will be carried alongside the command.

Each memory protection range has an allowed ID (AID) field associated with it that indicates which requestors may access the given address range. The MPU maps the privilege IDs of all the possible requestors to bits in the allowed IDs field in the memory protection page attribute registers (MPPA).

- AID0 through AID11 are used to specify the allowed privilege IDs.
- An additional allowed ID bit, AIDX, captures access made by all privilege IDs not covered by AID0 through AID11.

When set to 1, the AID bit grants access to the corresponding ID. When cleared to 0, the AID bit denies access to the corresponding requestor.

5.2.3.2 Request-Type Based Permissions

The memory protection model defines three fundamental functional access types: read, write, and execute. Read and write refer to data accesses -- accesses originating via the load/store units on the CPU or via a master peripheral. Execute refers to accesses associated with an instruction fetch.

The memory protection model allows controlling read, write, and execute permissions independently for both user and supervisor mode. This results in six permission bits, listed in [Table 5-4](#). For each bit, a 1 permits the access type and a 0 denies access. For example, UX = 1 means that User Mode may execute from the given page. The memory protection unit allows you to specify all six of these bits separately; 64 different encodings are permitted altogether, although programs might not use all of them.

Table 5-4. Request Type Access Controls

Bit	Field	Description
5	SR	Supervisor may read
4	SW	Supervisor may write
3	SX	Supervisor may execute
2	UR	User may read
1	UW	User may write
0	UX	User may execute

5.2.4 Protection Check

During a memory access, the MPU checks if the address range of the input transfer overlaps one of the address ranges. When the input transfer address is within a range the transfer parameters are checked against the address range permissions.

The MPU first checks the transfer's privilege ID against the AID settings. If the AID bit is 0, then the range will not be checked; if the AID bit is 1, then the transfer parameters are checked against the memory protection page attribute register (MPPA) values to detect an allowed access.

For non-debug accesses, the read, write, and execute permissions are also checked. There is a set of permissions for supervisor mode and a set for user mode. For supervisor mode accesses, the SR, SW, and SX bits are checked. For user mode accesses, the UR, UW, and UX bits are checked.

If the transfer address range does not match any address range then the transfer is either allowed or disallowed based on the configuration of the MPU. The MPU can be configured for "assumed allowed" or "assumed disallowed" mode as dictated by the ASSUME_ALLOWED bit in the configuration register (CONFIG).

In the case that a transfer spans multiple address ranges, all the overlapped ranges must allow the access, otherwise the access is not allowed. The final permissions given to the access are the lowest of each type of permission from any hit range. Therefore, if a transfer matches 2 ranges, one that is RW and one that is RX, then the final permission is just R.

The MPU has a special mechanism for handling DSP L1/L2 cache controller read accesses, see [Section 5.2.5](#) for more details.

5.2.5 DSP L1/L2 Cache Controller Accesses

A memory read access that originates from the DSP L1/L2 cache is treated differently to allow memory protection to be enforced by the DSP level. This is because a subsequent memory access that hits in the cache does not pass through the MPU. Instead the memory access is serviced directly by the L1/L2 memory controllers.

During a cache memory read, the permission settings stored in the memory protection page attribute registers (MPPA) are passed to the L1/L2 memory controllers along with the read data. The permissions settings returned by the MPU are taken from MPPA that covers the address range of the original request—only the SR, SW, SX, UR, UW, and UX bits are passed. If the request address is covered by multiple address ranges, then the returned value is the logical-AND of all MPPA permissions. If the transfer address range is not covered by an address range then the transfer is either allowed or disallowed based on the configuration of the MPU.

5.2.6 MPU Register Protection

Access to the range start and end address registers (MPSAR and MPEAR) and memory protection page attribute registers (MPPA) is also protected. All non-debug writes must be by a supervisor entity. A protection fault can occur from a register write with invalid permissions and this triggers an interrupt just like a memory access.

Faults are not recorded (nor interrupts generated) for debug accesses.

5.2.7 Invalid Accesses and Exceptions

When a transfer fails the protection check, the MPU does not pass the transfer to the output bus. The MPU instead services the transfer locally to prevent a hang and returns a protection error to the requestor. The behavior of the MPU depends on whether the access was a read or a write:

- For a read: The MPU returns 0s, a permission value is 0 (no access allowed), a protection error status.
- For a write: The MPU receives all the write data and returns a protection error status.

The MPU captures system faults due to addressing or protection violations in its registers. The MPU can store the fault information for only one fault, so the first detected fault is recorded into the fault registers and an interrupt is generated. Software must use the fault clear register (FLTCLR) to clear the fault status so that another fault can be recorded. The MPU will not record another fault nor generate another interrupt until the existing fault has been cleared. Also, additional faults will be ignored. Faults are not recorded (no interrupts generated) for debug accesses.

5.2.8 Reset Considerations

After reset, the memory protection page attribute registers (MPPA) default to 0. This disables all protection features.

5.2.9 Interrupt Support

5.2.9.1 Interrupt Events and Requests

The MPU generates two interrupts: an address error interrupt (MPU_ADDR_ERR_INT) and a protection interrupt (MPU_PROT_ERR_INT). The MPU_ADDR_ERR_INT is generated when there is an addressing violation due to an access to a non-existent location in the MPU register space. The MPU_PROT_ERR_INT interrupt is generated when there is a protection violation of either in the defined ranges or to the MPU registers.

The transfer parameters that caused the violation are saved in the MPU registers.

5.2.9.2 Interrupt Multiplexing

The interrupts from both MPUs are combined with the boot configuration module into a single interrupt called MPU_BOOTCFG_ERR. The combined interrupt is routed to the DSP interrupt controller. [Table 5-5](#) shows the interrupt sources that are combined to make MPU_BOOTCFG_ERR.

Table 5-5. MPU_BOOTCFG_ERR Interrupt Sources

Interrupt	Source
MPU1_ADDR_ERR_INT ⁽¹⁾	MPU1 address error interrupt
MPU1_PROT_ERR_INT ⁽¹⁾	MPU1 protection interrupt
MPU2_ADDR_ERR_INT	MPU2 address error interrupt
MPU2_PROT_ERR_INT	MPU2 protection interrupt
BOOTCFG_ADDR_ERR	Boot configuration address error
BOOTCFG_PROT_ERR	Boot configuration protection error

⁽¹⁾ MPU1 is not supported on the C6745 DSP.

5.2.10 Emulation Considerations

Memory and MPU registers are not protected against emulation accesses.

5.3 MPU Registers

There are two MPUs on the device. Each MPU contains a set of memory-mapped registers.

[Table 5-6](#) lists the memory-mapped registers for the MPU1. [Table 5-7](#) lists the memory-mapped registers for the MPU2.

Table 5-6. Memory Protection Unit 1 (MPU1) Registers

Address	Acronym	Register Description	Section
01E1 4000h	REVID	Revision identification register	Section 5.3.1
01E1 4004h	CONFIG	Configuration register	Section 5.3.2
01E1 4010h	IRAWSTAT	Interrupt raw status/set register	Section 5.3.3
01E1 4014h	IENSTAT	Interrupt enable status/clear register	Section 5.3.4
01E1 4018h	IENSET	Interrupt enable set register	Section 5.3.5
01E1 401Ch	IENCLR	Interrupt enable clear register	Section 5.3.6
01E1 4200h	PROG1_MPSAR	Programmable range 1 start address register	Section 5.3.10.1
01E1 4204h	PROG1_MPEAR	Programmable range 1 end address register	Section 5.3.11.1
01E1 4208h	PROG1_MPPA	Programmable range 1 memory protection page attributes register	Section 5.3.12
01E1 4210h	PROG2_MPSAR	Programmable range 2 start address register	Section 5.3.10.1
01E1 4214h	PROG2_MPEAR	Programmable range 2 end address register	Section 5.3.11.1
01E1 4218h	PROG2_MPPA	Programmable range 2 memory protection page attributes register	Section 5.3.12
01E1 4220h	PROG3_MPSAR	Programmable range 3 start address register	Section 5.3.10.1
01E1 4224h	PROG3_MPEAR	Programmable range 3 end address register	Section 5.3.11.1
01E1 4228h	PROG3_MPPA	Programmable range 3 memory protection page attributes register	Section 5.3.12
01E1 4230h	PROG4_MPSAR	Programmable range 4 start address register	Section 5.3.10.1
01E1 4234h	PROG4_MPEAR	Programmable range 4 end address register	Section 5.3.11.1
01E1 4238h	PROG4_MPPA	Programmable range 4 memory protection page attributes register	Section 5.3.12
01E1 4240h	PROG5_MPSAR	Programmable range 5 start address register	Section 5.3.10.1
01E1 4244h	PROG5_MPEAR	Programmable range 5 end address register	Section 5.3.11.1
01E1 4248h	PROG5_MPPA	Programmable range 5 memory protection page attributes register	Section 5.3.12
01E1 4250h	PROG6_MPSAR	Programmable range 6 start address register	Section 5.3.10.1
01E1 4254h	PROG6_MPEAR	Programmable range 6 end address register	Section 5.3.11.1
01E1 4258h	PROG6_MPPA	Programmable range 6 memory protection page attributes register	Section 5.3.12
01E1 4300h	FLTADDRR	Fault address register	Section 5.3.13
01E1 4304h	FLTSTAT	Fault status register	Section 5.3.14
01E1 4308h	FLTCLR	Fault clear register	Section 5.3.15

Table 5-7. Memory Protection Unit 2 (MPU2) Registers

Address	Acronym	Register Description	Section
01E1 5000h	REVID	Revision identification register	Section 5.3.1
01E1 5004h	CONFIG	Configuration register	Section 5.3.2
01E1 5010h	IRAWSTAT	Interrupt raw status/set register	Section 5.3.3
01E1 5014h	IENSTAT	Interrupt enable status/clear register	Section 5.3.4
01E1 5018h	IENSET	Interrupt enable set register	Section 5.3.5
01E1 501Ch	IENCLR	Interrupt enable clear register	Section 5.3.6
01E1 5100h	FXD_MPSAR	Fixed range start address register	Section 5.3.7
01E1 5104h	FXD_MPEAR	Fixed range end address register	Section 5.3.8
01E1 5108h	FXD_MPPA	Fixed range memory protection page attributes register	Section 5.3.9
01E1 5200h	PROG1_MPSAR	Programmable range 1 start address register	Section 5.3.10.2

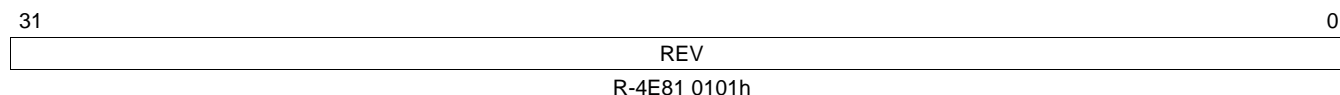
Table 5-7. Memory Protection Unit 2 (MPU2) Registers (continued)

Address	Acronym	Register Description	Section
01E1 5204h	PROG1_MPEAR	Programmable range 1 end address register	Section 5.3.11.2
01E1 5208h	PROG1_MPPA	Programmable range 1 memory protection page attributes register	Section 5.3.12
01E1 5210h	PROG2_MPSAR	Programmable range 2 start address register	Section 5.3.10.2
01E1 5214h	PROG2_MPEAR	Programmable range 2 end address register	Section 5.3.11.2
01E1 5218h	PROG2_MPPA	Programmable range 2 memory protection page attributes register	Section 5.3.12
01E1 5220h	PROG3_MPSAR	Programmable range 3 start address register	Section 5.3.10.2
01E1 5224h	PROG3_MPEAR	Programmable range 3 end address register	Section 5.3.11.2
01E1 5228h	PROG3_MPPA	Programmable range 3 memory protection page attributes register	Section 5.3.12
01E1 5230h	PROG4_MPSAR	Programmable range 4 start address register	Section 5.3.10.2
01E1 5234h	PROG4_MPEAR	Programmable range 4 end address register	Section 5.3.11.2
01E1 5238h	PROG4_MPPA	Programmable range 4 memory protection page attributes register	Section 5.3.12
01E1 5240h	PROG5_MPSAR	Programmable range 5 start address register	Section 5.3.10.2
01E1 5244h	PROG5_MPEAR	Programmable range 5 end address register	Section 5.3.11.2
01E1 5248h	PROG5_MPPA	Programmable range 5 memory protection page attributes register	Section 5.3.12
01E1 5250h	PROG6_MPSAR	Programmable range 6 start address register	Section 5.3.10.2
01E1 5254h	PROG6_MPEAR	Programmable range 6 end address register	Section 5.3.11.2
01E1 5258h	PROG6_MPPA	Programmable range 6 memory protection page attributes register	Section 5.3.12
01E1 5260h	PROG7_MPSAR	Programmable range 7 start address register	Section 5.3.10.2
01E1 5274h	PROG7_MPEAR	Programmable range 7 end address register	Section 5.3.11.2
01E1 5268h	PROG7_MPPA	Programmable range 7 memory protection page attributes register	Section 5.3.12
01E1 5270h	PROG8_MPSAR	Programmable range 8 start address register	Section 5.3.10.2
01E1 5274h	PROG8_MPEAR	Programmable range 8 end address register	Section 5.3.11.2
01E1 5278h	PROG8_MPPA	Programmable range 8 memory protection page attributes register	Section 5.3.12
01E1 5280h	PROG9_MPSAR	Programmable range 9 start address register	Section 5.3.10.2
01E1 5284h	PROG9_MPEAR	Programmable range 9 end address register	Section 5.3.11.2
01E1 5288h	PROG9_MPPA	Programmable range 9 memory protection page attributes register	Section 5.3.12
01E1 5290h	PROG10_MPSAR	Programmable range 10 start address register	Section 5.3.10.2
01E1 5294h	PROG10_MPEAR	Programmable range 10 end address register	Section 5.3.11.2
01E1 5298h	PROG10_MPPA	Programmable range 10 memory protection page attributes register	Section 5.3.12
01E1 52A0h	PROG11_MPSAR	Programmable range 11 start address register	Section 5.3.10.2
01E1 52A4h	PROG11_MPEAR	Programmable range 11 end address register	Section 5.3.11.2
01E1 52A8h	PROG11_MPPA	Programmable range 11 memory protection page attributes register	Section 5.3.12
01E1 52B0h	PROG12_MPSAR	Programmable range 12 start address register	Section 5.3.10.2
01E1 52B4h	PROG12_MPEAR	Programmable range 12 end address register	Section 5.3.11.2
01E1 52B8h	PROG12_MPPA	Programmable range 12 memory protection page attributes register	Section 5.3.12
01E1 5300h	FLTADDRR	Fault address register	Section 5.3.13
01E1 5304h	FLTSTAT	Fault status register	Section 5.3.14
01E1 5308h	FLTCLR	Fault clear register	Section 5.3.15

5.3.1 Revision Identification Register (REVID)

The revision ID register (REVID) contains the MPU revision. The REVID is shown in [Figure 5-3](#) and described in [Table 5-8](#).

Figure 5-3. Revision ID Register (REVID)



LEGEND: R = Read only; -n = value after reset

Table 5-8. Revision ID Register (REVID) Field Descriptions

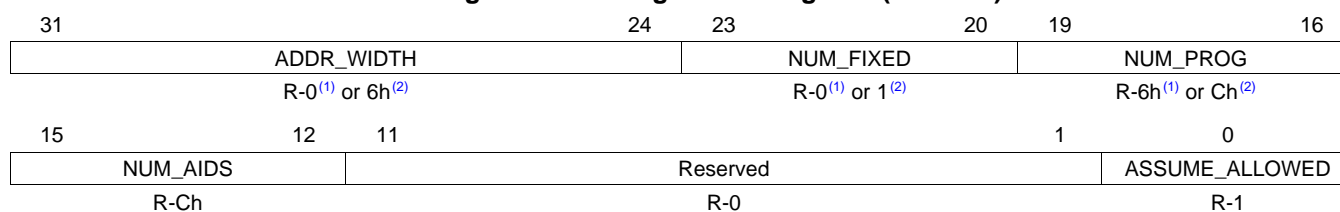
Bit	Field	Value	Description
31-0	REV	4E81 0101h	Revision ID of the MPU.

5.3.2 Configuration Register (CONFIG)

The configuration register (CONFIG) contains the configuration value of the MPU. The CONFIG is shown in [Figure 5-4](#) and described in [Table 5-9](#).

NOTE: Although the NUM_AIDS bit defaults to 12 (Ch), not all AIDs may be supported on your device. Unsupported AIDs should be cleared to 0 in the memory page protection attributes registers (MPPA). See [Table 5-3](#) for a list of AIDs supported on your device.

Figure 5-4. Configuration Register (CONFIG)



LEGEND: R = Read only; -n = value after reset

⁽¹⁾ For MPU1.

⁽²⁾ For MPU2.

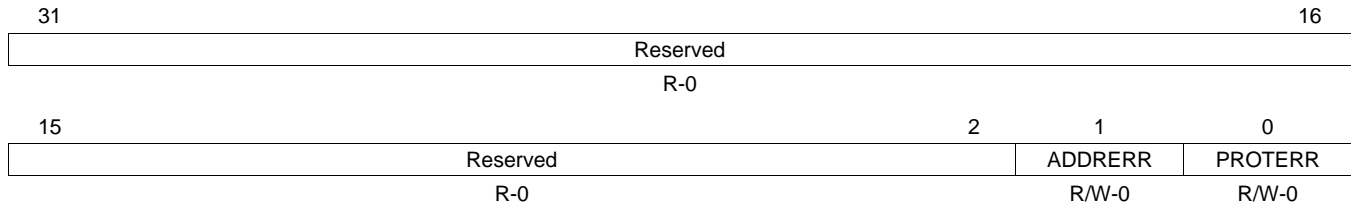
Table 5-9. Configuration Register (CONFIG) Field Descriptions

Bit	Field	Value	Description
31-24	ADDR_WIDTH	0-FFh	Address alignment (2 ⁿ KByte alignment) for range checking.
23-20	NUM_FIXED	0-Fh	Number of fixed address ranges.
19-16	NUM_PROG	0-Fh	Number of programmable address ranges.
15-12	NUM_AIDS	0-Fh	Number of supported AIDs.
11-1	Reserved	0	Reserved
0	ASSUME_ALLOWED	0 1	Assume allowed. When an address is not covered by any MPU protection range, this bit determines whether the transfer is assumed to be allowed or not allowed. Assume is disallowed. Assume is allowed.

5.3.3 Interrupt Raw Status/Set Register (IRAWSTAT)

Reading the interrupt raw status/set register (IRAWSTAT) returns the status of all interrupts. Software can write to IRAWSTAT to manually set an interrupt; however, an interrupt is generated only if the interrupt is enabled in the interrupt enable set register (IENSET). Writes of 0 have no effect. The IRAWSTAT is shown in Figure 5-5 and described in Table 5-10.

Figure 5-5. Interrupt Raw Status/Set Register (IRAWSTAT)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

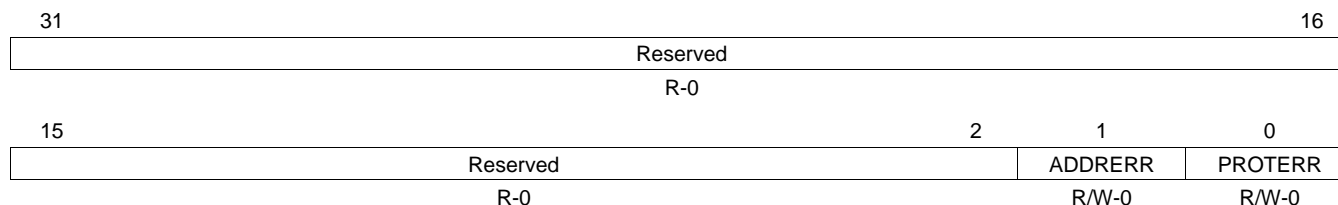
Table 5-10. Interrupt Raw Status/Set Register (IRAWSTAT) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	ADDRERR	0	Address violation error. Reading this bit reflects the status of the interrupt. Writing 1 sets the status; writing 0 has no effect.
		1	Interrupt is not set.
		1	Interrupt is set.
0	PROTERR	0	Protection violation error. Reading this bit reflects the status of the interrupt. Writing 1 sets the status; writing 0 has no effect.
		0	Interrupt is not set.
		1	Interrupt is set.

5.3.4 Interrupt Enable Status/Clear Register (IENSTAT)

Reading the interrupt enable status/clear register (IENSTAT) returns the status of only those interrupts that are enabled in the interrupt enable set register (IENSET). Software can write to IENSTAT to clear an interrupt; the interrupt is cleared from both IENSTAT and the interrupt raw status/set register (IRAWSTAT). Writes of 0 have no effect. The IENSTAT is shown in [Figure 5-6](#) and described in [Table 5-11](#).

Figure 5-6. Interrupt Enable Status/Clear Register (IENSTAT)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

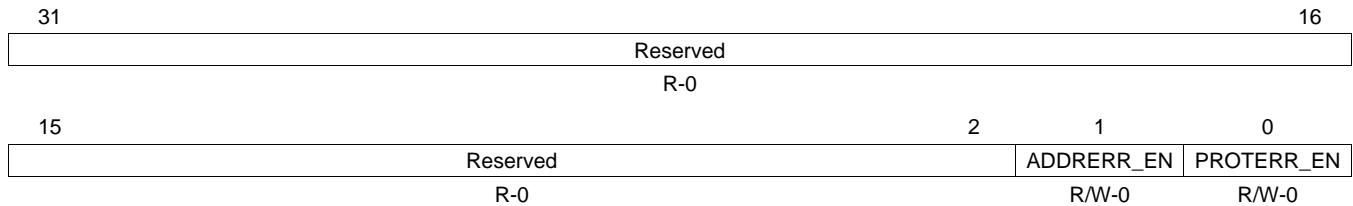
Table 5-11. Interrupt Enable Status/Clear Register (IENSTAT) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	ADDRERR	0 1	Address violation error. If the interrupt is enabled, reading this bit reflects the status of the interrupt. If the interrupt is disabled, reading this bit returns 0. Writing 1 sets the status; writing 0 has no effect. Interrupt is not set. Interrupt is set.
0	PROTERR	0 1	Protection violation error. If the interrupt is enabled, reading this bit reflects the status of the interrupt. If the interrupt is disabled, reading this bit returns 0. Writing 1 sets the status; writing 0 has no effect. Interrupt is not set. Interrupt is set.

5.3.5 Interrupt Enable Set Register (IENSET)

Reading the interrupt enable set register (IENSET) returns the interrupts that are enabled. Software can write to IENSET to enable an interrupt. Writes of 0 have no effect. The IENSET is shown in [Figure 5-7](#) and described in [Table 5-12](#).

Figure 5-7. Interrupt Enable Set Register (IENSET)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

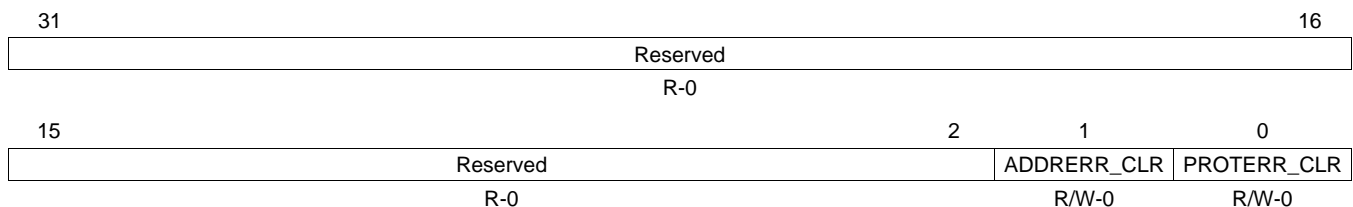
Table 5-12. Interrupt Enable Set Register (IENSET) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	ADDRERR_EN	0	Address violation error enable. Writing 0 has no effect.
		1	Interrupt is enabled.
0	PROTERR_EN	0	Protection violation error enable. Writing 0 has no effect.
		1	Interrupt is enabled.

5.3.6 Interrupt Enable Clear Register (IENCLR)

Reading the interrupt enable clear register (IENCLR) returns the interrupts that are enabled. Software can write to IENCLR to clear/disable an interrupt. Writes of 0 have no effect. The IENCLR is shown in [Figure 5-8](#) and described in [Table 5-13](#).

Figure 5-8. Interrupt Enable Clear Register (IENCLR)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

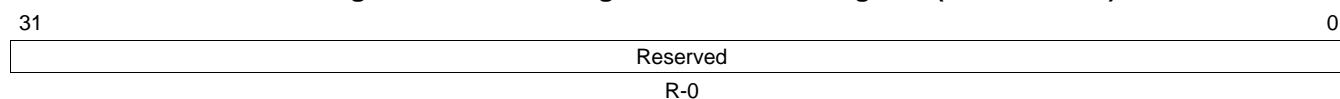
Table 5-13. Interrupt Enable Clear Register (IENCLR) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	ADDRERR_CLR	0	Address violation error disable. Writing 0 has no effect.
		1	Interrupt is cleared/disabled.
0	PROTERR_CLR	0	Protection violation error disable. Writing 0 has no effect.
		1	Interrupt is cleared/disabled.

5.3.7 Fixed Range Start Address Register (FXD_MPSAR)

The fixed range start address register (FXD_MPSAR) holds the start address for the fixed range. The fixed address range manages access to the EMIFB control registers (B000 0000h–B000 7FFFh). However, these addresses are *not* indicated in FXD_MPSAR and the fixed range end address register (FXD_MPEAR), which instead read as 0. The FXD_MPSAR is shown in [Figure 5-9](#).

Figure 5-9. Fixed Range Start Address Register (FXD_MPSAR)

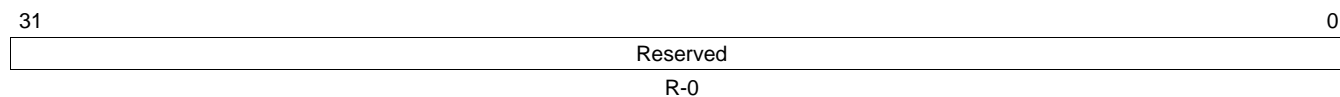


LEGEND: R = Read only; -n = value after reset

5.3.8 Fixed Range End Address Register (FXD_MPEAR)

The fixed range end address register (FXD_MPEAR) holds the end address for the fixed range. The fixed address range manages access to the EMIFB control registers (B000 0000h–B000 7FFFh). However, these addresses are *not* indicated in FXD_MPEAR and the fixed range start address register (FXD_MPSAR), which instead read as 0. The FXD_MPEAR is shown in [Figure 5-10](#).

Figure 5-10. Fixed Range End Address Register (FXD_MPEAR)



LEGEND: R = Read only; -n = value after reset

5.3.9 Fixed Range Memory Protection Page Attributes Register (FXD_MPPA)

The fixed range memory protection page attributes register (FXD_MPPA) holds the permissions for the fixed region. This register is writeable by a supervisor entity only. The FXD_MPPA is shown in [Figure 5-11](#) and described in [Table 5-14](#).

Figure 5-11. Fixed Range Memory Protection Page Attributes Register (FXD_MPPA)

31				26				25				22				21		20		19		18		17		16					
Reserved								Reserved								AID11	AID10	AID9	AID8	AID7	AID6										
R-0								R-Fh								R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1				
15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
AID5	AID4	AID3	AID2	AID1	AID0	AIDX	Rsvd	Rsvd	Rsvd	SR	SW	SX	UR	UW	UX																
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1		

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 5-14. Fixed Range Memory Protection Page Attributes Register (FXD_MPPA) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-22	Reserved	Fh	Reserved
21-10	AID _n	0 1	Controls access from ID = n. Access is denied. Access is granted.
9	AIDX	0 1	Controls access from ID > 11. Access is denied. Access is granted.
8	Reserved	0	Reserved
7	Reserved	1	Reserved. This bit must be written as 1.
6	Reserved	1	Reserved. This bit must be written as 1.
5	SR	0 1	Supervisor Read permission. Access is denied. Access is allowed.
4	SW	0 1	Supervisor Write permission. Access is denied. Access is allowed.
3	SX	0 1	Supervisor Execute permission. Access is denied. Access is allowed.
2	UR	0 1	User Read permission. Access is denied. Access is allowed.
1	UW	0 1	User Write permission. Access is denied. Access is allowed.
0	UX	0 1	User Execute permission. Access is denied. Access is allowed.

5.3.10 Programmable Range *n* Start Address Registers (PROG_{*n*}_MPSAR)

NOTE: In some cases the amount of physical memory in actual use may be less than the maximum amount of memory supported by the device. For example, the device may support a total of 512 Mbytes of SDRAM memory, but your design may only populate 128 Mbytes. In such cases, the unpopulated memory range must be protected in order to prevent unintended/disallowed aliased access to protected memory, especially memory. One of the programmable address ranges could be used to detect accesses to this unpopulated memory.

The programmable range *n* start address register (PROG_{*n*}_MPSAR) holds the start address for the range *n*. The PROG_{*n*}_MPSAR is writeable by a supervisor entity only.

The start address must be aligned on a page boundary. The size of the page depends on the MPU: the page size for MPU1 is 1 KByte; the page size for MPU2 is 64 KBytes. The size of the page determines the width of the address field in PROG_{*n*}_MPSAR and the programmable range *n* end address register (PROG_{*n*}_MPEAR). For example, to protect a 64-KB page starting at byte address 8001 0000h, write 8001 0000h to PROG_{*n*}_MPSAR and 8001 FFFFh to PROG_{*n*}_MPEAR.

5.3.10.1 MPU1 Programmable Range *n* Start Address Register (PROG₁_MPSAR-PROG₆_MPSAR)

The PROG_{*n*}_MPSAR for MPU1 is shown in [Figure 5-12](#) and described in [Table 5-15](#).

Figure 5-12. MPU1 Programmable Range *n* Start Address Register (PROG_{*n*}_MPSAR)

31	10 9	0
START_ADDR	Reserved	
R/W-20 0000h	R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 5-15. MPU1 Programmable Range *n* Start Address Register (PROG_{*n*}_MPSAR) Field Descriptions

Bit	Field	Value	Description
31-10	START_ADDR	20 0000h– 20 007Fh	Start address for range N.
9-0	Reserved	0	Reserved

5.3.10.2 MPU2 Programmable Range *n* Start Address Register (PROG₁_MPSAR-PROG₁₂_MPSAR)

The PROG_{*n*}_MPSAR for MPU2 is shown in [Figure 5-13](#) and described in [Table 5-16](#).

Figure 5-13. MPU2 Programmable Range *n* Start Address Register (PROG_{*n*}_MPSAR)

31	16 15	0
START_ADDR	Reserved	
R/W-C000h	R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 5-16. MPU2 Programmable Range *n* Start Address Register (PROG_{*n*}_MPSAR) Field Descriptions

Bit	Field	Value	Description
31-16	START_ADDR	C000h–DFFFh	Start address for range N.
15-0	Reserved	0	Reserved

5.3.11 Programmable Range n End Address Registers (PROG $_n$ _MPEAR)

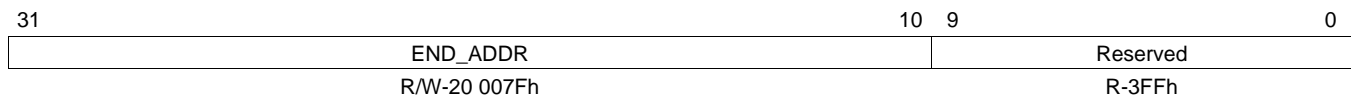
The programmable range n end address register (PROG $_n$ _MPEAR) holds the end address for the range n . This register is writeable by a supervisor entity only.

The end address must be aligned on a page boundary. The size of the page depends on the MPU: the page size for MPU1 is 1 KByte; the page size for MPU2 is 64 KBytes. The size of the page determines the width of the address field in the programmable range n start address register (PROG $_n$ _MPSAR) and PROG $_n$ _MPEAR. For example, to protect a 64-KB page starting at byte address 8001 0000h, write 8001 0000h to PROG $_n$ _MPSAR and 8001 FFFFh to PROG $_n$ _MPEAR.

5.3.11.1 MPU1 Programmable Range n End Address Register (PROG1_MPEAR-PROG6_MPEAR)

The PROG $_n$ _MPEAR for MPU1 is shown in [Figure 5-14](#) and described in [Table 5-17](#).

Figure 5-14. MPU1 Programmable Range n End Address Register (PROG $_n$ _MPEAR)



LEGEND: R/W = Read/Write; R = Read only; - n = value after reset

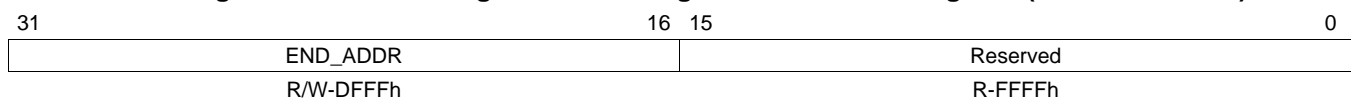
Table 5-17. MPU1 Programmable Range n End Address Register (PROG $_n$ _MPEAR) Field Descriptions

Bit	Field	Value	Description
31-10	END_ADDR	20 0000h– 20 007Fh	End address for range N.
9-0	Reserved	3FFh	Reserved

5.3.11.2 MPU2 Programmable Range n End Address Register (PROG1_MPEAR-PROG12_MPEAR)

The PROG $_n$ _MPEAR for MPU2 is shown in [Figure 5-15](#) and described in [Table 5-18](#).

Figure 5-15. MPU2 Programmable Range n End Address Register (PROG $_n$ _MPEAR)



LEGEND: R/W = Read/Write; R = Read only; - n = value after reset

Table 5-18. MPU2 Programmable Range n End Address Register (PROG $_n$ _MPEAR) Field Descriptions

Bit	Field	Value	Description
31-16	END_ADDR	C000h–DFFFh	Start address for range N.
15-0	Reserved	FFFFh	Reserved

5.3.12 Programmable Range n Memory Protection Page Attributes Register (PROG $_n$ MPPA)

The programmable range n memory protection page attributes register (PROG $_n$ MPPA) holds the permissions for the region n . This register is writeable only by a supervisor entity. The PROG $_n$ MPPA is shown in Figure 5-16 and described in Table 5-19.

Figure 5-16. Programmable Range Memory Protection Page Attributes Register (PROG $_n$ MPPA)

31						26		25		22		21	20	19	18	17	16														
Reserved						Reserved		AID11	AID10	AID9	AID8	AID7	AID6																		
R-0						R-Fh		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1													
15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
AID5	AID4	AID3	AID2	AID1	AID0	AIDX	Rsvd	Rsvd	Rsvd	SR	SW	SX	UR	UW	UX																
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	

LEGEND: R/W = Read/Write; R = Read only; - n = value after reset

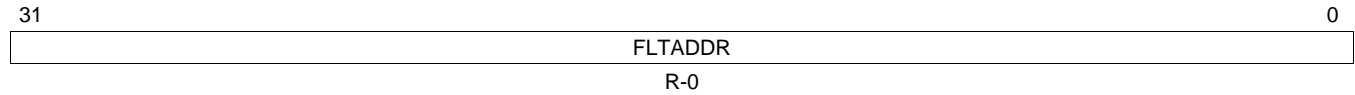
Table 5-19. Programmable Range Memory Protection Page Attributes Register (PROG $_n$ MPPA) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-22	Reserved	Fh	Reserved
21-10	AID n	0 1	Controls access from ID = n . Access is denied. Access is granted.
9	AIDX	0 1	Controls access from ID > 11. Access is denied. Access is granted.
8	Reserved	0	Reserved
7	Reserved	1	Reserved. This bit must be written as 1.
6	Reserved	1	Reserved. This bit must be written as 1.
5	SR	0 1	Supervisor Read permission. Access is denied. Access is allowed.
4	SW	0 1	Supervisor Write permission. Access is denied. Access is allowed.
3	SX	0 1	Supervisor Execute permission. Access is denied. Access is allowed.
2	UR	0 1	User Read permission. Access is denied. Access is allowed.
1	UW	0 1	User Write permission. Access is denied. Access is allowed.
0	UX	0 1	User Execute permission. Access is denied. Access is allowed.

5.3.13 Fault Address Register (FLTADDR)

The fault address register (FLTADDR) holds the address of the first protection fault transfer. The FLTADDR is shown in [Figure 5-17](#) and described in [Table 5-20](#).

Figure 5-17. Fault Address Register (FLTADDR)



LEGEND: R = Read only; -n = value after reset

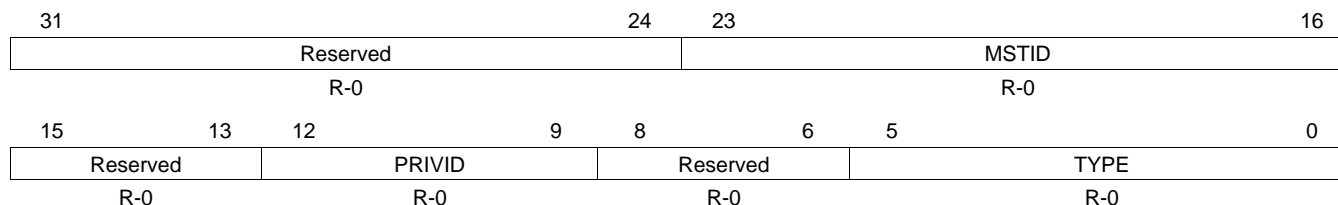
Table 5-20. Fault Address Register (FLTADDR) Field Descriptions

Bit	Field	Value	Description
31-0	FLTADDR	0-FFFF FFFFh	Memory address of fault.

5.3.14 Fault Status Register (FLTSTAT)

The fault status register (FLTSTAT) holds the status and attributes of the first protection fault transfer. The FLTSTAT is shown in [Figure 5-18](#) and described in [Table 5-21](#).

Figure 5-18. Fault Status Register (FLTSTAT)



LEGEND: R = Read only; -n = value after reset

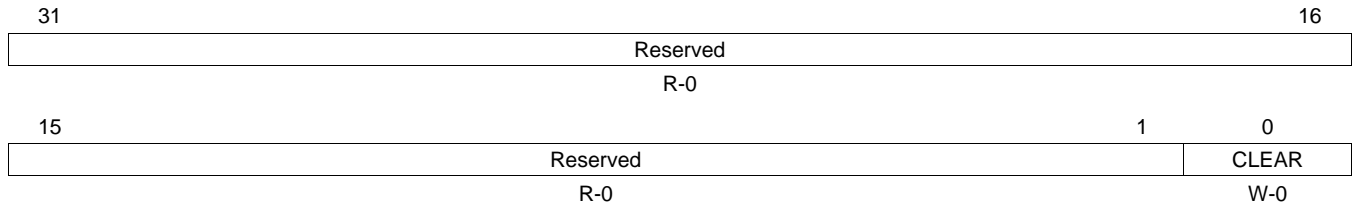
Table 5-21. Fault Status Register (FLTSTAT) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	MSTID	0-FFh	Master ID of fault transfer.
15-13	Reserved	0	Reserved
12-9	PRIVID	0-Fh	Privilege ID of fault transfer.
8-6	Reserved	0	Reserved
5-0	TYPE	0-3Fh	Fault type. The TYPE bit field is cleared when a 1 is written to the CLEAR bit in the fault clear register (FLTCLR).
		0	No fault.
		1h	User execute fault.
		2h	User write fault.
		3h	Reserved
		4h	User read fault.
		5h-7h	Reserved
		8h	Supervisor execute fault.
		9h-Fh	Reserved
		10h	Supervisor write fault.
		11h	Reserved
		12h	Relaxed cache write back fault.
		13h-1Fh	Reserved
		20h	Supervisor read fault.
		21h-3Eh	Reserved
		3Fh	Relaxed cache line fill fault.

5.3.15 Fault Clear Register (FLTCLR)

The fault clear register (FLTCLR) allows software to clear the current fault so that another can be captured in the fault status register (FLTSTAT) as well as produce an interrupt. Only the TYPE bit field in FLTSTAT is cleared when a 1 is written to the CLEAR bit. The FLTCLR is shown in [Figure 5-19](#) and described in [Table 5-22](#).

Figure 5-19. Fault Clear Register (FLTCLR)



LEGEND: R = Read only; W = Write only; -n = value after reset

Table 5-22. Fault Clear Register (FLTCLR) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	CLEAR	0	Command to clear the current fault. Writing 0 has no effect.
		0	No effect.
		1	Clear the current fault.

Device Clocking

Topic	Page
6.1 Overview	56
6.2 Frequency Flexibility	57
6.3 Peripheral Clocking	59

6.1 Overview

This device requires two primary reference clocks:

- One reference clock is required for the phase-locked loop controller (PLL)
- One reference clock is required for the real-time clock (RTC) module. The RTC is not supported on the C6745 DSP.

These reference clocks may be sourced from either a crystal input or by an external oscillator. For detailed specifications on clock frequency and voltage requirements, see the device-specific data manual.

In addition to the reference clocks required for the PLLC and RTC module, some peripherals, such as the USB, may also require an input reference clock to be supplied. All possible input clocks are described in [Table 6-1](#). The CPU and the majority of the device peripherals operate at fixed ratios of the primary system/CPU clock frequency, as listed in [Table 6-2](#). However, there are three system clock domains that do not require a fixed ratio to the CPU clock frequency, these are SYSCLK3, SYSCLK5, and SYSCLK7. [Figure 6-1](#) shows the clocking architecture.

Table 6-1. Device Clock Inputs

Peripheral	Input Clock Signal Name
Oscillator/PLL	OSCIN
RTC ⁽¹⁾	RTC_XI
JTAG	TCK
EMAC	RMII_MHZ_50_CLK
USB2.0 and USB1.1 ⁽¹⁾	USB_REFCLKIN
McASPs ⁽²⁾	ACLKRn, AHCLKRn, ACLKXn, AHCLKXn
I2Cs	I2Cn_SCL
SPIs	SPIn_CLK
Timer0	TM64P0_IN12

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

⁽²⁾ McASP2 is not supported on the C6745 DSP; only McASP0 and McASP1 are supported on the C6745 DSP.

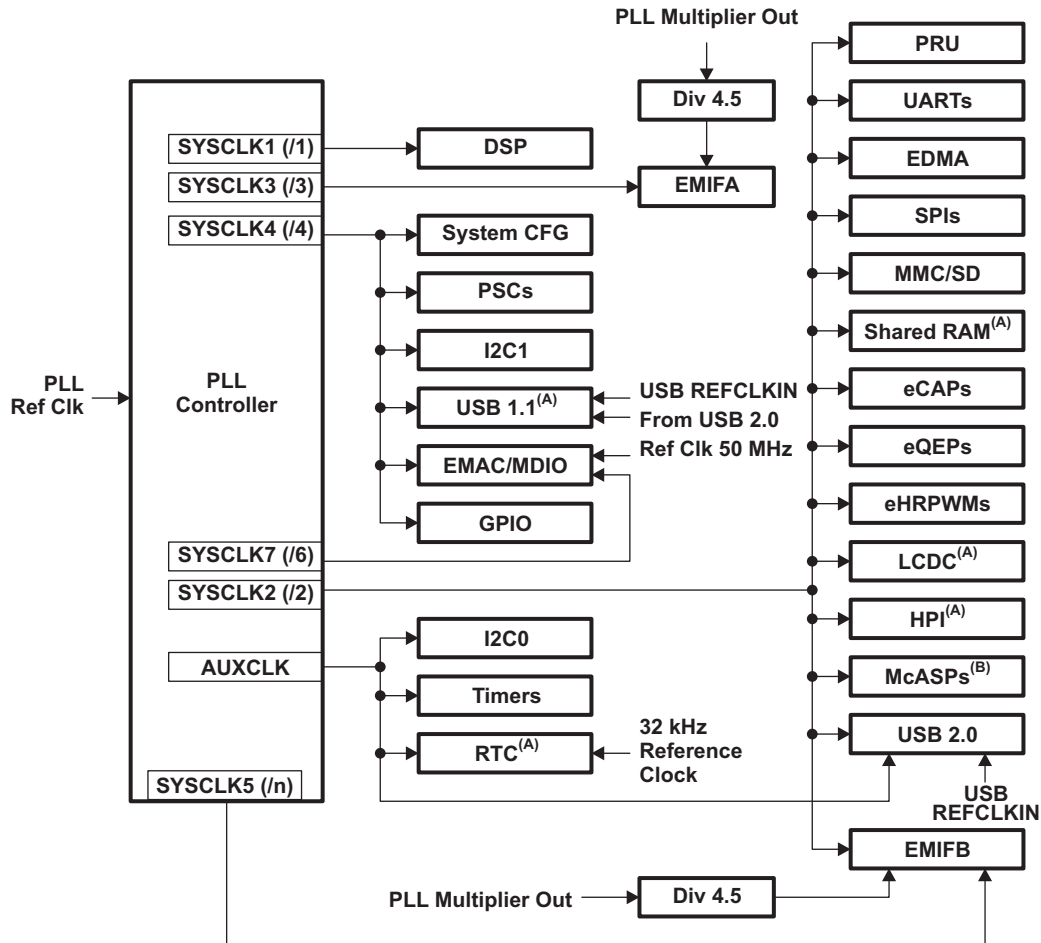
Table 6-2. System Clock Domains

CPU/Device Peripherals	System Clock Domain	Fixed Ratio to CPU Clock Required?	Default Ratio to CPU Clock
DSP	SYSCLK1	Yes	1:1
PRU, UARTs, EDMA, SPIs, MMC/SD, Shared RAM ⁽¹⁾ , eCAPs, eQEPs, eHRPWMs, LCD ⁽¹⁾ , HPI ⁽¹⁾ , McASPs ⁽²⁾ , USB2.0, EMIFB	SYSCLK2	Yes	1:2
EMIFA	SYSCLK3	No	1:3
SYSCFG, PSCs, I2C1, USB1.1 ⁽¹⁾ , EMAC/MDIO, GPIO	SYSCLK4	Yes	1:4
EMIFB I/O Clock	SYSCLK5	No	1:3
EMAC	SYSCLK7	No	1:6
I2C0, Timers, McASP serial clock ⁽²⁾ , RTC ⁽¹⁾ , USB2.0	AUXCLK	Not Applicable	PLL Bypass Clock

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

⁽²⁾ McASP2 is not supported on the C6745 DSP; only McASP0 and McASP1 are supported on the C6745 DSP.

Figure 6-1. Overall Clocking Diagram



A This peripheral is not supported on the C6745 DSP.

B McASP2 is not supported on the C6745 DSP; only McASP0 and McASP1 are supported on the C6745 DSP.

6.2 Frequency Flexibility

There are two clocking modes:

- PLL Bypass that can serve as a power savings mode
- PLL Active where the PLL is enabled and multiplies the input clock up to the desired operating frequency

When the PLL is in Bypass mode, the reference clock supplied on OSCIN serves as the clock source from which all of the system clocks (SYSCLK1-SYSCLK7) are derived. This means, when the PLL is in Bypass mode, the reference clock supplied on OSCIN passes directly to the system of PLLDIV blocks that creates each of the system clocks. When the PLL operates in Active mode, the PLL is enabled and the PLL multiplier setting is used to multiply the input clock frequency supplied on the OSCIN pin up to the desired frequency. It is this multiplied frequency that all system clocks are derived from in PLL Active mode.

The output of the PLL multiplier passes through a post divider (POSTDIV) block and then is applied to the system of PLLDIV blocks that creates each of the system clock domains (SYSCLK1-SYSCLK7). Each SYSCLK has a PLLDIV block associated with it. See [Chapter 7](#) for more details on the PLL.

The combination of the PLL multiplier, POSTDIV, and PLLDIV blocks provides flexibility in the frequencies that the system clock domains support. This flexibility does have limitations, as follows:

- OSCIN input frequency is limited to a supported range.
- The output of the PLL Multiplier must be within the range specified in the device-specific data manual.
- The output of each PLLDIV block must be less than or equal to the maximum device frequency specified in the device-specific data manual.

NOTE: The above limitations are provided here as an example and are used to illustrate the recommended configuration of the PLL controller. These limitations may vary based on core voltage and between devices. See the device-specific data manual for more details.

Table 6-3 shows examples of possible PLL multiplier settings, along with the available PLL post-divider modes. The PLL post-divider modes are defined by the value programmed in the RATIO field of the PLL post-divider control register (POSTDIV). For Div1, Div2, Div3, and Div4 modes, the RATIO field would be programmed to 0, 1, 2, and 3, respectively. The Div1, Div2, Div3, and Div4 modes are shown here as an example. Additional post-divider modes are supported and are documented in [Chapter 7](#).

As shown in Table 6-3, the Div1 mode is not supported. The RATIO field in POSTDIV must always be programmed to a value greater than or equal to 1.

NOTE: PLL power consumption increases as the frequency of the output of the PLL multiplier increases. To decrease power consumption, the lowest PLL multiplier should be chosen that achieves the desired frequency. For example, if 200 MHz is the desired CPU operating frequency and the OSCIN frequency is 25 MHz; lower power consumption is achieved by choosing a PLL multiplier setting of 16 and Div2 mode instead of a PLL multiplier setting of 30 and Div3 mode, even though both of these modes would result in a CPU frequency of 200 MHz.

Table 6-3. Example PLL Frequencies

OSCIN Frequency	PLL Multiplier	Multiplier Frequency (MHz)	Div1	Div2	Div3	Div4
20	30	600	Not Supported	300	200	150
24	25	600	Not Supported	300	200	150
25	24	600	Not Supported	300	200	150
30	20	600	Not Supported	300	200	150
20	25	500	Not Supported	250	167	125
24	20	480	Not Supported	240	160	120
25	18	450	Not Supported	225	150	112.5
30	14	420	Not Supported	210	140	105
25	16	400	Not Supported	200	133	100

6.3 Peripheral Clocking

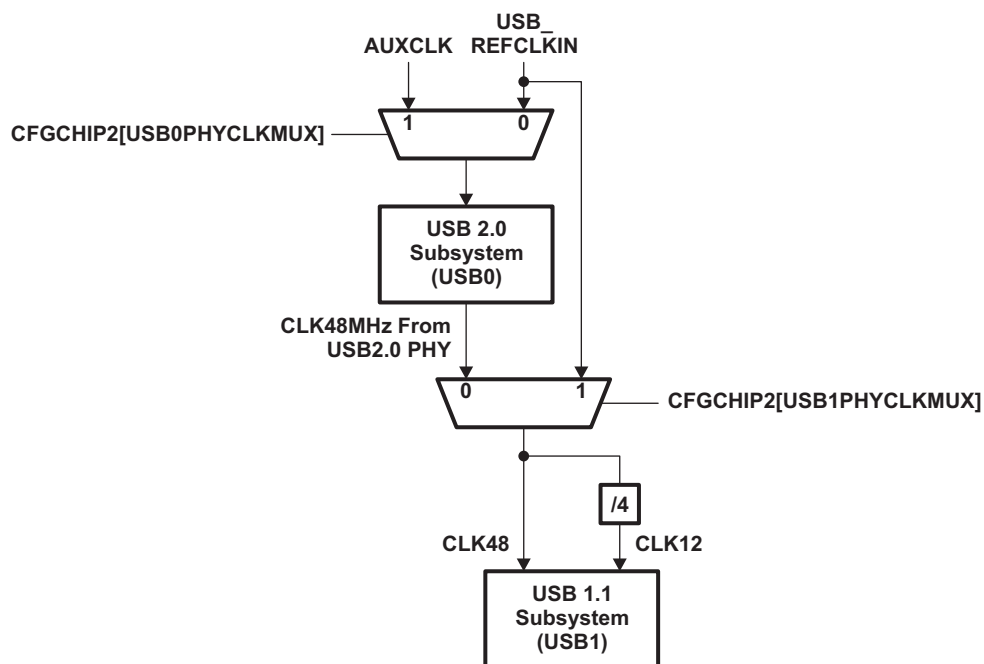
6.3.1 USB Clocking

Figure 6-2 displays the clock connections for the USB2.0 module. The USB2.0 subsystem requires a reference clock for its internal PLL. This reference clock can be sourced from either the USB_REFCLKIN pin or from the AUXCLK of the system PLL. The reference clock input to the USB2.0 subsystem is selected by programming the USB0PHYCLKMUX bit in the chip configuration 2 register (CFGCHIP2) of the System Configuration Module. The USB_REFCLKIN source should be selected when it is not possible (such as when specific audio rates are required) to operate the device at one of the allowed input frequencies to the USB2.0 subsystem. The USB2.0 subsystem peripheral bus clock is sourced from SYSCLK2.

The USB1.1 subsystem requires both a 48 MHz (CLK48) and a 12 MHz (CLK12) clock input. The 12 MHz clock is derived from the 48 MHz clock. The 48 MHz clock required by the USB1.1 subsystem can be sourced from either the USB_REFCLKIN or from the 48 MHz clock provided by the USB2.0 PHY. The CLK48 source is selected by programming the USB1PHYCLKMUX bit in CFGCHIP2 of the System Configuration Module. The USB1.1 subsystem peripheral bus clock is sourced from SYSCLK4. See Table 6-4.

NOTE: If the USB1.1 subsystem is used and the 48 MHz clock input is sourced from the USB2.0 PHY, then the USB2.0 must be configured to always generate the 48 MHz clock. The USB0PHY_PLLON bit in CFGCHIP2 controls the USB2.0 PHY, allowing or preventing it from stopping the 48 MHz clock during USB SUSPEND. When the USB0PHY_PLLON bit is set to 1, the USB2.0 PHY is prevented from stopping the 48 MHz clock during USB SUSPEND; when the USB0PHY_PLLON bit is cleared to 0, the USB2.0 PHY is allowed to stop the 48 MHz clock during USB SUSPEND.

Figure 6-2. USB Clocking Diagram



Note: The USB1.1 is not supported on the C6745 DSP.

Table 6-4. USB Clock Multiplexing Options

CFGCHIP2. USB0PHYCLKMUX bit	CFGCHIP2. USB1PHYCLKMUX bit	USB2.0 Clock Source	USB1.1 Clock Source	Additional Conditions
0	0	USB_REFCLKIN	CLK48MHZ output from USB2.0 PHY	USB_REFCLKIN must be 12, 24, 48, 19.2, 38.4, 13, 26, 20, or 40 MHz. The PLL inside the USB2.0 PHY can be configured to accept any of these input clock frequencies.
0	1	USB_REFCLKIN	USB_REFCLKIN	USB_REFCLKIN must be 48 MHz. The PLL inside the USB2.0 PHY can be configured to accept this input clock frequency.
1	0	PLL0_AUXCLK	CLK48MHZ output from USB2.0 PHY	PLL0_AUXCLK must be 12, 24, 48, 19.2, 38.4, 13, 26, 20, or 40 MHz. The PLL inside the USB2.0 PHY can be configured to accept any of these input clock frequencies.
1	1	PLL0_AUXCLK	USB_REFCLKIN	PLL0_AUXCLK must be 12, 24, 48, 19.2, 38.4, 13, 26, 20, or 40 MHz. The PLL inside the USB2.0 PHY can be configured to accept any of these input clock frequencies. USB_REFCLKIN must be 48 MHz.

6.3.2 EMIFB Clocking

The EMIFB requires two input clocks to source VCLK and MCLK (see [Figure 6-3](#)):

- VCLK is sourced from SYSCLK2 that clocks the peripheral bus interface of EMIFB
- MCLK, which sets the clock rate for the I/O clock (EMB_CLK), is sourced from either SYSCLK5 or DIV4P5. The EMB_CLKSRC bit in the chip configuration 3 register (CFGCHIP3) of the System Configuration Module controls whether SYSCLK5 or DIV4P5 is selected as the clock source for MCLK.

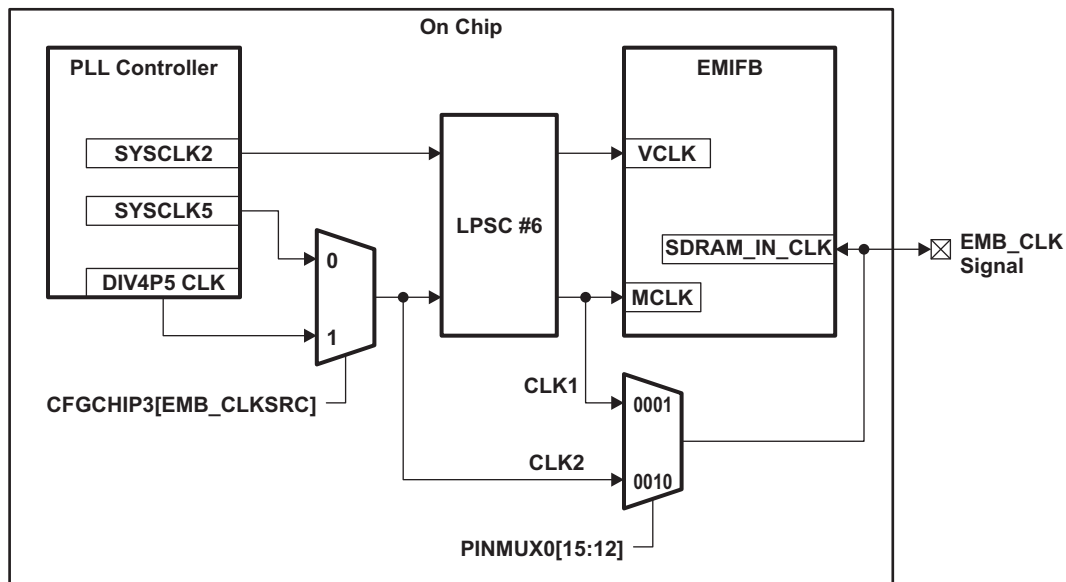
Selecting the appropriate clock source for MCLK is determined by the desired clock rate of the memory clock, EMB_CLK. [Table 6-5](#) shows example PLL register settings and the resulting DIV4P5 and SYSCLK5 frequencies based on the OSCIN reference clock frequency of 25 MHz. From these example configurations, the following observations can be made:

- To achieve the maximum frequency (133 MHz) supported by EMIFB and the typical CPU frequency of 300 MHz, the output of the PLL multiplier should be set to be 600 MHz and the EMB_CLK source should be set to DIV4P5.
- The frequency of the DIV4P5 clock is fixed at the output frequency of the PLL multiplier block divided by 4.5.
- The PLLDIV5 block that sets the divider ratio for SYSCLK5 can be changed to achieve various clock frequencies.
- For certain PLL multiplier and PLL post-divider control register (POSTDIV) settings, a higher clock frequency can be achieved by selecting SYSCLK5 as the clock source for MCLK.

As shown in [Figure 6-3](#), the EMIFB output clock, EMB_CLK, can be sourced from either the output of the EMIFB LPSC (CLK1 in [Figure 6-3](#)) or directly from the output of the clock multiplexer selecting either DIV4P5 or SYSCLK5 (CLK2 in [Figure 6-3](#)). The PINMUX0_15_12 bits in the pin multiplexing control 0 register (PINMUX0) of the SCM control this clock selection.

The purpose in providing two clock sources for EMB_CLK is to support the ability to generate a free running clock that could be used by an FPGA or for some other purpose. The difference between CLK1 and CLK2 is that if LPSC #6 is configured to clock gate the EMIFB, then CLK1 will also be clock gated, but CLK2 will not be clock gated. Therefore, if EMIFB is being used to interface to an SDRAM memory, it is best practice to choose CLK1 as the source for EMB_CLK. This will allow the maximum power savings when the LPSC is used to clock gate the EMIFB clock. If EMIFB is not in use and the EMB_CLK is used in the application as a free running clock, then CLK2 should be used as the source for EMB_CLK. This will allow clock gating of the majority of the logic in EMIFB via the LPSC while still providing a clock on the EMB_CLK.

NOTE: EMB_CLK is only an output clock. EMIFB does not support an externally provided input clock.

Figure 6-3. EMIFB Clocking Diagram

Table 6-5. EMIFB MCLK Frequencies

OSCIN Frequency	PLL Multiplier Register Setting	Multiplier Frequency (MHz)	Post Divider Mode ⁽¹⁾	POSTDIV Output Frequency	DIV4P5	PLLDIV5 Register Setting	SYSCLK5
25	24	600	Div2	300 MHz	133 MHz	2	100 MHz
			Div3	200 MHz	133 MHz	2	66.6 MHz
						1	100 MHz
			Div4	150 MHz	133 MHz	1	75 MHz
25	18	450	Div2	225 MHz	100 MHz	2	75 MHz
			Div3	150 MHz	100 MHz	1	75 MHz
						0	112.5 MHz
			Div4	112.5 MHz	100 MHz	1	56.3 MHz
25	16	400	Div2	200 MHz	89 MHz	2	66.6 MHz
			Div3	133 MHz	89 MHz	0	133 MHz
						1	100 MHz
			Div4	100 MHz	89 MHz	0	133 MHz

⁽¹⁾ See [Section 6.2](#) for an explanation of POSTDIV divider modes.

6.3.3 EMIFA Clocking

EMIFA requires a single input clock source. The EMIFA clock can be sourced from either SYSCLK3 or DIV4P5 (see Figure 6-4). The EMA_CLKSRC bit in the chip configuration 3 register (CFGCHIP3) of the System Configuration Module controls whether SYSCLK3 or DIV4P5 is selected as the clock source for EMIFA.

Selecting the appropriate clock source for EMIFA is determined by the desired clock rate. Table 6-6 shows example PLL register settings and the resulting DIV4P5 and SYSCLK3 frequencies based on the OSCIN reference clock frequency of 25 MHz. From these example configurations, the following observations can be made:

- To achieve the maximum frequency (100 MHz) supported by EMIFA and the typical CPU frequency of 300 MHz, the output of the PLL multiplier should be set to 600 MHz and the EMA_CLK source should be set to SYSCLK3 with the PLLDIV3 register set to 3.
- The frequency of the DIV4P5 clock is fixed at the output frequency of the PLL multiplier block divided by 4.5.
- The PLLDIV3 block that sets the divider ratio for SYSCLK3 can be changed to achieve various clock frequencies.

Figure 6-4. EMIFA Clocking Diagram

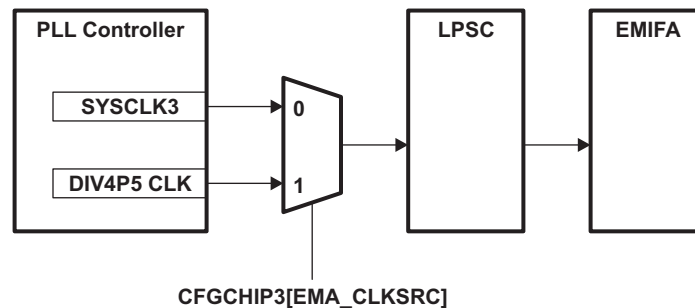


Table 6-6. EMIFA Frequencies

OSCIN Frequency	PLL Multiplier Register Setting	Multiplier Frequency (MHz)	Post Divider Mode ⁽¹⁾	POSTDIV Output Frequency	DIV4P5	PLLDIV3 Register Setting	SYSCLK3
25	24	600	Div2	300 MHz	133 MHz ⁽²⁾	2	100 MHz
			Div3	200 MHz	133 MHz ⁽²⁾	2	66.6 MHz
						1	100 MHz
			Div4	150 MHz	133 MHz ⁽²⁾	1	75 MHz
25	18	450	Div2	225 MHz	100 MHz	3	56.3 MHz
						2	75 MHz
			Div3	150 MHz	100 MHz	1	75 MHz
				Div4	112.5 MHz	100 MHz	1
			0	112.5 MHz			
25	16	400	Div2	200 MHz	89 MHz	2	66.6 MHz
						1	100 MHz
			Div3	133 MHz	89 MHz	1	66.5 MHz
				Div4	100 MHz	89 MHz	0

⁽¹⁾ See Section 6.2 for explanation of POSTDIV divider modes.

⁽²⁾ The maximum frequency supported by EMIFA is 100 MHz. 133 MHz is outside of the supported frequency range for EMIFA and is, therefore, not supported.

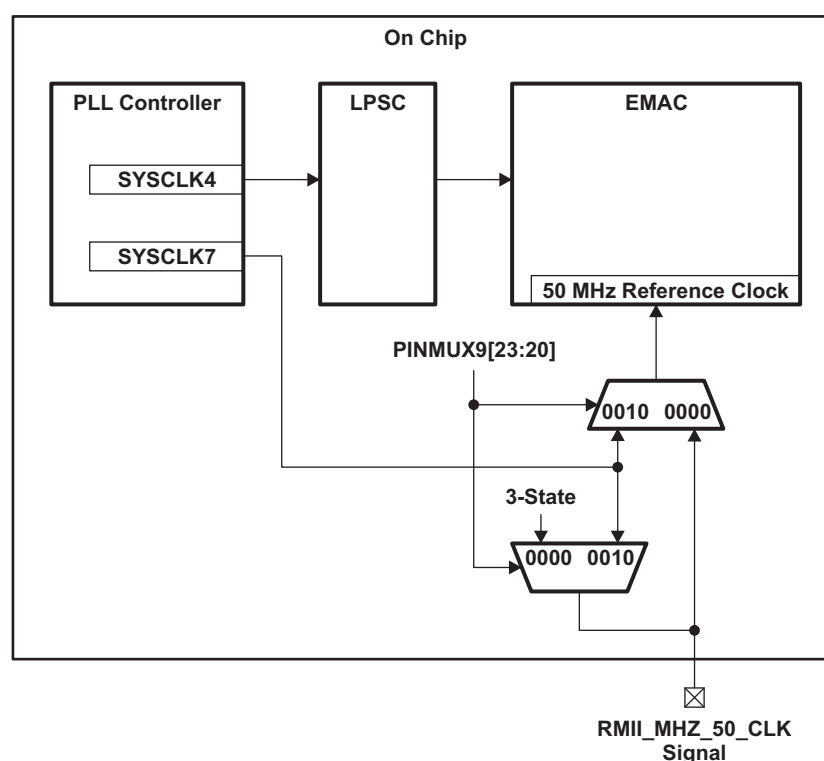
6.3.4 EMAC Clocking

The EMAC module sources its peripheral bus interface reference clock from SYSCLK4 that is at a fixed ratio of the CPU clock. The external clock requirement for EMAC varies with the interface used. When the MII interface is active, the MII_TXCLK and MII_RXCLK signals must be provided from an external source. When the RMI interface is active, the RMI 50 MHz reference clock is sourced either from an external clock on the RMI_MHZ_50_CLK pin or from SYSCLK7 (as shown in Figure 6-5). The PINMUX9_23_20 bits in the pin multiplexing control 9 register (PINMUX9) of the System Configuration Module control this clock selection:

- PINMUX9_23_20 = 0: enables sourcing of the 50 MHz reference clock from an external source on the RMI_MHZ_50_CLK pin.
- PINMUX9_23_20 = 2h: enables sourcing of the 50 MHz reference clock from SYSCLK7. Also, SYSCLK7 is driven out on the RMI_MHZ_50_CLK pin.

Table 6-7 shows example PLL register settings and the resulting SYSCLK7 frequencies based on the OSCIN reference clock frequency of 25 MHz.

Figure 6-5. EMAC Clocking Diagram



NOTE: The SYSCLK7 output clock does not meet the RMI reference clock specification of 50MHz +/-50ppm.

Table 6-7. EMAC Reference Clock Frequencies

OSCIN Frequency	PLL Multiplier Register Setting	Multiplier Frequency (MHz)	Post Divider Mode ⁽¹⁾	POSTDIV Output Frequency	PLLDIV7 Register Setting	SYSCLK7
25	24	600	Div2	300 MHz	5	50 MHz
			Div3	200 MHz	3	50 MHz
			Div4	150 MHz	2	50 MHz
25	18	450	Div2	225 MHz		Not Applicable ⁽²⁾
			Div3	150 MHz	2	50 MHz
			Div4	112.5 MHz		Not Applicable ⁽²⁾

⁽¹⁾ See [Section 6.2](#) for explanation of POSTDIV divider modes.

⁽²⁾ Certain PLL configurations do not support a 50 MHz clock on SYSCLK7.

6.3.5 I/O Domains

The I/O domains refer to the frequencies of the peripherals that communicate through device pins. In many cases, there are frequency requirements for a peripheral pin interface that are set by an outside standard and must be met. It is not necessarily possible to obtain these frequencies from the on-chip clock generation circuitry, so the frequencies must be obtained from external sources and are asynchronous to the CPU frequency by definition.

Peripherals can be divided into 4 groups, depending upon their clock requirements, as shown in [Table 6-8](#).

Table 6-8. Peripherals

Peripheral Group	Peripheral Group Definition	Peripherals Contained within Group	Source of Peripheral Clock
RTC	Operates off of a dedicated 32 kHz crystal oscillator.	RTC ⁽¹⁾	—
Fixed-Frequency Peripherals	As the name suggests, fixed-frequency peripherals have a fixed-frequency. They are fed the AUXCLK directly from the oscillator input.	Timers	—
		I2C0	—
Synchronous Peripherals	Synchronous peripherals have their frequencies derived from the CPU clock frequency. The peripheral system clock frequency changes accordingly, if the PLL1 frequency changes. Most synchronous peripherals have internal dividers so they can generate their required clock frequencies.	eCAP	—
		eQEP	—
		eHRPWM	—
		MMC/SD	—
		UARTs	—
		GPIO	—
Asynchronous Peripherals	Asynchronous peripherals are not required to operate at a fixed ratio of the CPU clock.	HPI ⁽¹⁾	—
		LCDC ⁽¹⁾	—
		EMIFA	DIV4P5 or SYSCLK3
		EMIFB	DIV4P4 or SYSCLK5
Synchronous/Asynchronous Peripherals	Synchronous/asynchronous peripherals can be run with either internally generated synchronous clocks, or externally generated asynchronous clocks.	McASPs ⁽²⁾	AUXCLK or Peripheral Serial Clocks
		SPIs	SYSCLK2 or Peripheral Serial Clock
		I2C1	SYSCLK4 or Peripheral Serial Clock
		USB ⁽¹⁾	USB_REF_CLK or AUXCLK
		EMAC	SYSCLK7 or RMII_MHZ_50_CLK

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

⁽²⁾ McASP2 is not supported on the C6745 DSP; only McASP0 and McASP1 are supported on the C6745 DSP.

Phase-Locked Loop Controller (PLL)

Topic	Page
7.1 Introduction	68
7.2 PLL0 Control	68
7.3 Locking/Unlocking PLL Register Access	72
7.4 PLLC Registers	73

7.1 Introduction

This device has one phase-locked loop (PLL) controller, PLL0, that provides a clock to different parts of the system. PLL0 provides clocks (through various dividers) to most of the components of the device.

The PLL0 provides the following:

- Glitch-Free Transitions (on changing clock settings)
- Domain Clocks Alignment
- Clock Gating
- PLL power-down

The various clock outputs given by the controller are as follows:

- Domain Clocks: SYSCLK [1:n]
- Auxiliary Clock from reference clock source: AUXCLK

Various dividers that can be used are as follows:

- Pre-PLL Divider: PREDIV
- Post-PLL Divider: POSTDIV
- SYSCLK Divider: D1, ..., Dn

Various other controls supported are as follows:

- PLL Multiplier Control: PLLM
- Software programmable PLL Bypass: PLEN

7.2 PLL0 Control

PLL0 supplies the primary system clock. Software controls the PLL0 operation through the system PLL controller 0 (PLLC0) registers. [Figure 7-1](#) shows the PLL0 in the device.

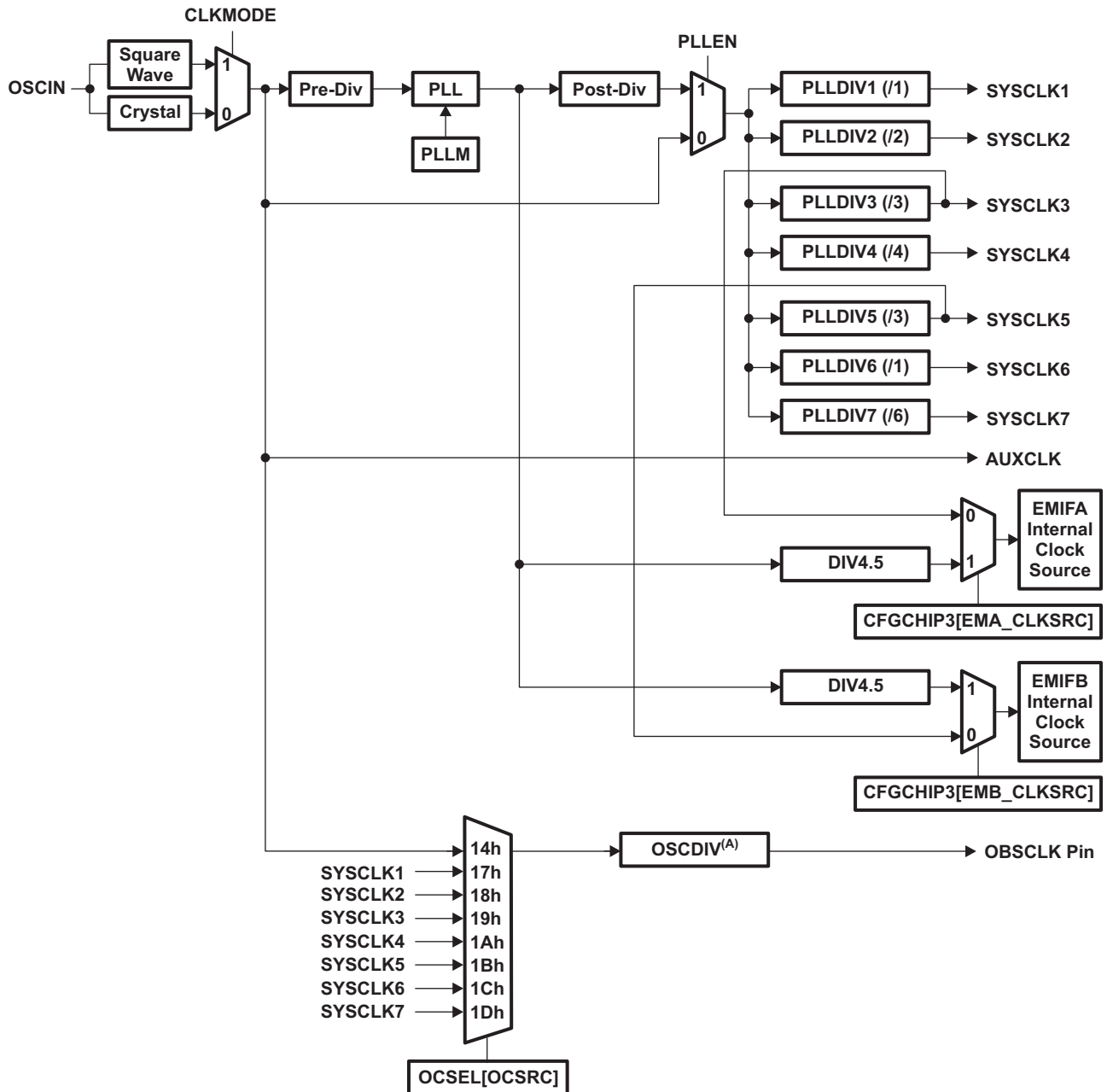
AUXCLK is the clock provided to the fixed clock domain.

The PLL0 multiplier is controlled by the PLLM bits in the PLL multiplier control register (PLLM) and is set to a default value of 0000 0013h at power-up, resulting in a PLL multiplier of 20x. The PLL0 output clock may be divided-down for slower device operation using the PLLC0 post-divider. This divider defaults to a /2 value, but may be modified by software (RATIO bit in POSTDIV) to achieve lower power device operation. These default settings yield a 300-MHz PLL output clock when using a 30-MHz clock source. The PLL0 multiplier may be modified by software.

At power-up, PLL0 is powered-down/disabled and must be powered-up by software through the PLLPWRDN bit in the PLL control register (PLLCTL). The system operates in bypass mode by default and the system clock (OSCIN) is provided directly from an input reference clock (square wave or internal oscillator) selected by the CLKMODE bit in PLLCTL. Once the PLL is powered-up and locked, software can switch the device to PLL mode operation (set the PLEN bit in PLLCTL to 1).

Registers used in PLLC0 are listed in [Section 7.4](#).

Figure 7-1. PLL0 Structure



A This register is not supported on the C6745 DSP.

7.2.1 Device Clock Generation

PLL0 is controlled by PLL controller 0. The PLLC0 manages the clock ratios, alignment, and gating for the system clocks to the chip. The PLLC is responsible for controlling all modes of the PLL through software, in terms of pre-division of the clock inputs, multiply factor within the PLL, and post-division for each of the chip-level clocks from the PLL output. The PLLC also controls reset propagation through the chip, clock alignment, and test points.

PLLC0 generates several clocks from the PLL0 output clock for use by the various processors and modules. These are summarized in [Table 7-1](#). The output clock divider values SYSCLK1 to SYSCLK n are fixed. This maintains the clock ratios between the various device components no matter what reference clock (PLL or bypass) or PLL frequency is used.

Table 7-1. System PLLC0 Output Clocks

Output Clock	Used by	Default Ratio (relative to SYSCLK1)	Notes
SYSCLK1	DSP	/1	Fixed Ratio
SYSCLK2	EDMA, DSP ports, EMIFB (bus ports), eCAPs, eHRPWMs, eQEPs, Shared RAM ⁽¹⁾ , LCD ⁽¹⁾ , McASPs ⁽²⁾ , SPIs, MMC/SD, HPI ⁽¹⁾ , USB2.0, UARTs, PRU	/2	Fixed Ratio
SYSCLK3	EMIFA	/3	No Required Ratio
SYSCLK4	System configuration (SYSCFG), PLLC0, PSCs, EMAC/MDIO, GPIO, I2C1, USB1.1 ⁽¹⁾	/4	Fixed Ratio
SYSCLK5	EMIFB	/3	No Required Ratio
SYSCLK7	RMI clock to EMAC	/6	No Required Ratio
AUXCLK	McASP serial clock ⁽²⁾ , Timers, I2C0, RTC ⁽¹⁾ , USB2.0	PLL Bypass Clock	Not Applicable
OBSCLK ⁽³⁾	Observation clock (OBSCLK) source	Pin configurable	Not Applicable

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

⁽²⁾ McASP2 is not supported on the C6745 DSP; only McASP0 and McASP1 are supported on the C6745 DSP.

⁽³⁾ This is not supported on the C6745 DSP. On the C6747 DSP, this is only available on the ZKB device package type.

- The divide values in PLL controller 0 for SYSCLK1/SYSCLK6, SYSCLK2, and SYSCLK4 are not fixed so that you can change the divide values for power saving reasons. But you are responsible to assure that the divide ratios between these clock domains must be fixed to 1:2:4.
- PLL controller supports post-divider value $n = 4.5$. When 4.5 divide values are used, the duty cycle of the resulting clock will not be 50%. In this case, the duty cycle will be 44.4%. For EMIF clock generation, see the next note.
- The DIV4P5 (/4.5) hardware clock divider is provided to generate 133 MHz from the 600 MHz PLL clock for use as clocks to the EMIFs. See [Figure 7-1](#).

7.2.2 Steps for Changing PLL0 Domain Frequency

Refer to the appropriate subsection on how to program the PLL0/Core Domain clocks:

- If the PLL is powered down (PLLPWDN bit in PLLCTL is set to 1), follow the full PLL initialization procedure in [Section 7.2.2.1](#) to initialize the PLL.
- If the PLL is not powered down (PLLPWDN bit in PLLCTL is cleared to 0), follow the sequence in [Section 7.2.2.2](#) to change the PLL multiplier.
- If the PLL is already running at a desired multiplier and you only want to change the SYSCLK dividers, follow the sequence in [Section 7.2.2.3](#).

Note that the PLL is powered down after the following device-level global resets:

- Power-on Reset (POR)
- Warm Reset (RESET)
- Max Reset

7.2.2.1 Initializing PLL Mode from PLL Power Down

If the PLL is powered down (PLLPWDN bit in PLLCTL is set to 1), perform the following procedure to initialize the PLL:

1. Clear the PLEN bit in PLLCTL to 0 (select PLL Bypass mode) and reset the PLL by clearing PLLRST bit in PLLCTL. Wait for 4 OSCIN cycles to ensure PLLC switches to bypass mode properly.
2. Select the clock mode by programming the CLKMODE bit in PLLCTL.
 - (a) Clear the PLENSRC bit in PLLCTL to 0 to allow PLLCTL.PLEN to take effect.
 - (b) PLLCTL.EXTCLKSRC should be left to 0.
3. Clear the PLLRST bit in PLLCTL to 0 (reset PLL).
4. Clear the PLLPWDN bit in PLLCTL to 0 to bring the PLL out of power-down mode.
5. Program the required multiplier value in PLLM. If desired to scale all the SYSCLK frequencies of a given PLLC, program the POSTDIV ratio.
6. If necessary, program PLLDIVn registers to change the SYSCLK0 to SYSCLKn divide values:
 - (a) Check for GOSTAT bit in PLLSTAT to clear to 0 to indicate that no GO operation is currently in progress.
 - (b) Program the RATIO field in PLLDIVx with the desired divide factors.
 - (c) Set the GOSET bit in PLLCMD to 1 to initiate a new divider transition.
 - (d) Wait for the GOSTAT bit in PLLSTAT to clear to 0 (completion of phase alignment).
7. Set the PLLRST bit in PLLCTL to 1 to bring the PLL out of reset.
8. Wait for PLL to lock. See the device-specific data manual for PLL lock time.
9. Set the PLEN bit in PLLCTL to 1 to remove the PLL from bypass mode.

7.2.2.2 Changing PLL Multiplier

If the PLL is not powered down (PLL_PWRDN bit in PLL_CTL is cleared to 0), perform the following procedure to change PLL0 multiplier.

1. Before changing the PLL frequency, switch to PLL bypass mode:
 - (a) Clear the PLENSRC bit in PLL_CTL to 0 to allow PLL_CTL.PLEN to take effect.
 - (b) Clear the PLEN bit in PLL_CTL to 0 (select PLL bypass mode).
 - (c) Wait for 4 OSCIN cycles to ensure PLLC switches to bypass mode properly.
2. Clear the PLLRST bit in PLL_CTL to 0 (reset PLL).
3. Program the required multiplier value in PLLM. If desired to scale all the SYSCLK frequencies of a given PLLC, program the POSTDIV ratio.
4. If necessary, program PLLDIVn registers to change the SYSCLKn divide values:
 - (a) Program the RATIO field in PLLDIVn with the desired divide factors.
 - (b) Set the GOSET bit in PLLCMD to 1 to initiate a new divider transition.
 - (c) Wait for the GOSTAT bit in PLLSTAT to clear to 0 (completion of phase alignment).
5. Set the PLLRST bit in PLL_CTL to 1 to bring the PLL out of reset.
6. Wait for PLL to lock. See the device-specific data manual for PLL lock time.
7. Set the PLEN bit in PLL_CTL to 1 to remove the PLL from bypass mode.

7.2.2.3 Changing SYSCLK Dividers

This section discusses the software sequence to change the SYSCLK dividers. The SYSCLK divider change sequence is also referred to as GO operation, as it involves hitting the GO bit (GOSET bit in PLLCMD) to initiate the divider change.

1. Check for the GOSTAT bit in PLLSTAT to clear to 0 to indicate that no GO operation is currently in progress.
2. Program the RATIO field in PLLDIVn with the desired divide factors.
3. Set the GOSET bit in PLLCMD to 1 to initiate a new divider transition.
4. Wait for the GOSTAT bit in PLLSTAT to clear to 0 (completion of divider change).

7.3 Locking/Unlocking PLL Register Access

A lock mechanism is present on the device that can prevent inadvertent reconfiguration of the PLLC registers. This primarily provides protection for the watchdog timer that runs on the AUXCLK output of PLL0. The PLL has a bit that is capable of disabling AUXCLK and therefore capable of stopping the watchdog timer.

To prevent this, when the PLL_MASTER_LOCK bit of the chip configuration 0 register (CFGCHIP0) in the System Configuration Module is set, writes to any PLLC registers are locked. The PLL_MASTER_LOCK bit is protected as type "Priv" and it is also protected by the Kick0 and Kick1 registers in the System Configuration Module. The master writing to the Kick0/Kick1/CFGCHIP0 registers needs to have appropriate privilege, and write the correct key values to the Kick0 and Kick 1 registers before writing to the PLLC registers. See [Chapter 10](#) for information on privilege type and the Kick0 and Kick1 registers.

To modify the PLLC registers, use the following sequence:

1. Write the correct key values to Kick0 and Kick1 registers.
2. Clear the PLL_MASTER_LOCK bit in CFGCHIP0.
3. Configure the desired PLLC register values.
4. Write an incorrect key value to the Kick registers.

NOTE: The PLL_MASTER_LOCK bit in CFGCHIP0 defaults to unlocked after reset, so the above procedure is only required after the PLL_MASTER_LOCK bit has been locked (set to 1).

7.4 PLLC Registers

[Table 7-2](#) lists the memory-mapped registers for the PLLC.

Table 7-2. PLL Controller (PLLC) Registers

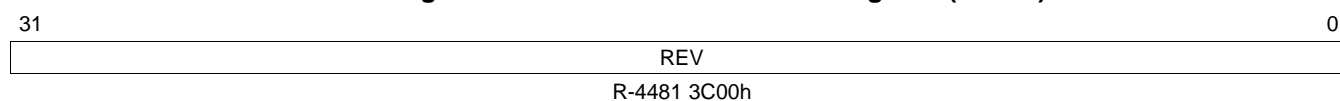
Address	Acronym	Register Description	Section
01C1 1000h	REVID	Revision Identification Register	Section 7.4.1
01C1 10E4h	RSTYPE	Reset Type Status Register	Section 7.4.2
01C1 1100h	PLLCTL	PLL Control Register	Section 7.4.3
01C1 1104h	OCSEL ⁽¹⁾	OBSCLK Select Register	Section 7.4.4
01C1 1110h	PLLM	PLL Multiplier Control Register	Section 7.4.5
01C1 1114h	PREDIV	PLL Pre-Divider Control Register	Section 7.4.6
01C1 1118h	PLLDIV1	PLL Controller Divider 1 Register	Section 7.4.7
01C1 111Ch	PLLDIV2	PLL Controller Divider 2 Register	Section 7.4.8
01C1 1120h	PLLDIV3	PLL Controller Divider 3 Register	Section 7.4.9
01C1 1124h	OSCDIV ⁽¹⁾	Oscillator Divider 1 Register (OBSCLK)	Section 7.4.14
01C1 1128h	POSTDIV	PLL Post-Divider Control Register	Section 7.4.15
01C1 1138h	PLLCMD	PLL Controller Command Register	Section 7.4.16
01C1 113Ch	PLLSTAT	PLL Controller Status Register	Section 7.4.17
01C1 1140h	ALNCTL	PLL Controller Clock Align Control Register	Section 7.4.18
01C1 1144h	DCHANGE	PLLDIV Ratio Change Status Register	Section 7.4.19
01C1 1148h	CKEN	Clock Enable Control Register	Section 7.4.20
01C1 114Ch	CKSTAT	Clock Status Register	Section 7.4.21
01C1 1150h	SYSTAT	SYCLK Status Register	Section 7.4.22
01C1 1160h	PLLDIV4	PLL Controller Divider 4 Register	Section 7.4.10
01C1 1164h	PLLDIV5	PLL Controller Divider 5 Register	Section 7.4.11
01C1 1168h	PLLDIV6	PLL Controller Divider 6 Register	Section 7.4.12
01C1 116Ch	PLLDIV7	PLL Controller Divider 7 Register	Section 7.4.13
01C1 11F0h	EMUCNT0	Emulation Performance Counter 0 Register	Section 7.4.23
01C1 11F4h	EMUCNT1	Emulation Performance Counter 1 Register	Section 7.4.24

⁽¹⁾ This register is not supported on the C6745 DSP.

7.4.1 Revision Identification Register (REVID)

The revision identification register (REVID) is shown in [Figure 7-2](#) and described in [Table 7-3](#).

Figure 7-2. Revision Identification Register (REVID)



LEGEND: R = Read only; -n = value after reset

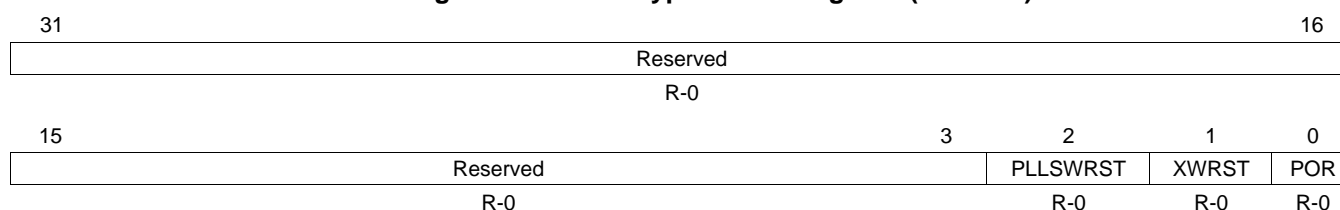
Table 7-3. Revision Identification Register (REVID) Field Descriptions

Bit	Field	Value	Description
31-0	REV	4481 3C00h	Peripheral revision ID.

7.4.2 Reset Type Status Register (RSTYPE)

The reset type status register (RSTYPE) is shown in [Figure 7-3](#) and described in [Table 7-4](#). RSTYPE latches the cause of the last reset. If multiple reset sources are asserted simultaneously, RSTYPE records the reset source that deasserts last. If multiple reset sources are asserted and deasserted simultaneously, RSTYPE latches the highest priority reset source.

Figure 7-3. Reset Type Status Register (RSTYPE)



LEGEND: R = Read only; -n = value after reset

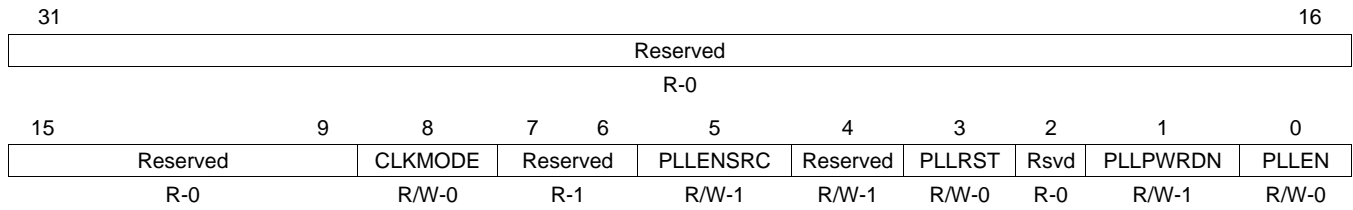
Table 7-4. Reset Type Status Register (RSTYPE) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	PLLSWRST	0	PLL software reset.
		1	PLL soft reset was not the last reset to occur.
1	XWRST	0	PLL soft was the last reset to occur.
		1	External warm reset.
0	POR	0	External warm reset was not the last reset to occur.
		1	External warm reset was the last reset to occur.
0	POR	0	Power on reset.
		1	Power On Reset (POR) was not the last reset to occur.
		1	Power On Reset (POR) was the last reset to occur.

7.4.3 PLL Control Register (PLLCTL)

The PLL control register (PLLCTL) is shown in [Figure 7-4](#) and described in [Table 7-5](#).

Figure 7-4. PLL Control Register (PLLCTL)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 7-5. PLL Control Register (PLLCTL) Field Descriptions

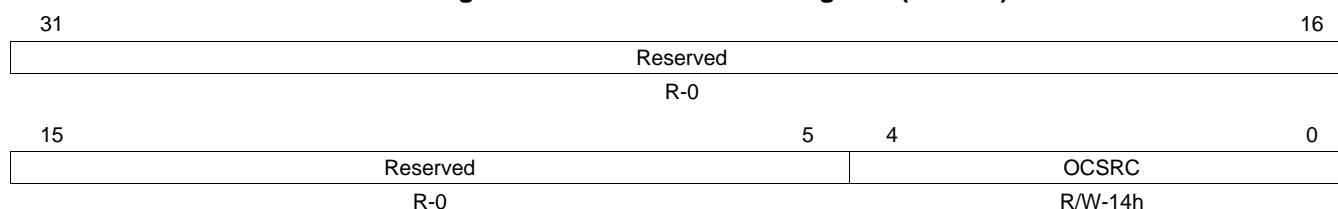
Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	CLKMODE	0 1	Reference Clock Selection Internal oscillator (crystal) Square wave
7-6	Reserved	1	Reserved
5	P LLENSRC	0	This bit must be cleared before P LLEN will have any effect.
4	Reserved	1	Reserved. Write the default value when modifying this register.
3	P LLRST	0 1	Asserts RESET to PLL if supported. PLL reset is asserted PLL reset is not asserted
2	Reserved	0	Reserved
1	P LLPWRDN	0 1	PLL power-down. PLL operation PLL power-down
0	P LLEN	0 1	PLL mode enables. Bypass mode PLL mode, not bypassed

7.4.4 OBSCLK Select Register (OCSEL)

NOTE: This register is not supported on the C6745 DSP.

The OBSCLK select register (OCSEL) controls which clock is output on the OBSCLK pin so that it may be used for test and debug purposes (in addition to its normal function of being a direct input clock divider). The OCSEL is shown in [Figure 7-5](#) and described in [Table 7-6](#).

Figure 7-5. OBSCLK Select Register (OCSEL)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

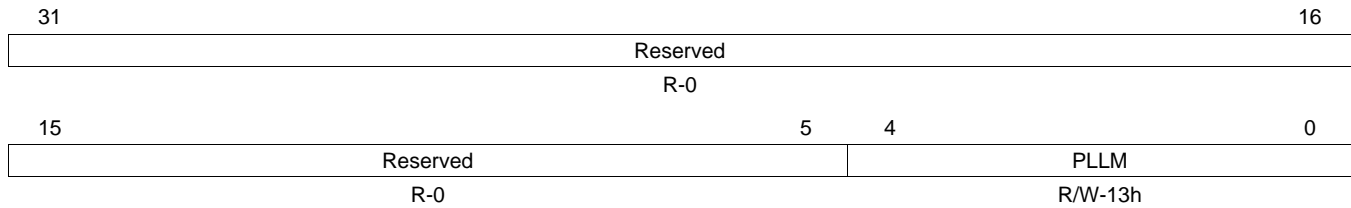
Table 7-6. OBSCLK Select Register (OCSEL) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4-0	OCSRC	0-1Fh	OBSCLK source. Output on OBSCLK pin.
		0-13h	Reserved
		14h	OSCIN
		15h-16h	Reserved
		17h	PLLC0 SYSCLK1
		18h	PLLC0 SYSCLK2
		19h	PLLC0 SYSCLK3
		1Ah	PLLC0 SYSCLK4
		1Bh	PLLC0 SYSCLK5
		1Ch	PLLC0 SYSCLK6
		1Dh	PLLC0 SYSCLK7
		1Eh	Reserved
		1Fh	Disabled

7.4.5 PLL Multiplier Control Register (PLLM)

The PLL multiplier control register (PLLM) is shown in [Figure 7-6](#) and described in [Table 7-7](#).

Figure 7-6. PLL Multiplier Control Register (PLLM)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

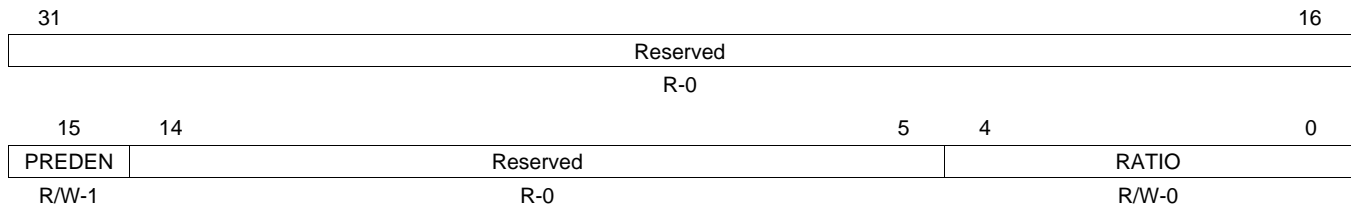
Table 7-7. PLL Multiplier Control Register (PLLM) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4-0	PLLM	0-1Fh	PLL Multiplier Select. Multiplier Value = PLLM + 1. The valid range of multiplier values for a given MXI/CLKIN is defined by the minimum and maximum frequency limits on the PLL VCO frequency. See the device-specific data manual for PLL VCO frequency specification limits.

7.4.6 PLL Pre-Divider Control Register (PREDIV)

The PLL pre-divider control register (PREDIV) is shown in [Figure 7-7](#) and described in [Table 7-8](#).

Figure 7-7. PLL Pre-Divider Control Register (PREDIV)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

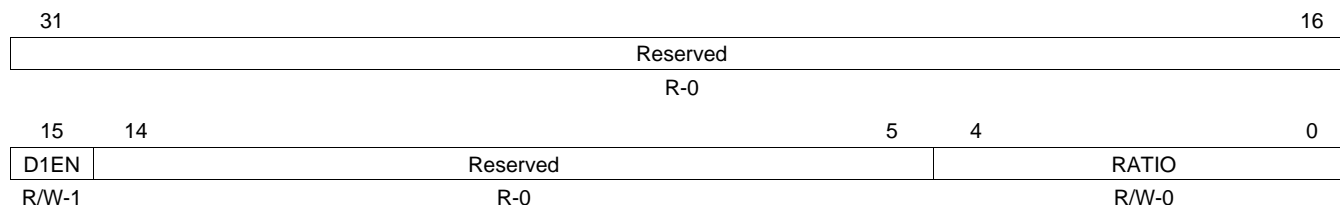
Table 7-8. PLL Pre-Divider Control Register (PREDIV) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved
15	PREDEN	0 1	Pre_Divider enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults to 0 (PLL pre-divide by 1).

7.4.7 PLL Controller Divider 1 Register (PLLDIV1)

The PLL controller divider 1 register (PLLDIV1) is shown in [Figure 7-8](#) and described in [Table 7-9](#). Divider 1 controls the divider for SYSCLK1.

Figure 7-8. PLL Controller Divider 1 Register (PLLDIV1)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

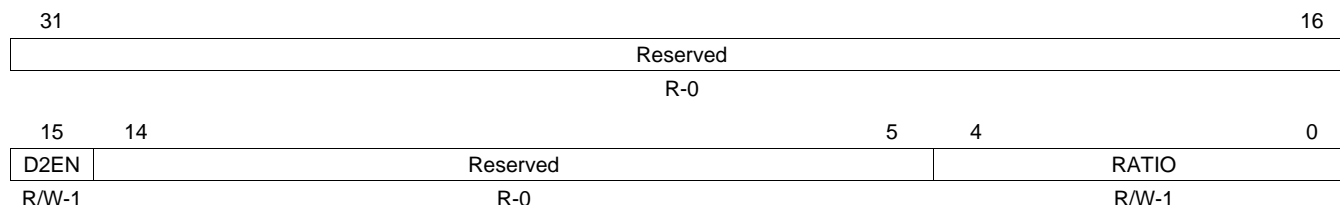
Table 7-9. PLL Controller Divider 1 Register (PLLDIV1) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	D1EN	0 1	Divider Enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults to 0 (PLL divide by 1).

7.4.8 PLL Controller Divider 2 Register (PLLDIV2)

The PLL controller divider 2 register (PLLDIV2) is shown in [Figure 7-9](#) and described in [Table 7-10](#). Divider 2 controls the divider for SYSCLK2.

Figure 7-9. PLL Controller Divider 2 Register (PLLDIV2)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 7-10. PLL Controller Divider 2 Register (PLLDIV2) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	D2EN	0 1	Divider Enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults to 1 (PLL divide by 2).

7.4.9 PLL Controller Divider 3 Register (PLLDIV3)

The PLL controller divider 3 register (PLLDIV3) is shown in [Figure 7-10](#) and described in [Table 7-11](#). Divider 3 controls the divider for SYSCLK3.

Figure 7-10. PLL Controller Divider 3 Register (PLLDIV3)

31	Reserved				16
R-0					
15	14	Reserved		5	4
D3EN				RATIO	
R/W-1		R-0		R/W-2h	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 7-11. PLL Controller Divider 3 Register (PLLDIV3) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	D3EN	0 1	Divider Enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults to 2h (PLL divide by 3).

7.4.10 PLL Controller Divider 4 Register (PLLDIV4)

The PLL controller divider 4 register (PLLDIV4) is shown in [Figure 7-11](#) and described in [Table 7-12](#). Divider 4 controls the divider for SYSCLK4.

Figure 7-11. PLL Controller Divider 4 Register (PLLDIV4)

31	Reserved				16
R-0					
15	14	Reserved		5	4
D4EN				RATIO	
R/W-1		R-0		R/W-3h	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

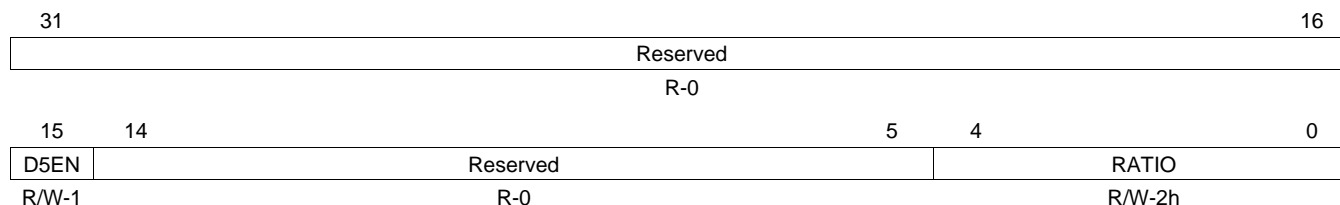
Table 7-12. PLL Controller Divider 4 Register (PLLDIV4) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	D4EN	0 1	Divider Enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults 3 (PLL divide by 4).

7.4.11 PLL Controller Divider 5 Register (PLLDIV5)

The PLL controller divider 5 register (PLLDIV5) is shown in [Figure 7-12](#) and described in [Table 7-13](#). Divider 5 controls the divider for SYSCLK5.

Figure 7-12. PLL Controller Divider 5 Register (PLLDIV5)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

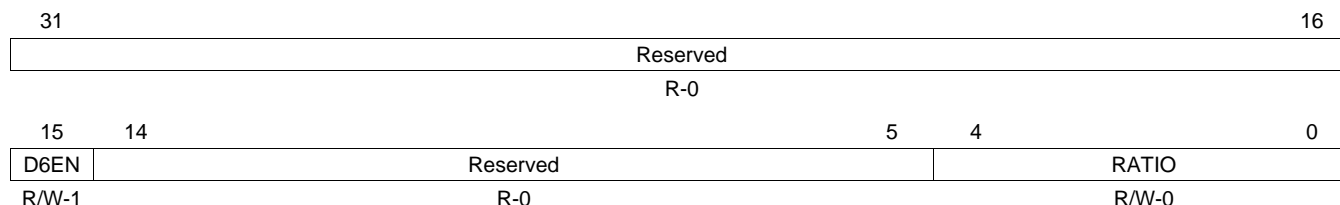
Table 7-13. PLL Controller Divider 5 Register (PLLDIV5) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	D5EN	0 1	Divider Enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults 2 (PLL divide by 3).

7.4.12 PLL Controller Divider 6 Register (PLLDIV6)

The PLL controller divider 6 register (PLLDIV6) is shown in [Figure 7-13](#) and described in [Table 7-14](#). Divider 6 controls the divider for SYSCLK6.

Figure 7-13. PLL Controller Divider 6 Register (PLLDIV6)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

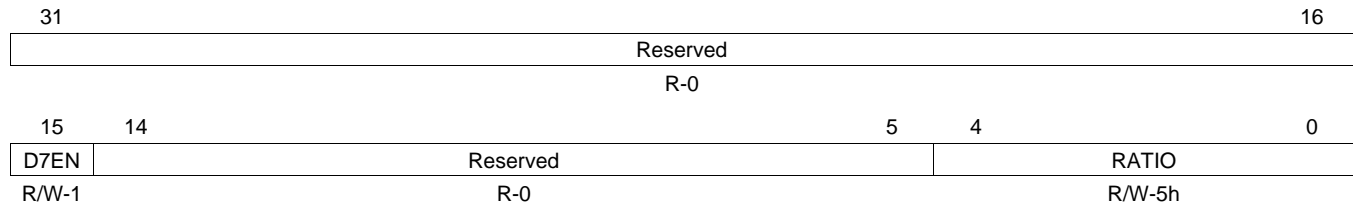
Table 7-14. PLL Controller Divider 6 Register (PLLDIV6) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	D6EN	0 1	Divider Enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults to 0 (PLL divide by 1).

7.4.13 PLL Controller Divider 7 Register (PLLDIV7)

The PLL controller divider 7 register (PLLDIV7) is shown in [Figure 7-14](#) and described in [Table 7-15](#). Divider 7 controls the divider for SYSCLK7.

Figure 7-14. PLL Controller Divider 7 Register (PLLDIV7)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 7-15. PLL Controller Divider 7 Register (PLLDIV7) Field Descriptions

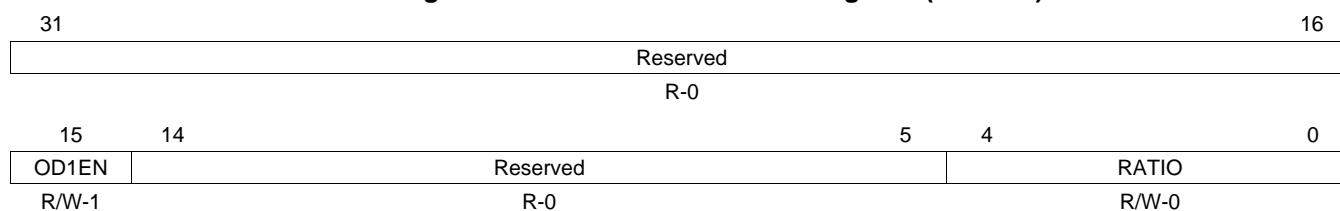
Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	D7EN	0 1	Divider Enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults to 5 (PLL divide by 6).

7.4.14 Oscillator Divider 1 Register (OSCDIV)

NOTE: This register is not supported on the C6745 DSP.

The oscillator divider 1 register (OSCDIV) controls the divider for OBSCLK, dividing down the clock selected as the OBSCLK source from the OBSCLK select register (OCSEL). The OBSCLK is connected to the OBSCLK pin. The OSCDIV is shown in [Figure 7-15](#) and described in [Table 7-16](#).

Figure 7-15. Oscillator Divider 1 Register (OSCDIV)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

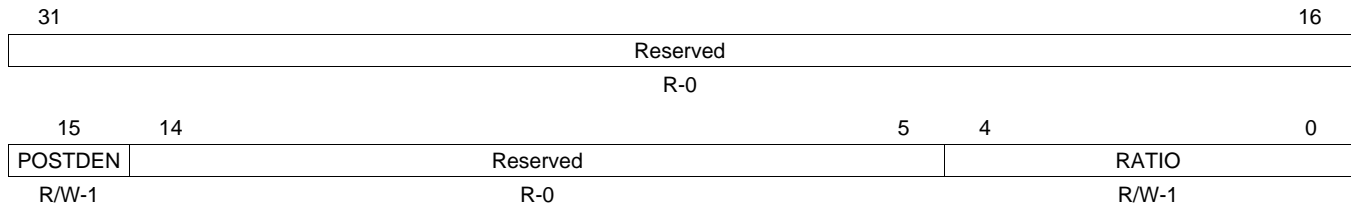
Table 7-16. Oscillator Divider 1 Register (OSCDIV) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	OD1EN	0	Oscillator divider 1 enable.
		0	Oscillator divider 1 is disabled.
		1	Oscillator divider 1 is enabled. For OBSCLK to toggle, both the OD1EN bit and the OBSEN bit in the clock enable control register (CKEN) must be set to 1.
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider value = RATIO + 1. For example, RATIO = 0 means divide by 1.

7.4.15 PLL Post-Divider Control Register (POSTDIV)

The PLL post-divider control register (POSTDIV) is shown in [Figure 7-16](#) and described in [Table 7-17](#).

Figure 7-16. PLL Post-Divider Control Register (POSTDIV)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

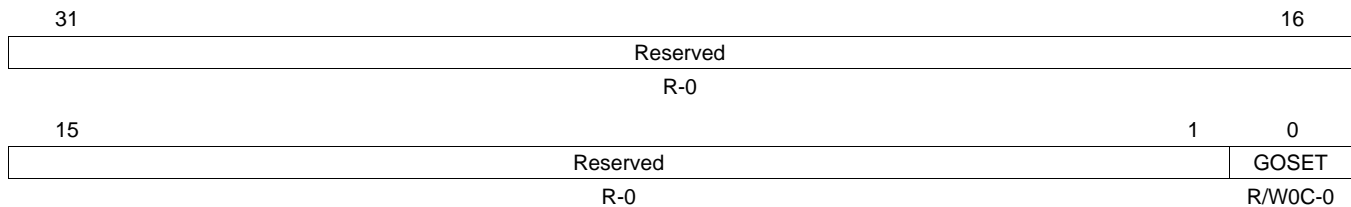
Table 7-17. PLL Post-Divider Control Register (POSTDIV) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	POSTDEN	0 1	Post_Divider enable. Disable Enable
14-5	Reserved	0	Reserved
4-0	RATIO	0-1Fh	Divider ratio. Divider Value = RATIO + 1. RATIO defaults to 1 (PLL post-divide by 2).

7.4.16 PLL Controller Command Register (PLLCMD)

The PLL controller command register (PLLCMD) is shown in [Figure 7-17](#) and described in [Table 7-18](#). contains command bits for various operations. Writes of 1 initiate command; writes of 0 clear the bit, but have no effect.

Figure 7-17. PLL Controller Command Register (PLLCMD)



LEGEND: R/W = Read/Write; R = Read only; W0C = Write 0 to clear bit; -n = value after reset

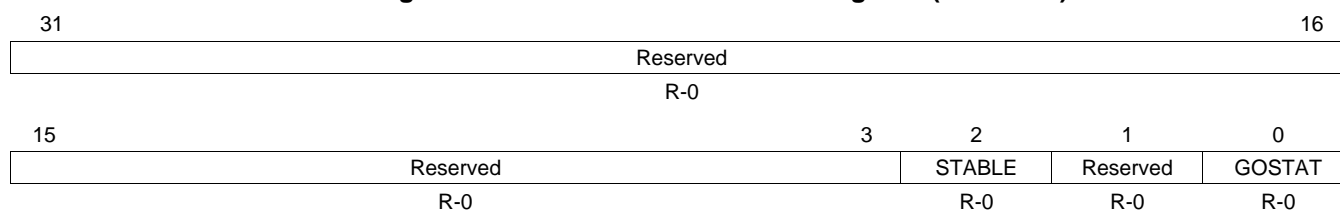
Table 7-18. PLL Controller Command Register (PLLCMD) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	GOSET	0 1	GO bit for SYSCLKx phase alignment. Clear bit (no effect) Phase alignment

7.4.17 PLL Controller Status Register (PLLSTAT)

The PLL controller status register (PLLSTAT) is shown in [Figure 7-18](#) and described in [Table 7-19](#).

Figure 7-18. PLL Controller Status Register (PLLSTAT)



LEGEND: R = Read only; -n = value after reset

Table 7-19. PLL Controller Status Register (PLLSTAT) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	STABLE	0 1	OSC counter done, oscillator assumed to be stable. By the time the device comes out of reset, this bit should become 1. No Yes
1	Reserved	0	Reserved
0	GOSTAT	0 1	Status of GO operation. If 1, indicates GO operation is in progress. GO operation is not in progress. GO operation is in progress.

7.4.18 PLL Controller Clock Align Control Register (ALNCTL)

The PLL controller clock align control register (ALNCTL) is shown in [Figure 7-19](#) and described in [Table 7-20](#). Indicates which SYSCLKs need to be aligned for proper device operation.

Figure 7-19. PLL Controller Clock Align Control Register (ALNCTL)

31	Reserved								16
R-0									
15	7	6	5	4	3	2	1	0	
Reserved		ALN7	ALN6	ALN5	ALN4	ALN3	ALN2	ALN1	
R-0		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 7-20. PLL Controller Clock Align Control Register (ALNCTL) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6	ALN7	0 1	SYSCLK7 needs to be aligned to others selected in this register. No Yes
5	ALN6	0 1	SYSCLK6 needs to be aligned to others selected in this register. No Yes
4	ALN5	0 1	SYSCLK5 needs to be aligned to others selected in this register. No Yes
3	ALN4	0 1	SYSCLK4 needs to be aligned to others selected in this register. No Yes
2	ALN3	0 1	SYSCLK3 needs to be aligned to others selected in this register. No Yes
1	ALN2	0 1	SYSCLK2 needs to be aligned to others selected in this register. No Yes
0	ALN1	0 1	SYSCLK1 needs to be aligned to others selected in this register. No Yes

7.4.19 PLLDIV Ratio Change Status Register (DCHANGE)

The PLLDIV ratio change status register (DCHANGE) is shown in [Figure 7-20](#) and described in [Table 7-21](#). Indicates if SYSCLK divide ratio has been modified.

Figure 7-20. PLLDIV Ratio Change Status Register (DCHANGE)

31	Reserved								16					
R-0														
15	Reserved						7	6	5	4	3	2	1	0
R-0							SYS7	SYS6	SYS5	SYS4	SYS3	SYS2	SYS1	
R-0							R-0	R-0	R-0	R-0	R-0	R-0	R-0	

LEGEND: R = Read only; -n = value after reset

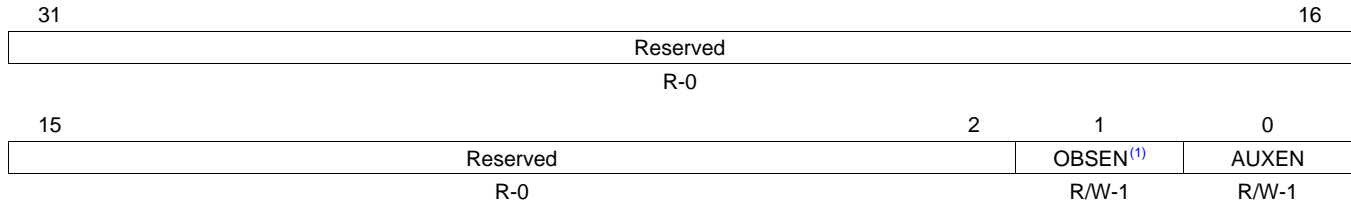
Table 7-21. PLLDIV Ratio Change Status Register (DCHANGE) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6	SYS7	0	SYSCLK7 divide ratio is modified.
		1	Ratio is not modified.
5	SYS6	0	SYSCLK6 divide ratio is modified.
		1	Ratio is not modified.
4	SYS5	0	SYSCLK5 divide ratio is modified.
		1	Ratio is not modified.
3	SYS4	0	SYSCLK4 divide ratio is modified.
		1	Ratio is not modified.
2	SYS3	0	SYSCLK3 divide ratio is modified.
		1	Ratio is not modified.
1	SYS2	0	SYSCLK2 divide ratio is modified.
		1	Ratio is not modified.
0	SYS1	0	SYSCLK1 divide ratio is modified.
		1	Ratio is not modified.

7.4.20 Clock Enable Control Register (CKEN)

The clock enable control register (CKEN) is shown in [Figure 7-21](#) and described in [Table 7-22](#). Clock enable control for miscellaneous output clocks.

Figure 7-21. Clock Enable Control Register (CKEN)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

⁽¹⁾ This bit is not supported and is Reserved on the C6745 DSP.

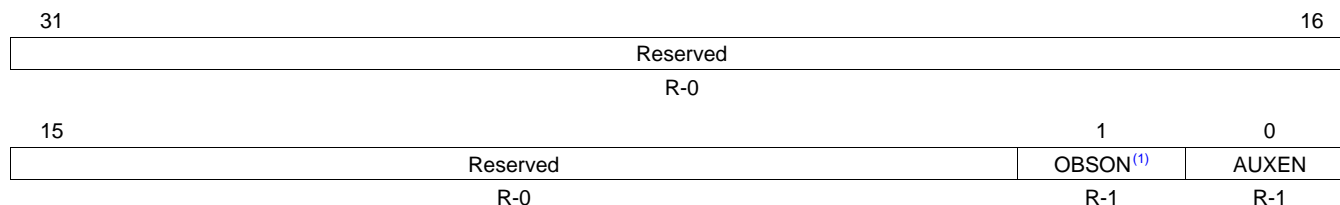
Table 7-22. Clock Enable Control Register (CKEN) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	OBSEN	0	OBSCLK enable. Actual OBSCLK status is shown in the clock status register (CKSTAT). This bit is not supported and is Reserved on the C6745 DSP. Write the default value when modifying this register.
		1	OBSCLK is disabled.
		1	OBSCLK is enabled. For OBSCLK to toggle, both the OBSEN bit and the OD1EN bit in the oscillator divider 1 register (OSCDIV) must be set to 1.
0	AUXEN	0	AUXCLK enable. Actual AUXCLK status is shown in the clock status register (CKSTAT).
		1	AUXCLK is disabled.
		1	AUXCLK is enabled.

7.4.21 Clock Status Register (CKSTAT)

The clock status register (CKSTAT) is shown in [Figure 7-22](#) and described in [Table 7-23](#). Clock status for all clocks, except SYSCLKn.

Figure 7-22. Clock Status Register (CKSTAT)



LEGEND: R = Read only; -n = value after reset

⁽¹⁾ This bit is not supported and is Reserved on the C6745 DSP.

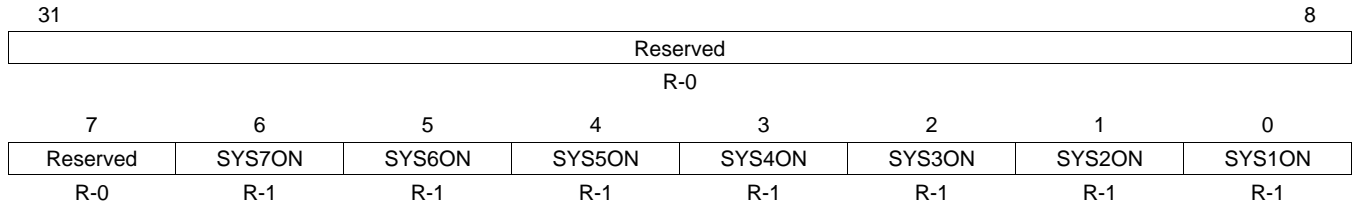
Table 7-23. Clock Status Register (CKSTAT) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	OBSON	0	OBSCCLK on status. OBSCCLK is controlled in the oscillator divider 1 register (OSCDIV) and by the OBSSEN bit in the clock enable control register (CKEN). This bit is not supported and is Reserved on the C6745 DSP.
		0	OBSCCLK is off.
		1	OBSCCLK is on.
0	AUXEN	0	AUXCLK on status. AUXCLK is controlled by the AUXEN bit in the clock enable control register (CKEN).
		0	AUXCLK is off.
		1	AUXCLK is on.

7.4.22 SYSCLK Status Register (SYSTAT)

The SYSCLK status register (SYSTAT) is shown in [Figure 7-23](#) and described in [Table 7-24](#). Indicates SYSCLK on/off status. Actual default is determined by actual clock on/off status, which depends on the DnEN bit in PLLDIVn default.

Figure 7-23. SYSCLK Status Register (SYSTAT)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

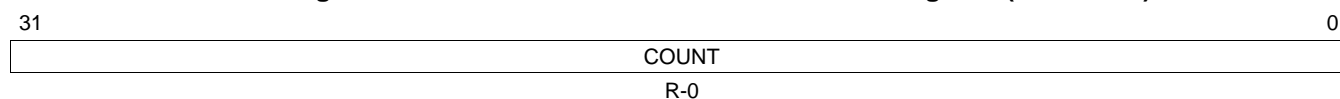
Table 7-24. SYSCLK Status Register (SYSTAT) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6	SYS7ON	0 1	SYSCLK7 on status Off On
5	SYS6ON	0 1	SYSCLK6 on status Off On
4	SYS5ON	0 1	SYSCLK5 on status Off On
3	SYS4ON	0 1	SYSCLK4 on status Off On
2	SYS3ON	0 1	SYSCLK3 on status Off On
1	SYS2ON	0 1	SYSCLK2 on status Off On
0	SYS1ON	0 1	SYSCLK1 on status Off On

7.4.23 Emulation Performance Counter 0 Register (EMUCNT0)

The emulation performance counter 0 register (EMUCNT0) is shown in [Figure 7-24](#) and described in [Table 7-25](#). EMUCNT0 is for emulation performance profiling. It counts in a divide-by-4 of the system clock. To start the counter, a write must be made to EMUCNT0. This register is not writable, but only used to start the register. After the register is started, it can not be stopped except for power on reset. When EMUCNT0 is read, it snapshots EMUCNT0 and EMUCNT1. The snapshot version is what is read. It is important to read the EMUCNT0 followed by EMUCNT1 or else the snapshot version may not get updated correctly.

Figure 7-24. Emulation Performance Counter 0 Register (EMUCNT0)



LEGEND: R = Read only; -n = value after reset

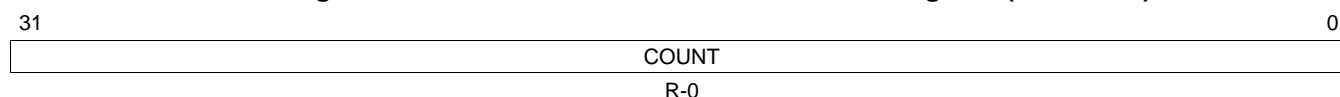
Table 7-25. Emulation Performance Counter 0 Register (EMUCNT0) Field Descriptions

Bit	Field	Value	Description
31-0	COUNT	0-FFFF FFFFh	Counter value for lower 64-bits.

7.4.24 Emulation Performance Counter 1 Register (EMUCNT1)

The emulation performance counter 1 register (EMUCNT1) is shown in [Figure 7-25](#) and described in [Table 7-26](#). EMUCNT1 is for emulation performance profiling. To start the counter, a write must be made to EMUCNT0. This register is not writable, but only used to start the register. After the register is started, it can not be stopped except for power on reset. When EMUCNT0 is read, it snapshots EMUCNT0 and EMUCNT1. The snapshot version is what is read. It is important to read the EMUCNT0 followed by EMUCNT1 or else the snapshot version may not get updated correctly.

Figure 7-25. Emulation Performance Counter 1 Register (EMUCNT1)



LEGEND: R = Read only; -n = value after reset

Table 7-26. Emulation Performance Counter 1 Register (EMUCNT1) Field Descriptions

Bit	Field	Value	Description
31-0	COUNT	0-FFFF FFFFh	Counter value for upper 64-bits.

Power and Sleep Controller (PSC)

Topic	Page
8.1 Introduction	92
8.2 Power Domain and Module Topology	92
8.3 Executing State Transitions	96
8.4 IcePick Emulation Support in the PSC	97
8.5 PSC Interrupts	97
8.6 PSC Registers	100

8.1 Introduction

The Power and Sleep Controllers (PSC) are responsible for managing transitions of system power on/off, clock on/off, resets (device level and module level). It is used primarily to provide granular power control for on chip modules (peripherals and CPU). A PSC module consists of a Global PSC (GPSC) and a set of Local PSCs (LPSCs). The GPSC contains memory mapped registers, PSC interrupts, a state machine for each peripheral/module it controls. An LPSC is associated with every module that is controlled by the PSC and provides clock and reset control. Many of the operations of the PSC are transparent to user (software), such as power on and reset control. However, the PSC module(s) also provide you with interface to control several important power, clock and reset operations. The module level power, clock and reset operations managed and controlled by the PSC are the focus of this chapter.

The PSC includes the following features:

- Manages chip power-on/off
- Provides a software interface to:
 - Control module clock enable/disable
 - Control module reset
 - Control CPU local reset
- Manages on-chip RAM sleep modes (for DSP memories and L3 RAM)
- Supports IcePick emulation features: power, clock and reset

8.2 Power Domain and Module Topology

This device includes two PSC modules. Each PSC module consists of an Always On power domain and an additional pseudo/internal power domain that manages the sleep modes for the RAMs present in the DSP subsystem and the L3 RAM, respectively.

Each PSC module controls clock states for several on the on chip modules, controllers and interconnect components. [Table 8-1](#) and [Table 8-2](#) lists the set of peripherals/modules that are controlled by the PSC, the power domain they are associated with, the LPSC assignment and the default (power-on reset) module states. See the device-specific data manual for the peripherals available on a given device. The module states and terminology are defined in [Section 8.2.2](#).

Even though there are 2 PSC modules with 2 power domains each on the device, both PSC modules and all the power domains are powered by the CVDD pins of the device. All power domains are on when the chip is powered on. There is no provision to remove power externally for the non Always On domains, that is, the pseudo/internal power domains.

There are a few modules/peripherals on the device that do not have a LPSC assigned to them. These modules do not have their module reset/clocks controlled by the PSC module. The decision to assign an LPSC to a module on a device is primarily based on whether or not disabling the clocks to a module will result in significant power savings. This typically depends on the size and the frequency of operation of the module.

Table 8-1. PSC0 Default Module Configuration

LPSC Number	Module Name	Power Domain	Default Module State	Auto Sleep/Wake Only
0	EDMA3 Channel Controller	AlwaysON (PD0)	SwRstDisable	—
1	EDMA3 Transfer Controller 0	AlwaysON (PD0)	SwRstDisable	—
2	EDMA3 Transfer Controller 1	AlwaysON (PD0)	SwRstDisable	—
3	EMIFA (BR7)	AlwaysON (PD0)	SwRstDisable	—
4	SPI0	AlwaysON (PD0)	SwRstDisable	—
5	MMC/SD0	AlwaysON (PD0)	SwRstDisable	—
6-8	Not Used	—	—	—
9	UART0	AlwaysON (PD0)	SwRstDisable	—
10	Not Used	—	—	—
11	SCR1 (BR4)	AlwaysON (PD0)	Enable	Yes
12	SCR2 (BR3, BR5, BR6)	AlwaysON (PD0)	Enable	Yes

Table 8-1. PSC0 Default Module Configuration (continued)

LPSC Number	Module Name	Power Domain	Default Module State	Auto Sleep/Wake Only
13	PRU	AlwaysON (PD0)	SwRstDisable	—
14	Not Used	—	—	—
15	DSP	PD_DSP (PD1)	Enable	—

Table 8-2. PSC1 Default Module Configuration

LPSC Number	Module Name	Power Domain	Default Module State	Auto Sleep/Wake Only
0	Not Used	—	—	—
1	USB0 (USB2.0)	AlwaysON (PD0)	SwRstDisable	—
2	USB1 (USB1.1) ⁽¹⁾	AlwaysON (PD0)	SwRstDisable	—
3	GPIO	AlwaysON (PD0)	SwRstDisable	—
4	HPI ⁽¹⁾	AlwaysON (PD0)	SwRstDisable	—
5	EMAC	AlwaysON (PD0)	SwRstDisable	—
6	EMIFB (BR20)	AlwaysON (PD0)	SwRstDisable	—
7	McASP0 (+ McASP0 FIFO)	AlwaysON (PD0)	SwRstDisable	—
8	McASP1 (+ McASP1 FIFO)	AlwaysON (PD0)	SwRstDisable	—
9	McASP2 (+ McASP2 FIFO) ⁽¹⁾	AlwaysON (PD0)	SwRstDisable	—
10	SPI1	AlwaysON (PD0)	SwRstDisable	—
11	I2C1	AlwaysON (PD0)	SwRstDisable	—
12	UART1	AlwaysON (PD0)	SwRstDisable	—
13	UART2	AlwaysON (PD0)	SwRstDisable	—
14-15	Not Used	—	—	—
16	LCDC ⁽¹⁾	AlwaysON (PD0)	SwRstDisable	—
17	eHRPWM0/1/2	AlwaysON (PD0)	SwRstDisable	—
18-19	Not Used	—	—	—
20	eCAP0/1/2	AlwaysON (PD0)	SwRstDisable	—
21	eQEP0/1	AlwaysON (PD0)	SwRstDisable	—
22-23	Not Used	—	—	—
24	SCR8 (BR15)	AlwaysON (PD0)	Enable	Yes
25	SCR7 (BR12)	AlwaysON (PD0)	Enable	Yes
26	SCR12 (BR18)	AlwaysON (PD0)	Enable	Yes
27-30	Not Used	—	—	—
31	Shared RAM (BR13) ⁽¹⁾	PD_SHRAM	Enable	Yes

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

8.2.1 Power Domain States

A power domain can only be in one of the two states: ON or OFF, defined as follows:

- ON: power to the domain is on
- OFF: power to the domain is off

In this device, for both PSC0 and PSC1, the Always ON domain (or PD0 power domain), is always in the ON state when the chip is powered-on. This domain is not programmable to OFF state (See details on PDCTL register).

Additionally, for both PSC0 and PSC1, the PD1 power domains, the internal/pseudo power domain can either be in the ON state or OFF state. Furthermore, for these power domains the transition from ON to OFF state is further qualified by the PSC0/1.PDCTL1.PDMODE settings. The PDCTL1.PDMODE settings determines the various sleep mode for the on-chip RAM associated with module in the PD1 domain.

- On PSC0 PD1/PD_DSP Domain: Controls the sleep state for DSP L1 and L2 Memories
- On PSC1 PD1/PD_SHRAM Domain: Controls the sleep state for the 128K Shared RAM

NOTE: Currently programming the PD1 power domain state to OFF is not supported. You should leave both the PDCTL1.NEXT and PDCTL1.PDMODE values at default/power on reset values.

Both PD0 and PD1 power domains in PSC0 and PSC1 are powered by the CVDD pins of the device. There is no capability to individually remove voltage/power from the DSP or Shared RAM power domains.

8.2.2 Module States

The PSC defines several possible states for a module. This various states are essentially a combination of the module reset asserted or de-asserted and module clock on/enabled or off/disabled. The various module states are defined in [Table 8-3](#).

The key difference between the Auto Sleep and Auto Wake states is that once the module is configured in Auto Sleep mode, it will transition back to the clock disabled state (automatically sleep) after servicing the internal read/write access request where as in Auto Wake mode, on receiving the first internal read/write access request, the module will permanently transition from the clock disabled to clock enabled state (automatically wake).

When the module state is programmed to Disable, SwRstDisable, Auto Sleep or Auto Wake modes, where in the module clocks are off/disabled, an external event or I/O request cannot enable the clocks. For the module to appropriately respond to such external request, it would need to be reconfigured to the Enable state.

8.2.2.1 Auto Sleep/Wake Only Configurations and Limitation

NOTE: Currently no modules should be configured in Auto Sleep or Auto Wake modes. If the module clocks need to gated/disabled for power savings, you should program the module state to Disable. For Auto Sleep/Auto Wake Only modules, disabling the clock is not supported and they should be kept in their default "Enable" state.

[Table 8-1](#) and [Table 8-2](#) each have a column to indicate whether or not the LPSC configuration for a module is Auto Sleep/Wake Only. Modules that have a "Yes" marked for the Auto Sleep/Wake Only column can be programmed in software to be in Enable, Auto Sleep and Auto Wake states only; that is, if the software tries to program these modules to Disable, SyncReset, or SwRstDisable state the power sleep controller ignores these transition requests and transitions the module state to Enable.

Table 8-3. Module States

Module State	Module Reset	Module Clock	Module State Definition
Enable	De-asserted	On	A module in the enable state has its module reset de-asserted and it has its clock on. This is the normal operational state for a given module
Disable	De-asserted	Off	A module in the disabled state has its module reset de-asserted and it has its module clock off. This state is typically used for disabling a module clock to save power. This device is designed in full static CMOS, so when you stop a module clock, it retains the module's state. When the clock is restarted, the module resumes operating from the stopping point.
SyncReset	Asserted	On	A module state in the SyncReset state has its module reset asserted and it has its clock on. Generally, software is not expected to initiate this state
SwRstDisable	Asserted	Off	A module in the SwResetDisable state has its module reset asserted and it has its clock disabled. After initial power-on, several modules come up in the SwRstDisable state. Generally, software is not expected to initiate this state
Auto Sleep	De-asserted	Off	A module in the Auto Sleep state also has its module reset de-asserted and its module clock disabled, similar to the Disable state. However this is a special state, once a module is configured in this state by software, it can "automatically" transition to "Enable" state whenever there is an internal read/write request made to it, and after servicing the request it will "automatically" transition into the sleep state (with module reset re de-asserted and module clock disabled), without any software intervention. The transition from sleep to enabled and back to sleep state has some cycle latency associated with it. It is not envisioned to use this mode when peripherals are fully operational and moving data. See Section 8.2.2.1 for additional considerations, constraints, limitations around this mode.
Auto Wake	De-asserted	Off	A module in the Auto Wake state also has its module reset de-asserted and its module clock disabled, similar to the Disable state. However this is a special state, once a module is configured in this state by software, it will "automatically" transition to "Enable" state whenever there is an internal read/write request made to it, and will remain in the "Enabled" state from then on (with module reset re de-asserted and module clock on), without any software intervention. The transition from sleep to enabled state has some cycle latency associated with it. It is not envisioned to use this mode when peripherals are fully operational and moving data. See Section 8.2.2.1 for additional considerations, constraints, limitations around this mode.

8.2.2.2 Local Reset

In addition to module reset, some modules can be reset using a special local reset that is also a part of the PSC module control for resets. The modules that support the local reset are:

- DSP: When the DSP local reset is asserted the DSP internal memories (L1P, L1D and L2) are still accessible. The local reset only resets the DSP CPU core, not the rest of DSP subsystem, as the DSP module reset would. Local Reset is useful in cases where the DSP is in enable or disable state; since when module is in SyncReset or SwRstDisable state the module reset is asserted, and the module reset takes precedence over the local reset.

The procedures for asserting and de-asserting the local reset are as follows (where n corresponds to the module that supports local reset):

1. Clear the LRST bit in the module control register (MDCTL n) to 0 to assert the module's local reset.
2. Set the LRST bit in the module control register (MDCTL n) to 1 to de-assert module's local reset.

If the CPU is in the enable state, it immediately executes program instructions after reset is de-asserted.

8.3 Executing State Transitions

This section describes how to execute the state transitions modules.

8.3.1 Power Domain State Transitions

This device consists of 2 types of domain (in each PSC controller): the Always On Domain(s) and the pseudo/RAM power domain(s). The Always On power domains are always in the ON state when the chip is powered on. You are not allowed to change the power domain state to OFF.

The pseudo/RAM power domains allow internally powering down the state of the RAMs associated with these domains (L1/L2 for PD_DSP in PSC0 and Shared RAM for PD_SHRAM in PSC1) so that these RAMs can run in lower power sleep modes via the power sleep controller.

NOTE: Currently powering down the RAMs via the pseudo/RAM power domain is not supported; therefore, these domains and the RAM should be left in their default power on state.

As mentioned in [Section 8.2](#), the pseudo/RAM power domains are powered down internally, and in this context powering down does not imply removing the core voltage from pins externally.

8.3.2 Module State Transitions

This section describes the procedure for transitioning the module state (clock and reset control). Note that some peripherals have special programming requirements and additional recommended steps you must take before you can invoke the PSC module state transition. See the individual peripheral user guides for more details. For example, the external memory controller requires that you first place the SDRAM memory in self-refresh mode before you invoke the PSC module state transitions, if you want to maintain the memory contents.

The following procedure is directly applicable for all modules that are controlled via the PSC (shown in [Table 8-1](#) and [Table 8-2](#)), except for the core(s). To transition the DSP module state, there are additional system considerations and constraints that you should be aware of. These system considerations and the procedure for transitioning the DSP module state are described in details in the [Chapter 9](#).

NOTE: In the following procedure, x is 0 for modules in PD0 (Power Domain 0 or Always On domain) and x is 1 for modules in PD1 (Power Domain 1). See [Table 8-1](#) and [Table 8-2](#) for power domain associations.

The procedure for module state transitions is:

1. Wait for the GOSTAT[x] bit in PTSTAT to clear to 0. You must wait for any previously initiated transitions to finish before initiating a new transition.
2. Set the NEXT bit in MDCTL_n to SwRstDisable (0), SyncReset (1), Disable (2h), Enable (3h), Auto Sleep (4h) or Auto Wake (5h).

NOTE: You may set transitions in multiple NEXT bits in MDCTL_n in this step. Transitions do not actually take place until you set the GO[x] bit in PTCMD in a later step.

3. Set the GO[x] bit in PTCMD to 1 to initiate the transition(s).
4. Wait for the GOSTAT[x] bit in PTSTAT to clear to 0. The modules are safely in the new states only after the GOSTAT[x] bit in PTSTAT is cleared to 0.

8.4 IcePick Emulation Support in the PSC

The PSC supports IcePick commands that allow IcePick emulation tools to have some control over the state of power domains and modules. This IcePick support only applies to the following modules:

- DSP [MDCTL15]

In particular, [Table 8-4](#) shows IcePick emulation commands recognized by the PSC.

Table 8-4. IcePick Emulation Commands

Power On and Enable Features	Power On and Enable Descriptions	Reset Features	Reset Descriptions
Inhibit Sleep	Allows emulation to prevent software from transitioning the module out of the enable state.	Assert Reset	Allows emulation to assert the module's local reset.
Force Power	Allows emulation to force the power domain into an on state. Not applicable as AlwaysOn power domain is always on.	Wait Reset	Allows emulation to keep local reset asserted for an extended period of time after software initiates local reset de-assert.
Force Active	Allows emulation to force the module into the enable state.	Block Reset	Allows emulation to block software initiated local and module resets.

NOTE: When emulation tools remove the above commands, the PSC immediately executes a state transition based on the current values in the NEXT bit in PDCTL0 and the NEXT bit in MDCTL n , as set by software.

8.5 PSC Interrupts

The PSC has an interrupt that is tied to the core interrupt controller. This interrupt is named PSCINT in the interrupt map. The PSC interrupt is generated when certain IcePick emulation events occur.

8.5.1 Interrupt Events

The PSC interrupt is generated when any of the following events occur:

- Power Domain Emulation Event (applies to pseudo/RAM power domain only)
- Module State Emulation event
- Module Local Reset Emulation event

These interrupt events are summarized in [Table 8-5](#) and described in more detail in this section.

Table 8-5. PSC Interrupt Events

Interrupt Enable Bits		Interrupt Condition
Control Register	Enable Bit	
PDCTL n	EMUHIBIE	Interrupt occurs when the emulation alters the power domain state
MDCTL n	EMUHIBIE	Interrupt occurs when the emulation alters the module state
MDCTL n	EMURSTIE	Interrupt occurs when the emulation tries to alter the module's local reset

The PSC interrupt events only apply when IcePick emulation alters the state of the module from the user-programmed state in the NEXT bit in the MDCTL/PDCTL registers. IcePick support only applies to the modules listed in [Section 8.4](#); therefore, the PSC interrupt conditions only apply to those modules listed.

8.5.1.1 Power Domain Emulation Events

A power domain emulation event occurs when emulation alters the state of a power domain (does not apply to the Always On domain). Status is reflected in the EMUIHB bit in PDSTAT n . In particular, a power domain emulation event occurs under the following conditions:

- When inhibit sleep is asserted by emulation and software attempts to transition the module out of the on state
- When force power is asserted by emulation and power domain is not already in the on state
- When force active is asserted by emulation and power domain is not already in the on state

NOTE: Putting the pseudo/RAM power domain associated with DSP (PD_DSP) to off state is currently **not** supported.

8.5.1.2 Module State Emulation Events

A module state emulation event occurs when emulation alters the state of a module. Status is reflected in the EMUIHB bit in the module status register (MDSTAT n). In particular, a module state emulation event occurs under the following conditions:

- When inhibit sleep is asserted by emulation and software attempts to transition the module out of the enable state
- When force active is asserted by emulation and module is not already in the enable state

8.5.1.3 Local Reset Emulation Events

A local reset emulation event occurs when emulation alters the local reset of a module. Status is reflected in the EMURST bit in the module status register (MDSTAT n). In particular, a module local reset emulation event occurs under the following conditions:

- When assert reset is asserted by emulation although software de-asserted the local reset
- When wait reset is asserted by emulation
- When block reset is asserted by emulation and software attempts to change the state of local reset

8.5.2 Interrupt Registers

The PSC interrupt enable bits are: the EMUIHBIE bit in PDCTL1 (PSC0), the EMUIHBIE and the EMURSTIE bits in MDCTL n (where n is the modules that have IcePick emulation support, as specified in [Section 8.4](#)).

NOTE: To interrupt the CPU, the power sleep controller interrupt (PSC0_ALLINT and PSC1_ALLINT) must also be enabled in the DSP interrupt controller. For details on the DSP interrupt controller, see [Chapter 2](#).

The PSC interrupt status bits are:

- For DSP:
 - The M[15] bit in the module error pending register 0 (MERRPR0) in PSC0 module.
 - The EMUIHB and the EMURST bits in the module status register for DSP (MDSTAT15).
 - The P[1] bit in the power error pending register (PERRPR) for the pseudo/RAM power domain associated with DSP memories.

The status bit in MERRPR0 and PERRPR registers is read by software to determine which module or power domain has generated an emulation interrupt and then software can read the corresponding status bits in MDSTAT register or the PDSTAT n (PDCTL1 for pseudo/RAM power domain in PSC0) to determine which event caused the interrupt.

The PSC interrupt can be cleared by writing to bit corresponding to the module number in the module error clear register (MERRCR0), or the bit corresponding to the power domain number in the power error clear register (PERRCR) in PSC0 module.

The PSC interrupt evaluation bit is the ALLEV bit in the INTEVAL register. When set, this bit forces the PSC interrupt logic to re-evaluate event status. If any events are still active (if any status bits are set) when the ALLEV bit in the INTEVAL is set to 1, the PSC interrupt is re-asserted to the interrupt controller. Set the ALLEV bit in the INTEVAL before exiting your PSC interrupt service routine to ensure that you do not miss any PSC interrupts.

See [Section 8.6](#) for a description of the PSC registers.

8.5.3 Interrupt Handling

Handle the PSC interrupts as described in the following procedure:

First, enable the interrupt:

1. Set the EMUIHBIE bit in PDCTL n , the EMUIHBIE and the EMURSTIE bits in MDCTL n to enable the interrupt events that you want.

NOTE: The PSC interrupt is sent to the device interrupt controller when at least one enabled event becomes active.

2. Enable the power sleep controller interrupt (PSC n _ALLINT) in the device interrupt controller. To interrupt the CPU, PSC n _ALLINT must be enabled in the device interrupt controller. See [Chapter 2](#) for more information on interrupts.

The CPU enters the interrupt service routine (ISR) when it receives the interrupt.

1. Read the P[n] bit in PERRPR, and/or the M[n] bit in MERRPR0, the M[n] bit in MERRPR1, to determine the source of the interrupt(s).
2. For each active event that you want to service:
 - (a) Read the event status bits in PDSTAT n and MDSTAT n , depending on the status bits read in the previous step to determine the event that caused the interrupt.
 - (b) Service the interrupt as required by your application.
 - (c) Write the M[n] bit in MERRCR n and the P[n] bit in PERRCR to clear corresponding status.
 - (d) Set the ALLEV bit in INTEVAL. Setting this bit reasserts the PSC interrupt to the device interrupt controller, if there are still any active interrupt events.

8.6 PSC Registers

Table 8-6 lists the memory-mapped registers for the PSC0 and Table 8-7 lists the memory-mapped registers for the PSC1.

Table 8-6. Power and Sleep Controller 0 (PSC0) Registers

Address	Acronym	Register Description	Section
01C1 0000h	REVID	Revision Identification Register	Section 8.6.1
01C1 0018h	INTEVAL	Interrupt Evaluation Register	Section 8.6.2
01C1 0040h	MERRPR0	Module Error Pending Register 0 (module 0-15)	Section 8.6.3
01C1 0050h	MERRCR0	Module Error Clear Register 0 (module 0-15)	Section 8.6.5
01C1 0060h	PERRPR	Power Error Pending Register	Section 8.6.7
01C1 0068h	PERRCR	Power Error Clear Register	Section 8.6.8
01C1 0120h	PTCMD	Power Domain Transition Command Register	Section 8.6.9
01C1 0128h	PTSTAT	Power Domain Transition Status Register	Section 8.6.10
01C1 0200h	PDSTAT0	Power Domain 0 Status Register	Section 8.6.11
01C1 0204h	PDSTAT1	Power Domain 1 Status Register	Section 8.6.12
01C1 0300h	PDCTL0	Power Domain 0 Control Register	Section 8.6.13
01C1 0304h	PDCTL1	Power Domain 1 Control Register	Section 8.6.14
01C1 0400h	PDCFG0	Power Domain 0 Configuration Register	Section 8.6.15
01C1 0404h	PDCFG1	Power Domain 1 Configuration Register	Section 8.6.16
01C1 0800h- 01C1 083Ch	MDSTAT0- MDSTAT15	Module Status <i>n</i> Register (modules 0-15)	Section 8.6.17
01C1 0A00h- 01C1 0A3Ch	MDCTL0- MDCTL15	Module Control <i>n</i> Register (modules 0-15)	Section 8.6.18

Table 8-7. Power and Sleep Controller 1 (PSC1) Registers

Address	Acronym	Register Description	Section
01E2 7000h	REVID	Revision Identification Register	Section 8.6.1
01E2 7018h	INTEVAL	Interrupt Evaluation Register	Section 8.6.2
01E2 7040h	MERRPR0	Module Error Pending Register 0 (module 0-31)	Section 8.6.4
01E2 7050h	MERRCR0	Module Error Clear Register 0 (module 0-31)	Section 8.6.6
01E2 7060h	PERRPR	Power Error Pending Register	Section 8.6.7
01E2 7068h	PERRCR	Power Error Clear Register	Section 8.6.8
01E2 7120h	PTCMD	Power Domain Transition Command Register	Section 8.6.9
01E2 7128h	PTSTAT	Power Domain Transition Status Register	Section 8.6.10
01E2 7200h	PDSTAT0	Power Domain 0 Status Register	Section 8.6.11
01E2 7204h	PDSTAT1	Power Domain 1 Status Register	Section 8.6.12
01E2 7300h	PDCTL0	Power Domain 0 Control Register	Section 8.6.13
01E2 7304h	PDCTL1	Power Domain 1 Control Register	Section 8.6.14
01E2 7400h	PDCFG0	Power Domain 0 Configuration Register	Section 8.6.15
01E2 7404h	PDCFG1	Power Domain 1 Configuration Register	Section 8.6.16
01E2 7800h- 01E2 787Ch	MDSTAT0- MDSTAT31	Module Status <i>n</i> Register (modules 0-31)	Section 8.6.17
01E2 7A00h- 01E2 7A7Ch	MDCTL0- MDCTL31	Module Control <i>n</i> Register (modules 0-31)	Section 8.6.19

8.6.1 Revision Identification Register (REVID)

The revision identification register (REVID) is shown in [Figure 8-1](#) and described in [Table 8-8](#).

Figure 8-1. Revision Identification Register (REVID)



LEGEND: R = Read only; -n = value after reset

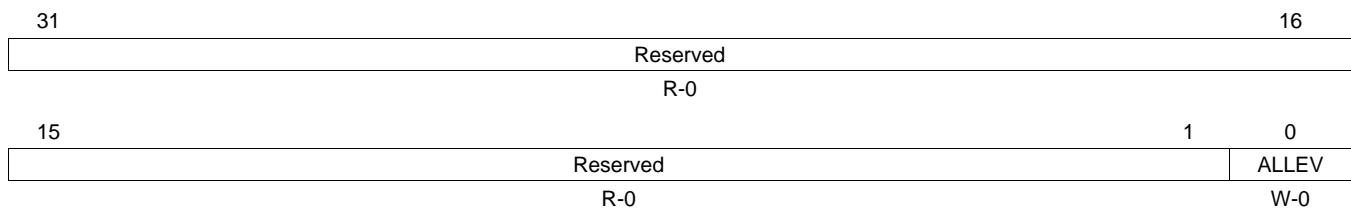
Table 8-8. Revision Identification Register (REVID) Field Descriptions

Bit	Field	Value	Description
31-0	REV	4482 3A00h	Peripheral revision ID.

8.6.2 Interrupt Evaluation Register (INTEVAL)

The interrupt evaluation register (INTEVAL) is shown in [Figure 8-2](#) and described in [Table 8-9](#).

Figure 8-2. Interrupt Evaluation Register (INTEVAL)



LEGEND: R = Read only; W= Write only; -n = value after reset

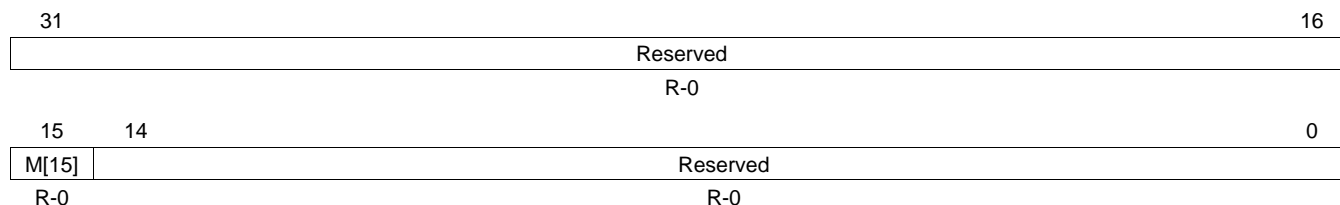
Table 8-9. Interrupt Evaluation Register (INTEVAL) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	ALLEV	0	Evaluate PSC interrupt (PSCn_ALLINT). A write of 0 has no effect.
		1	A write of 1 re-evaluates the interrupt condition.

8.6.3 PSC0 Module Error Pending Register 0 (modules 0-15) (MERRPR0)

The PSC0 module error pending register 0 (MERRPR0) is shown in [Figure 8-3](#) and described in [Table 8-10](#).

Figure 8-3. PSC0 Module Error Pending Register 0 (MERRPR0)



LEGEND: R = Read only; -n = value after reset

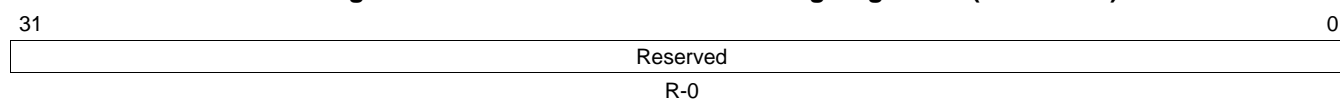
Table 8-10. PSC0 Module Error Pending Register 0 (MERRPR0) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	M[15]	0	Module 15 does not have an error condition.
		1	Module 15 has an error condition. See the module status 15 register (MDSTAT15) for the error condition.
14-0	Reserved	0	Reserved

8.6.4 PSC1 Module Error Pending Register 0 (modules 0-31) (MERRPR0)

The PSC1 module error pending register 0 (MERRPR0) is shown in [Figure 8-4](#).

Figure 8-4. PSC1 Module Error Pending Register 0 (MERRPR0)

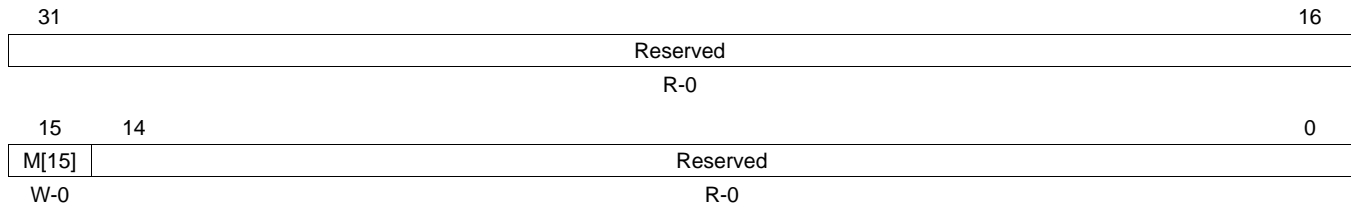


LEGEND: R = Read only; -n = value after reset

8.6.5 PSC0 Module Error Clear Register 0 (modules 0-15) (MERRCR0)

The PSC0 module error clear register 0 (MERRCR0) is shown in [Figure 8-5](#) and described in [Table 8-11](#).

Figure 8-5. PSC0 Module Error Clear Register 0 (MERRCR0)



LEGEND: R = Read only; W = Write only; -n = value after reset

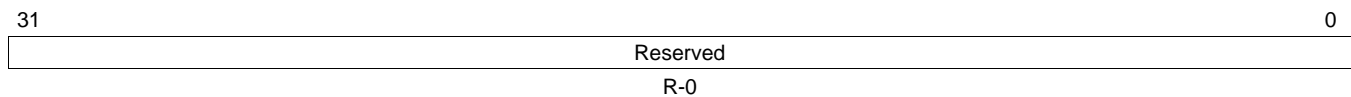
Table 8-11. PSC0 Module Error Clear Register 0 (MERRCR0) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	M[15]	0 1	Clears the interrupt status bit (M[15]) set in the PSC0 module error pending register 0 (MERRPR0) and the interrupt status bits set in the module status 15 register (MDSTAT15). A write of 0 has no effect. A write of 1 clears the M[15] bit in MERRPR0 and the EMUIHB and EMURST bits in MDSTAT15.
14-0	Reserved	0	Reserved

8.6.6 PSC1 Module Error Clear Register 0 (modules 0-31) (MERRCR0)

The PSC1 module error clear register 0 (MERRCR0) is shown in [Figure 8-6](#).

Figure 8-6. PSC1 Module Error Clear Register 0 (MERRCR0)

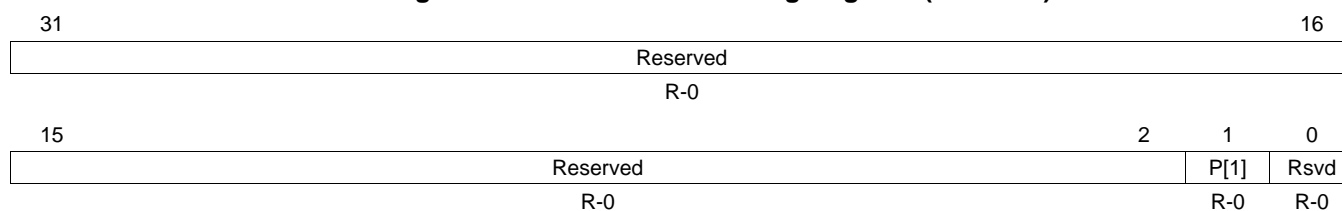


LEGEND: R = Read only; -n = value after reset

8.6.7 Power Error Pending Register (PERRPR)

The power error pending register (PERRPR) is shown in [Figure 8-7](#) and described in [Table 8-12](#).

Figure 8-7. Power Error Pending Register (PERRPR)



LEGEND: R = Read only; -n = value after reset

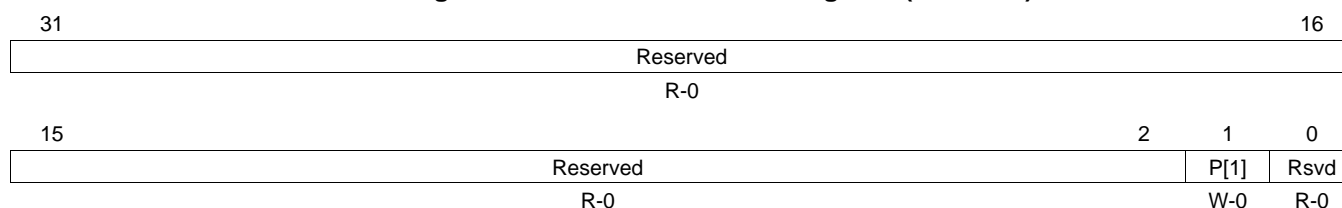
Table 8-12. Power Error Pending Register (PERRPR) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	P[1]	0	RAM/Pseudo (PD1) power domain interrupt status. RAM/Pseudo power domain does not have an error condition.
		1	RAM/Pseudo power domain has an error condition. See the power domain 1 status register (PDSTAT1) for the error condition.
0	Reserved	0	Reserved

8.6.8 Power Error Clear Register (PERRCR)

The power error clear register (PERRCR) is shown in [Figure 8-8](#) and described in [Table 8-13](#).

Figure 8-8. Power Error Clear Register (PERRCR)



LEGEND: R = Read only; W = Write only; -n = value after reset

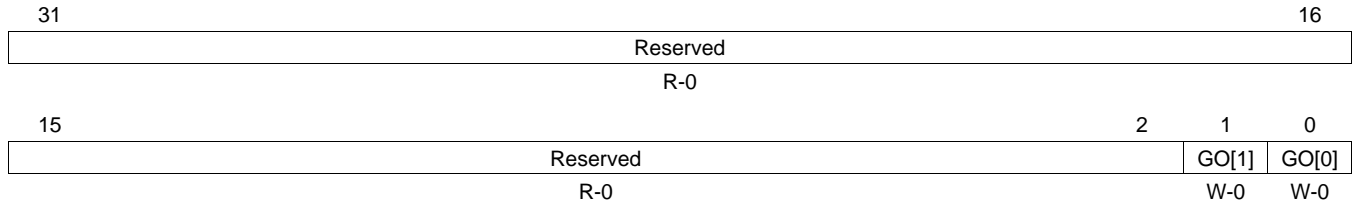
Table 8-13. Power Error Clear Register (PERRCR) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	P[1]	0	Clears the interrupt status bit (P) set in the power error pending register (PERRPR) and the interrupt status bits set in the power domain 1 status register (PDSTAT1). A write of 0 has no effect.
		1	A write of 1 clears the P bit in PERRPR and the interrupt status bits in PDSTAT1.
0	Reserved	0	Reserved

8.6.9 Power Domain Transition Command Register (PTCMD)

The power domain transition command register (PTCMD) is shown in [Figure 8-9](#) and described in [Table 8-14](#).

Figure 8-9. Power Domain Transition Command Register (PTCMD)



LEGEND: R = Read only; W = Write only; -n = value after reset

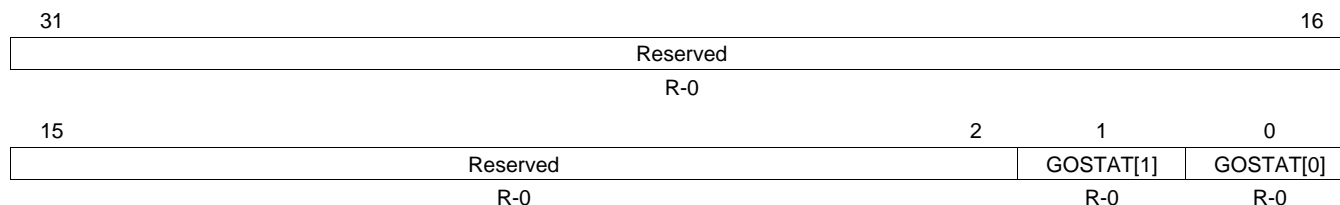
Table 8-14. Power Domain Transition Command Register (PTCMD) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	GO[1]	0	RAM/Pseudo (PD1) power domain GO transition command. A write of 0 has no effect.
		1	A write of 1 causes the PSC to evaluate all the NEXT fields relevant to this power domain (including PDCTL.NEXT for this domain, and MDCTL.NEXT for all the modules residing on this domain). If any of the NEXT fields are not matching the corresponding current state (PDSTAT.STATE, MDSTAT.STATE), the PSC will transition those respective domain/modules to the new NEXT state.
0	GO[0]	0	Always ON (PD0) power domain GO transition command. A write of 0 has no effect.
		1	A write of 1 causes the PSC to evaluate all the NEXT fields relevant to this power domain (including MDCTL.NEXT for all the modules residing on this domain). If any of the NEXT fields are not matching the corresponding current state (MDSTAT.STATE), the PSC will transition those respective domain/modules to the new NEXT state.

8.6.10 Power Domain Transition Status Register (PTSTAT)

The power domain transition status register (PTSTAT) is shown in [Figure 8-10](#) and described in [Table 8-15](#).

Figure 8-10. Power Domain Transition Status Register (PTSTAT)



LEGEND: R = Read only; -n = value after reset

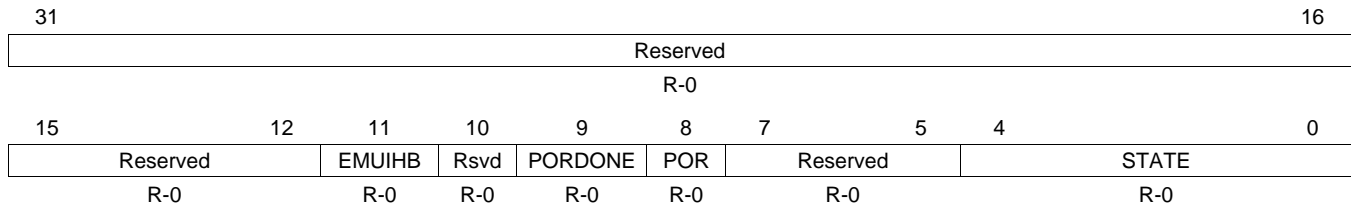
Table 8-15. Power Domain Transition Status Register (PTSTAT) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	GOSTAT[1]	0	RAM/Pseudo (PD1) power domain transition status. No transition in progress.
		1	RAM/Pseudo power domain is transitioning (that is, either the power domain is transitioning or modules in this power domain are transitioning).
0	GOSTAT[0]	0	Always ON (PD0) power domain transition status. No transition in progress.
		1	Modules in Always ON power domain are transitioning. Always On power domain is transitioning.

8.6.11 Power Domain 0 Status Register (PDSTAT0)

The power domain 0 status register (PDSTAT0) is shown in [Figure 8-11](#) and described in [Table 8-16](#).

Figure 8-11. Power Domain 0 Status Register (PDSTAT0)



LEGEND: R = Read only; -n = value after reset

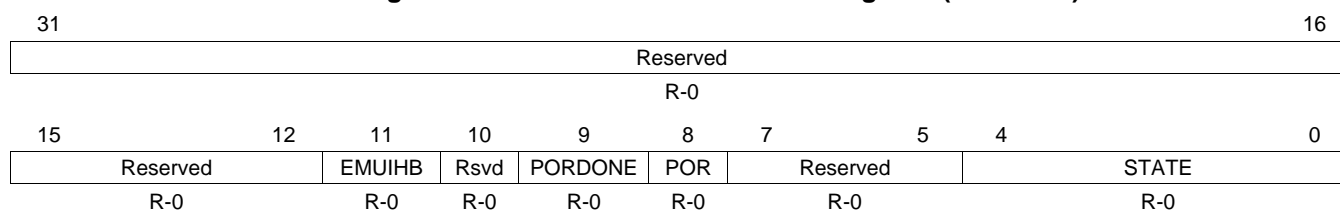
Table 8-16. Power Domain 0 Status Register (PDSTAT0) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11	EMUIHB	0	Emulation alters domain state. Interrupt is not active. No emulation altering user-desired power domain states.
		1	Interrupt is active. Emulation alters user-desired power domain state.
10	Reserved	0	Reserved
9	PORDONE	0	Power_On_Reset (POR) Done status Power domain POR is not done.
		1	Power domain POR is done.
8	POR	0	Power Domain Power_On_Reset (POR) status. This bit reflects the POR status for this power domain including all modules in the domain. Power domain POR is asserted.
		1	Power domain POR is de-asserted.
7-5	Reserved	0	Reserved
4-0	STATE	0-1Fh	Power Domain Status.
		0	Power domain is in the off state.
		1h	Power domain is in the on state.
		2h-Fh	Reserved
		10h-1Ah	Power domain is in transition.
		1Bh-1Fh	Reserved

8.6.12 Power Domain 1 Status Register (PDSTAT1)

The power domain 1 status register (PDSTAT1) is shown in [Figure 8-12](#) and described in [Table 8-17](#).

Figure 8-12. Power Domain 1 Status Register (PDSTAT1)



LEGEND: R = Read only; -n = value after reset

Table 8-17. Power Domain 1 Status Register (PDSTAT1) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11	EMUIHB	0	Emulation alters domain state. Interrupt is not active. No emulation altering user-desired power domain states.
		1	Interrupt is active. Emulation alters user-desired power domain state.
10	Reserved	0	Reserved
9	PORDONE	0	Power_On_Reset (POR) Done status Power domain POR is not done.
		1	Power domain POR is done.
8	POR	0	Power Domain Power_On_Reset (POR) status. This bit reflects the POR status for this power domain including all modules in the domain. Power domain POR is asserted.
		1	Power domain POR is de-asserted.
7-5	Reserved	0	Reserved
4-0	STATE	0-1Fh	Power Domain Status.
		0	Power domain is in the off state.
		1h	Power domain is in the on state.
		2h-Fh	Reserved
		10h-1Ah	Power domain is in transition.
		1Bh-1Fh	Reserved

8.6.13 Power Domain 0 Control Register (PDCTL0)

The power domain 0 control register (PDCTL0) is shown in [Figure 8-13](#) and described in [Table 8-18](#).

Figure 8-13. Power Domain 0 Control Register (PDCTL0)

31	Reserved										24	23	WAKECNT										16
R-0											R/W-1Fh												
15	PDMODE				12	Reserved		11	EMUIHBIE		10	Rsvd		Reserved				7	NEXT		1	0	
R-Fh				R-0		R/W-0		R-1		R-0				R/W-1									

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 8-18. Power Domain 0 Control Register (PDCTL0) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	WAKECNT	0-FFh	RAM wake count delay value. Not recommended to change the default value (1Fh). Bits 23-30: GOOD2ACCESS wake delay. Bits 19-16: ON2GOOD wake delay.
15-12	PDMODE	0-Fh 0-Eh Fh	Power down mode. Reserved Core on, RAM array on, RAM periphery on.
11-10	Reserved	0	Reserved
9	EMUIHBIE	0 1	Emulation alters power domain state interrupt enable. Disable interrupt. Enable interrupt.
8	Reserved	1	Reserved
7-1	Reserved	0	Reserved
0	NEXT	0 1	Power domain next state. For Always ON power domain this bit is read/write, but writes have no effect since internally this power domain always remains in the on state. Power domain off. Power domain on.

8.6.14 Power Domain 1 Control Register (PDCTL1)

The power domain 1 control register (PDCTL1) is shown in [Figure 8-14](#) and described in [Table 8-19](#).

Figure 8-14. Power Domain 1 Control Register (PDCTL1)

31											24	23					16	
Reserved											WAKECNT							
R-0											R/W-1Fh							
15				12	11	10			9	8			7				1	0
PDMODE				Reserved			EMUIHBIE		Rsvd		Reserved				NEXT			
R-Fh				R-0			R/W-0		R-1		R-0				R/W-1			

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

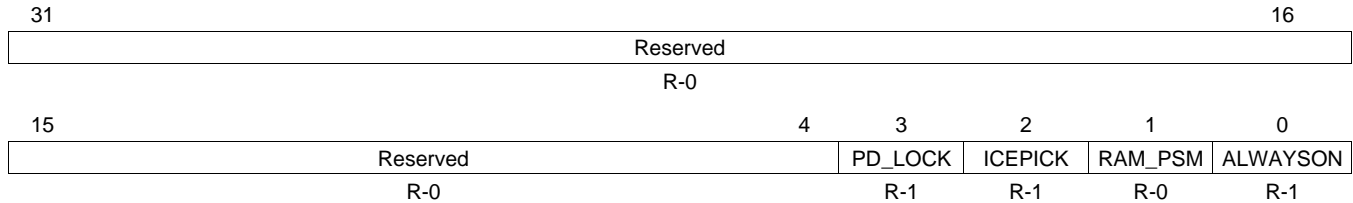
Table 8-19. Power Domain 1 Control Register (PDCTL1) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	WAKECNT	0-FFh	RAM wake count delay value. Not recommended to change the default value (1Fh). Bits 23-30: GOOD2ACCESS wake delay. Bits 19-16: ON2GOOD wake delay.
15-12	PDMODE	0-Fh	Power down mode. 0 Core off, RAM array off, RAM periphery off. 1h Core off, RAM array retention, RAM periphery off (deep sleep). 2h-3h Reserved 4h Core retention, RAM array off, RAM periphery off. 5h Core retention, RAM array retention, RAM periphery off (deep sleep). 6h-7h Reserved 8h Core on, RAM array off, RAM periphery off. 9h Core on, RAM array retention, RAM periphery off (deep sleep). Ah Core on, RAM array retention, RAM periphery off (light sleep). Bh Core on, RAM array retention, RAM periphery on. Ch-Eh Reserved Fh Core on, RAM array on, RAM periphery on.
11-10	Reserved	0	Reserved
9	EMUIHBIE	0 1	Emulation alters power domain state interrupt enable. 0 Disable interrupt. 1 Enable interrupt.
8	Reserved	1	Reserved
7-1	Reserved	0	Reserved
0	NEXT	0 1	User-desired power domain next state. 0 Power domain off. 1 Power domain on.

8.6.15 Power Domain 0 Configuration Register (PDCFG0)

The power domain 0 configuration register (PDCFG0) is shown in [Figure 8-15](#) and described in [Table 8-20](#).

Figure 8-15. Power Domain 0 Configuration Register (PDCFG0)



LEGEND: R = Read only; -n = value after reset

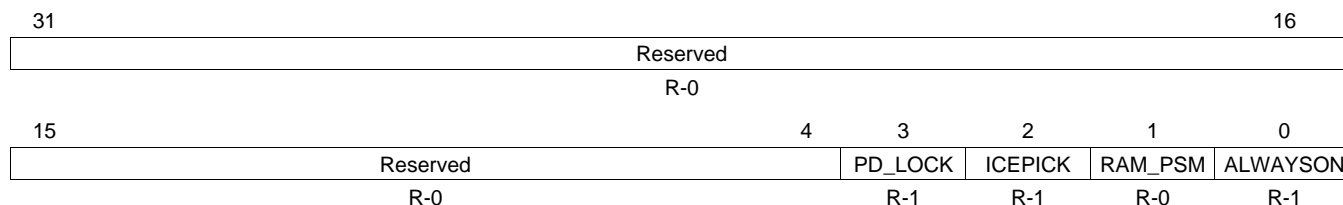
Table 8-20. Power Domain 0 Configuration Register (PDCFG0) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	PD_LOCK	0	PDCTL.NEXT lock. For Always ON power domain this bit is a don't care.
		1	PDCTL.NEXT bit is locked and cannot be changed in software.
		1	PDCTL.NEXT bit is not locked.
2	ICEPICK	0	IcePick support.
		1	Not present
		1	Present
1	RAM_PSM	0	RAM power domain.
		1	Not a RAM power domain.
		1	RAM power domain.
0	ALWAYSON	0	Always ON power domain.
		1	Not an Always ON power domain.
		1	Always ON power domain.

8.6.16 Power Domain 1 Configuration Register (PDCFG1)

The power domain 1 configuration register (PDCFG1) is shown in [Figure 8-16](#) and described in [Table 8-21](#).

Figure 8-16. Power Domain 1 Configuration Register (PDCFG1)



LEGEND: R = Read only; -n = value after reset

Table 8-21. Power Domain 1 Configuration Register (PDCFG1) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	PD_LOCK	0	PDCTL.NEXT lock. For Always ON power domain this bit is a don't care.
		1	PDCTL.NEXT bit is locked and cannot be changed in software.
		1	PDCTL.NEXT bit is not locked.
2	ICEPICK	0	IcePick support.
		1	Not present
		1	Present
1	RAM_PSM	0	RAM power domain.
		1	Not a RAM power domain.
		1	RAM power domain.
0	ALWAYSON	0	Always ON power domain.
		1	Not an Always ON power domain.
		1	Always ON power domain.

8.6.17 Module Status *n* Register (MDSTAT_{*n*})

The module status *n* register (MDSTAT_{*n*}) is shown in [Figure 8-17](#) and described in [Table 8-22](#).

Figure 8-17. Module Status *n* Register (MDSTAT_{*n*})

31														18		17	16
Reserved														EMUIHB	EMURST		
R-0														R-0	R-0		
15		13	12	11	10	9	8	7	6	5	0						
Reserved		MCKOUT	Rsvd	MRST	LRSTDONE	LRST	Reserved	STATE									
R-0		R-0	R-1	R-0	R-1	R-1	R-0	R-0									

LEGEND: R = Read only; -*n* = value after reset

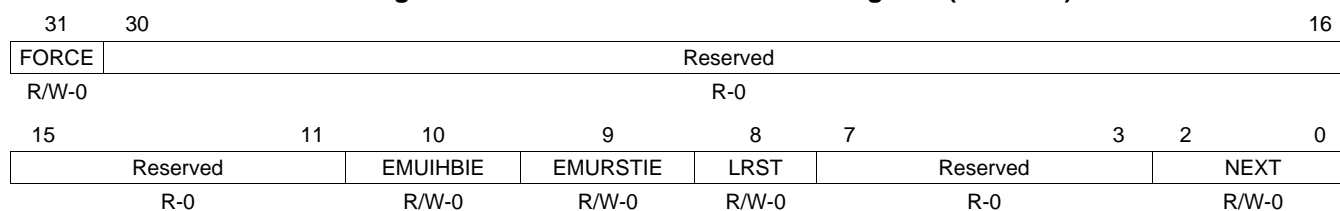
Table 8-22. Module Status *n* Register (MDSTAT_{*n*}) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved
17	EMUIHB	0 1	Emulation alters module state. This bit applies to DSP module (module 15). This field is 0 for all other modules. 0 No emulation altering user-desired module state programmed in the NEXT bit in the module control 15 register (MDCTL15). 1 Emulation altered user-desired state programmed in the NEXT bit in MDCTL15. If you desire to generate a PSCINT upon this event, you must set the EMUIHBIE bit in MDCTL15.
16	EMURST	0 1	Emulation alters module reset. This bit applies to DSP module (module 15). This field is 0 for all other modules. 0 No emulation altering user-desired module reset state. 1 Emulation altered user-desired module reset state. If you desire to generate a PSCINT upon this event, you must set the EMURSTIE bit in the module control 15 register (MDCTL15).
15-13	Reserved	0	Reserved
12	MCKOUT	0 1	Module clock output status. Shows status of module clock. 0 Module clock is off. 1 Module clock is on.
11	Reserved	1	Reserved
10	MRST	0 1	Module reset status. Reflects actual state of module reset. 0 Module reset is asserted. 1 Module reset is de-asserted.
9	LRSTDONE	0 1	Local reset done. Software is responsible for checking if local reset is done before accessing this module. This bit applies to DSP module (module 15). This field is 1 for all other modules. 0 Local reset is not done. 1 Local reset is done.
8	LRST	0 1	Module local reset status. This bit applies to DSP module (module 15). 0 Local reset is asserted. 1 Local reset is de-asserted.
7-6	Reserved	0	Reserved
5-0	STATE	0-3Fh 0 1h 2h 3h 4h-3Fh	Module state status: indicates current module status. 0 SwRstDisable state 1h SyncReset state 2h Disable state 3h Enable state 4h-3Fh Indicates transition

8.6.18 PSC0 Module Control n Register (modules 0-15) (MDCTL n)

The PSC0 module control n register (MDCTL n) is shown in [Figure 8-18](#) and described in [Table 8-23](#).

Figure 8-18. PSC0 Module Control n Register (MDCTL n)



LEGEND: R/W = Read/Write; R = Read only; - n = value after reset

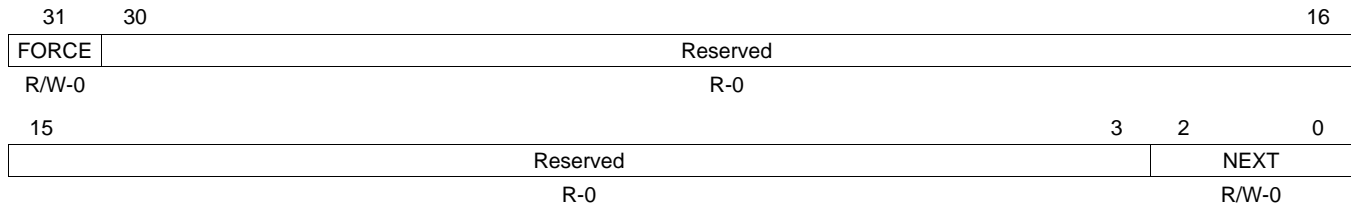
Table 8-23. PSC0 Module Control n Register (MDCTL n) Field Descriptions

Bit	Field	Value	Description
31	FORCE	0 1	Force enable. This bit forces the module state programmed in the NEXT bit in the module control 15 register (MDCTL15), ignoring and bypassing all the clock stop request handshakes managed by the PSC to change the state of the clocks to the module. Note: It is not recommended to use the FORCE bit to disable the module clock, unless specified. Force is disabled. Force is enabled.
30-11	Reserved	0	Reserved
10	EMUIHBIE	0 1	Interrupt enable for emulation alters module state. This bit applies to DSP module (module 15). Disable interrupt. Enable interrupt.
9	EMURSTIE	0 1	Interrupt enable for emulation alters reset. This bit applies to DSP module (module 15). Disable interrupt. Enable interrupt.
8	LRST	0 1	Module local reset control. This bit applies to DSP module (module 15). Assert local reset De-assert local reset
7-3	Reserved	0	Reserved
2-0	NEXT	0-3h 0 1h 2h 3h	Module next state. SwRstDisable state SyncReset state Disable state Enable state

8.6.19 PSC1 Module Control n Register (modules 0-31) (MDCTL n)

The PSC1 module control n register (MDCTL n) is shown in [Figure 8-19](#) and described in [Table 8-24](#).

Figure 8-19. PSC1 Module Control n Register (MDCTL n)



LEGEND: R/W = Read/Write; R = Read only; - n = value after reset

Table 8-24. PSC1 Module Control n Register (MDCTL n) Field Descriptions

Bit	Field	Value	Description
31	FORCE	0 1	Force enable. This bit forces the module state programmed in the NEXT bit in the module control 15 register (MDCTL15), ignoring and bypassing all the clock stop request handshakes managed by the PSC to change the state of the clocks to the module. Note: It is not recommended to use the FORCE bit to disable the module clock, unless specified. Force is disabled. Force is enabled.
30-3	Reserved	0	Reserved
2-0	NEXT	0-3h 0 1h 2h 3h	Module next state. SwRstDisable state SyncReset state Disable state Enable state

Power Management

Topic	Page
9.1 Introduction	118
9.2 Power Consumption Overview	118
9.3 Features	119
9.4 PSC and PLLC Overview	119
9.5 Clock Management	120
9.6 DSP Sleep Mode Management	121
9.7 RTC-Only Mode	121
9.8 Additional Peripheral Power Management Considerations	122

9.1 Introduction

Power management is an important aspect for most embedded applications. For several applications and target markets, there may be a specific power budget and requirements to minimize power consumption for both power supply sizing and battery life considerations. Additionally, lower power consumption results in more optimal and efficient designs from cost, design, and energy perspectives. This device has several means of managing the power consumption. This chapter discusses the various power management features.

9.2 Power Consumption Overview

Power consumed by semiconductor devices has two components: dynamic and static. This can be shown as:

$$P_{total} = P_{dynamic} + P_{static}$$

The dynamic power is the power consumed to perform work when the device is in active modes (clocks applied, busses, and I/O switching), that is, analog circuits changing states. The dynamic power is defined by:

$$P_{dynamic} = \text{Capacitance} \times \text{Voltage}^2 \times \text{Frequency}$$

From the above formula, the dynamic power scales with the clock frequency (device/module frequency for core operations and switching frequency for I/O). Dynamic power can be reduced by controlling the clocks in such a way as to either operate at a clock setting just high enough to complete the required operation in the required timeline or to run at a clock setting until the work is complete and then drastically reduce the clock frequency or cut off the clocks until additional work must be performed.

In the formula, the dynamic power varies with the voltage squared, so the voltage of operations has significant impact on overall power consumption and, thus, on the battery life. Dynamic power can be reduced by scaling the operating voltage, when the performance requirements are not that high and the device can be operated at a corresponding lower frequency.

The capacitance is the capacitance of the switching nodes, or the load capacitances on the switching I/O pins.

The static power, as the name suggests, is independent of the switching frequency of the logic. It can be shown as:

$$P_{static} = f_{(leakage\ current)}$$

It is essentially a function of the "leakage", or the power consumed by the logic when it is not switching or is not performing any work. Leakage current is dependent mostly on the manufacturing process used, the size of the die, etc. Leakage current is unavoidable while power is applied and scales roughly with the operating junction temperatures. Leakage power can only be avoided by removing power completely from a device or subsystem. The static power consumption plays a significant role in the Standby Modes (when the application is not running and in a dormant state) and plays an important role in the battery life for portable applications, etc.

9.3 Features

This device has several means of managing power consumption, as detailed in the subsequent sections. This device uses the state-of-the-art 65 nm process, which provides a good balance on power and performance, providing high-performance transistors with relatively less leakage current and, thereby, low standby-power consumption modes.

There are several features in design as well as user driven software control to reduce dynamic power consumption. The design features (not under user control) include a power optimized clock tree design to reduce overall clock tree power consumption and automatic clock gating in several modules when the logic in the modules is not active.

The on-chip power and sleep controller (PSC) module provides granular software controlled module level clock gating, which reduces both clock tree and module power by basically disabling the clocks when the modules are not being used. Clock management also allows you to slow down the clocks, to reduce the dynamic power.

[Table 9-1](#) describes the power management features.

Table 9-1. Power Management Features

Power Management	Description	Features
Clock Management		
PLL power-down	The PLL can be powered-down and run in bypass modes when not in use.	Reduces the dynamic power consumption of the core.
Module clock ON/OFF	Module clocks can be turned on/off without requiring reconfiguring the registers.	Reduces the dynamic/switching power consumption of the core and I/O (if any free running I/O clocks).
Core/module clock frequency scaling	The device can be run at a lower frequency using the PLLM/PLL dividers. Many modules have internal clock dividers to scale module/I/O frequency.	Reduces the dynamic/switching power consumption of core and I/O.
Core Sleep Management		
DSP subsystem sleep mode	The DSP CPU can be put in sleep (IDLE) mode.	Reduces the dynamic power.
Voltage Management		
RTC-only mode ⁽¹⁾	Allows removing power from all core and I/O supply and just have the real-time clock (RTC) running.	Reduces the dynamic and static power for standby modes that require only the RTC to be functional.
Peripheral I/O Power Management		
USB Phy power-down	The USB2.0 Phy can be powered-down.	Minimizes USB2.0 I/O power consumption when not in use.

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

9.4 PSC and PLLC Overview

The power and sleep controller (PSC) module plays an important role in managing the enabling/disabling of the clocks to the core and various peripheral modules. The PSC provides a granular support to turn on/off clocks on a module by module basis. Similarly, the PLL controller (PLLC) plays an important role in device and module clock generation, and manages the frequency scaling operations for the device. Together, both of these modules play a significant role in managing the clocks from a power management feature standpoint. For detailed information on the PSC, see [Chapter 8](#). For detailed information on the PLLC, see [Chapter 6](#) and [Chapter 7](#).

9.5 Clock Management

9.5.1 Module Clock ON/OFF

The module clock on/off feature allows software to disable clocks to module individually, in order to reduce the module's dynamic/switching power consumption down to zero. This device is designed in full static CMOS; thus, when a module clock stops, the module's state is preserved and retained. When the clock is restarted, the module resumes operating from the stopping point.

NOTE: Stopping clocks to a module only affects dynamic power consumption, it does not affect static power consumption of the module or the device.

The power and sleep controller (PSC) module controls module clock gating. If a module's clock(s) is stopped while being accessed, the access may not occur, and it can potentially result in unexpected behavior. The PSC provides some protection against such erroneous conditions by monitoring the internal bus activity to ensure there are no accesses to the module from the internal bus, before allowing module's internal clock to be gated. However, it is still recommended that software must ensure that all of the transactions to the module are finished prior to disabling the clocks.

The procedure to turn module clocks on/off using the PSC is described in [Chapter 8](#).

Furthermore, special consideration must be given to DSP clock on/off. The procedure to turn the core clock on/off is further described in .

Additionally some peripherals implement additional power saving features by automatically shutting of clock to components within the module , when the logic is not active. This is transparent to you, but reduces overall dynamic power consumption when modules are not active.

9.5.2 Module Clock Frequency Scaling

Module clock frequency is scalable by programming the PLL multiply and divide parameters. Additionally, some modules might also have internal clock dividers. Reducing the clock frequency reduces the dynamic/switching power consumption, which scales linearly with frequency.

[Chapter 6](#) and [Chapter 7](#) describe the how to program the PLL frequency and the frequency constraints.

9.5.3 PLL Bypass and Power Down

You can bypass the PLL in the device. Bypassing the PLL sends the PLL reference clock (MXI/CLKIN) instead of the PLL VCO output (PLLOUT) to the system clocks of the PLLC. The PLL MXI/CLKIN is typically, at most, up to 50 MHz. You can use this mode to reduce the core and module clock frequencies to very low maintenance levels without using the PLL during periods of very low system activity, this again can lower the overall dynamic/switching power consumption, which is linearly proportional to the frequency. Furthermore, you can also power-down the PLL when bypassing it to minimize the overall power consumed by the PLL module.

[Chapter 6](#) and [Chapter 7](#) describe PLL bypass and PLL power down.

9.6 DSP Sleep Mode Management

9.6.1 C674x DSP CPU Sleep Mode

The DSP CPU can be put in a low-power state by executing the IDLE instruction. For information on the IDLE instruction, see the *TMS320C674x DSP CPU and Instruction Set Reference Guide* ([SPRUFE8](#)).

9.6.2 C674x Megamodule Sleep Mode

The IDLE instruction is used as part of the procedure for shutting down the entire C674x megamodule, by the power-down controller (PDC) module. In shutting down the entire C674x megamodule, the PDC can internally clock gate off the following components of the megamodule and internal memories of the DSP subsystem:

- C674x CPU
- Program Memory Controller (PMC)
- Data Memory Controller (DMC)
- Unified Memory Controller (UMC)
- Extended Memory Controller (EMC)
- L1P Memory
- L1D Memory
- L2 Memory

Putting the entire C674x megamodule into the low-power sleep mode is typically more useful and saves a lot more power, as compared to just executing the IDLE instruction to put only the CPU in idle mode.

For information on putting the C674x megamodule in the low-power mode using the PDC, see the *TMS320C674x DSP Megamodule Reference Guide* ([SPRUFK5](#)).

9.7 RTC-Only Mode

NOTE: To put the device in RTC-only mode, there is no software control sequence. You can put the device in the RTC-only mode by removing the power supply from all core and I/O logic, except for the RTC core logic supply (RTC_CVDD).

When the rest of device is powered off, there is no up mechanism from the RTC logic to wake-up the rest of the chip or signal the external power supply on when to reapply the power. If the device is put in the RTC-only mode, then external control/decision making logic would be required to reapply power to the device.

In real-time clock (RTC)-only mode, the RTC is powered on and the rest of the device can be completely powered off (core and I/O voltage removed). In this mode, the RTC is fully functional and keeps track of date, hours, minutes, and seconds. In this mode, the overall power consumption would be significantly lower, as voltage from the rest of the core and I/O logic can be completely removed, eliminating most of the active and static power of the device, except for what is consumed by the RTC module, running at 32 kHz.

NOTE: The RTC is not supported on the C6745 DSP.

9.8 Additional Peripheral Power Management Considerations

This section lists additional power management features and considerations that might be part of other chip-level or peripheral logic, apart from the features supported by the core, PLL controller (PLL), and power and sleep controller (PSC).

9.8.1 USB PHY Power Down Control

The USB modules can be clock gated using the PSC; however, this does not power down/clock gate the PHY logic. You can put the USB2.0 PHY and OTG module in the lowest power state, when not in use, by writing to the USB0PHYPWDN and the USB0OTGPWRDN bits in the chip configuration 2 register (CFGCHIP2) of the system configuration (SYSCFG) module.

NOTE: If the USB1.1 subsystem is used and the 48 MHz clock input is sourced from the USB2.0 PHY, then the USB2.0 PHY should not be powered down.

9.8.2 EMIFB Memory Clock Gating

As discussed in [Chapter 6](#), the EMIFB output clock (EMB_CLK) can be sourced from either the output of the EMIFB LPSC (CLK1) or directly from the output of the clock multiplexer (CLK2). If the EMB_CLK is not intended to be used as a free-running clock and the EMIFB is being used as an SDRAM interface, it is recommended to use CLK1 as the source, as it allows maximal power savings (clock gating both VCLK/MCLK and EMB_CLK signal) via the PSC.

System Configuration (SYSCFG) Module

Topic	Page
10.1 Introduction	124
10.2 Protection	125
10.3 Master Priority Control	127
10.4 Host-DSP Communication Interrupts	128
10.5 SYSCFG Registers	129

10.1 Introduction

The system configuration (SYSCFG) module is a system-level module containing status and top level control logic required by the device. The system configuration module consists of a set of memory-mapped status and control registers, accessible by the CPU, supporting all of the following system features, and miscellaneous functions and operations.

- Device Identification
- Device Configuration
 - Pin multiplexing control
 - Device Boot Configuration Status
- Master Priority Control
 - Controls the system priority for all master peripherals (including EDMA3TC)
- Emulation Control
 - Emulation suspend control for peripherals that support the feature
- Special Peripheral Status and Control
 - Locking of PLL control settings
 - Default burst size configuration for EDMA3 transfer controllers
 - Event source selection for the eCAP peripheral input capture
 - McASP AMUTEIN selection and clearing of AMUTE
 - USB PHY Control
 - Clock source selection for EMIFA and EMIFB
 - HPI Control (this peripheral is not supported on the C6745 DSP)

The system configuration module controls several global operations of the device; therefore, the module supports protection against erroneous and illegal accesses to the registers in its memory-map. The protection mechanisms that are present in the module are:

- A special key sequence that needs to be written into a set of registers in the system configuration module, to allow write ability to the rest of registers in the system configuration module.
- Several registers in the module are only accessible when the CPU requesting read/write access is in privileged mode.

10.2 Protection

[Table 10-1](#) provides the list of registers in the SYSCFG module; it also indicates whether a particular register can be accessed only when the CPU is in privileged mode. See [Section 10.5](#) for a description of these registers.

Table 10-1. System Configuration (SYSCFG) Module Register Access

Offset	Acronym	Register Description	Access
0h	REVID	Revision Identification Register	—
8h-14h	DIEIDR0-DIEIDR3	Die Identification 0-3 Registers	—
18h	DEVIDR0	Device Identification Register 0	—
20h	BOOTCFG	Boot Configuration Register	Privileged mode
38h	KICK0R	Kick 0 Register	Privileged mode
3Ch	KICK1R	Kick 1 Register	Privileged mode
44h	HOST1CFG	Host 1 Configuration Register	—
E0h	IRAWSTAT	Interrupt Raw Status/Set Register	Privileged mode
E4h	IENSTAT	Interrupt Enable Status/Clear Register	Privileged mode
E8h	IENSET	Interrupt Enable Register	Privileged mode
ECh	IENCLR	Interrupt Enable Clear Register	Privileged mode
F0h	EOI	End of Interrupt Register	Privileged mode
F4h	FLTADDRR	Fault Address Register	Privileged mode
F8h	FLTSTAT	Fault Status Register	—
110h-118h	MSTPRI0-MSTPRI2	Master Priority 0-2 Registers	Privileged mode
120h-16Ch	PINMUX0-PINMUX19	Pin Multiplexing Control 0-19 Registers	Privileged mode
170h	SUSPSRC	Suspend Source Register	Privileged mode
174h	CHIPSIG	Chip Signal Register	—
178h	CHIPSIG_CLR	Chip Signal Clear Register	—
17Ch-18Ch	CFGCHIP0-CFGCHIP4	Chip Configuration 0-4 Registers	Privileged mode

10.2.1 Requirements to Access SYSCFG Registers

As mentioned previously, the SYSCFG module controls several global operations of the device; therefore, it has protection mechanism that prevents spurious and illegal accesses to the registers in its memory map. The protection mechanism enables accesses to these registers only if certain conditions are met. The protection mechanisms that are present in the module are described in the following sections.

10.2.1.1 Privilege Mode Protection

The CPU supports two privilege levels: Supervisor and User. Several registers in the SYSCFG memory-map can only be accessed when the accessing host (CPU or master peripheral) is operating in privileged mode, that is, in Supervisor mode. The registers that can only be accessed in privileged mode are listed in [Section 10.5](#). See the *TMS320C674x DSP CPU and Instruction Set Reference Guide (SPRUFE8)* for details on privilege levels.

10.2.1.2 Kicker Mechanism Protection

NOTE: The Kick 0 and Kick 1 registers can only be accessed in privileged mode (the host needs to be in Supervisor mode). Any number of accesses may be performed to the SYSCFG module, while the module is unlocked.

The SYSCFG module remains unlocked after the unlock sequence, until locked again. Locking the module is accomplished by writing any value other than the key values to either KICK0 or KICK1.

To access any registers in the SYSCFG module, it is required to follow a special sequence of writes to the Kick registers (Kick0 and Kick1) with correct key values. Writing the correct key value to the kick registers unlocks the registers in the SYSCFG memory-map. In order to access the SYSCFG registers, the following unlock sequence needs to be executed in software:

1. Write the key value of 83E7 0B13h to Kick 0 register.
2. Write the key value of 95A4 F1E0h to Kick 1 register.

After steps 1 and 2, the SYSCFG module registers are accessible and can be configured as per the application requirements.

10.3 Master Priority Control

The on-chip peripherals/modules are essentially divided into two broad categories, masters and slaves. The master peripherals are typically capable of initiating their own read/write data access requests, this includes the DSP, EDMA3 transfer controllers, and peripherals that do not rely on the CPU or EDMA3 for initiating the data transfer to/from them. In order to determine allowed connection between masters and slave, each master request source must have a unique master ID (mstid) associated with it. The master ID is shown in [Table 10-2](#). See the device-specific data manual to determine the masters present on your device.

Each switched central resource (SCR) performs prioritization based on priority level of the master that sends the read/write requests. For all peripherals/ports classified as masters on the device, the priority is programmed in the master priority registers (MSTPRI0-3) in the SYSCFG modules. The default priority levels for each bus master is shown in [Table 10-3](#). Application software is expected to modify these values to obtain the desired performance.

Table 10-2. Master IDs

Master ID	Peripheral
0-1	Reserved
2	DSP MDMA
3	DSP CFG
4-7	Reserved
8	PRU0
9	PRU1
10	TPCC0
11-15	Reserved
16	TPTC0 - read
17	TPTC0 - write
18	TPTC1 - read
19	TPTC1 - write
20-33	Reserved
34	USB2.0 CFG
35	USB2.0 DMA
36	Reserved
37	HPI ⁽¹⁾
38-63	Reserved
64	EMAC
65	USB1.1 ⁽¹⁾
66-95	Reserved
96	LCDC ⁽¹⁾
97-255	Reserved

⁽¹⁾ This peripheral is not supported on the C6745 DSP.

Table 10-3. Default Master Priority

Master	Default Priority ⁽¹⁾	Master Priority Register
PRU0	0	MSTPRI1
PRU1	0	MSTPRI1
EDMA3TC0 ⁽²⁾	0	MSTPRI1
EDMA3TC1	0	MSTPRI1
DSP MDMA ⁽³⁾	2	MSTPRI0
DSP CFG ⁽⁴⁾	2	MSTPRI0
EMAC	4	MSTPRI2
USB2.0 CFG	4	MSTPRI2
USB2.0 DMA	4	MSTPRI2
USB1.1 ⁽⁵⁾	4	MSTPRI2
LCDC ^{(5) (6)}	5	MSTPRI2
HPI ⁽⁵⁾	6	MSTPRI2

⁽¹⁾ The default priority settings might not be optimal for all applications. The master priority should be changed from default based on application specific requirement, in order to get optimal performance and prioritization for masters moving data that is real time sensitive.

⁽²⁾ The priority for EDMA3TC0 and EDMA3TC1 is configurable through fields in MSTPRI1, not the EDMA3CC QUEPRI register.

⁽³⁾ The priority for DSP MDMA and DSP CFG is controlled by fields in MSTPRI0 and not DSP.MDMAARBE.PRI (DSP Bandwidth manager module).

⁽⁴⁾ The priority for DSP MDMA and DSP CFG is controlled by fields in MSTPRI0 and not DSP.MDMAARBE.PRI (DSP Bandwidth manager module).

⁽⁵⁾ This peripheral is not supported on the C6745 DSP.

⁽⁶⁾ LCDC traffic is typically real-time sensitive, therefore, the default priority of 5, which is lower as compared to the default priority of several masters, is not recommended. You should reconfigure LCDC priority to the highest or equal to other high-priority masters in an application to ensure that throughput/latency requirements for LCDC are met.

10.4 Host-DSP Communication Interrupts

The SYSCFG module also has a set of registers to facilitate host-to-processor communication. This is generally used to allow an external host and the DSP to coordinate.

Either of the processors can set specific bits in this SYSCFG register, which in turn can interrupt the other processor, if the interrupts have been appropriately enabled in the processor's interrupt controller.

10.5 SYSCFG Registers

Table 10-4 lists the memory-mapped registers for the system configuration module (SYSCFG).

Table 10-4. System Configuration Module (SYSCFG) Registers

Address	Acronym	Register Description	Section
01C1 4000h	REVID	Revision Identification Register	Section 10.5.1
01C1 4008h	DIEIDR0 ⁽¹⁾	Die Identification Register 0	—
01C1 400Ch	DIEIDR1 ⁽¹⁾	Die Identification Register 1	—
01C1 4010h	DIEIDR2 ⁽¹⁾	Die Identification Register 2	—
01C1 4014h	DIEIDR3 ⁽¹⁾	Die Identification Register 3	—
01C1 4018h	DEVIDR0	Device Identification Register 0	Section 10.5.2
01C1 4020h	BOOTCFG	Boot Configuration Register	Section 10.5.3
01C1 4038h	KICK0R	Kick 0 Register	Section 10.5.4.1
01C1 403Ch	KICK1R	Kick 1 Register	Section 10.5.4.2
01C1 4044h	HOST1CFG	Host 1 Configuration Register	Section 10.5.5
01C1 40E0h	IRAWSTAT	Interrupt Raw Status/Set Register	Section 10.5.6.1
01C1 40E4h	IENSTAT	Interrupt Enable Status/Clear Register	Section 10.5.6.2
01C1 40E8h	IENSET	Interrupt Enable Register	Section 10.5.6.3
01C1 40ECh	IENCLR	Interrupt Enable Clear Register	Section 10.5.6.4
01C1 40F0h	EOI	End of Interrupt Register	Section 10.5.6.5
01C1 40F4h	FLTADDRR	Fault Address Register	Section 10.5.7.1
01C1 40F8h	FLTSTAT	Fault Status Register	Section 10.5.7.2
01C1 4110h	MSTPRI0	Master Priority 0 Register	Section 10.5.8.1
01C1 4114h	MSTPRI1	Master Priority 1 Register	Section 10.5.8.2
01C1 4118h	MSTPRI2	Master Priority 2 Register	Section 10.5.8.3
01C1 4120h	PINMUX0	Pin Multiplexing Control 0 Register	Section 10.5.9.1
01C1 4124h	PINMUX1	Pin Multiplexing Control 1 Register	Section 10.5.9.2
01C1 4128h	PINMUX2	Pin Multiplexing Control 2 Register	Section 10.5.9.3
01C1 412Ch	PINMUX3	Pin Multiplexing Control 3 Register	Section 10.5.9.4
01C1 4130h	PINMUX4	Pin Multiplexing Control 4 Register	Section 10.5.9.5
01C1 4134h	PINMUX5	Pin Multiplexing Control 5 Register	Section 10.5.9.6
01C1 4138h	PINMUX6	Pin Multiplexing Control 6 Register	Section 10.5.9.7
01C1 413Ch	PINMUX7	Pin Multiplexing Control 7 Register	Section 10.5.9.8
01C1 4140h	PINMUX8	Pin Multiplexing Control 8 Register	Section 10.5.9.9
01C1 4144h	PINMUX9	Pin Multiplexing Control 9 Register	Section 10.5.9.10
01C1 4148h	PINMUX10	Pin Multiplexing Control 10 Register	Section 10.5.9.11
01C1 414Ch	PINMUX11	Pin Multiplexing Control 11 Register	Section 10.5.9.12
01C1 4150h	PINMUX12	Pin Multiplexing Control 12 Register	Section 10.5.9.13
01C1 4154h	PINMUX13	Pin Multiplexing Control 13 Register	Section 10.5.9.14
01C1 4158h	PINMUX14	Pin Multiplexing Control 14 Register	Section 10.5.9.15
01C1 415Ch	PINMUX15	Pin Multiplexing Control 15 Register	Section 10.5.9.16
01C1 4160h	PINMUX16	Pin Multiplexing Control 16 Register	Section 10.5.9.17
01C1 4164h	PINMUX17	Pin Multiplexing Control 17 Register	Section 10.5.9.18
01C1 4168h	PINMUX18	Pin Multiplexing Control 18 Register	Section 10.5.9.19
01C1 416Ch	PINMUX19	Pin Multiplexing Control 19 Register	Section 10.5.9.20
01C1 4170h	SUSPSRC	Suspend Source Register	Section 10.5.10
01C1 4174h	CHIPSIG	Chip Signal Register	Section 10.5.11
01C1 4178h	CHIPSIG_CLR	Chip Signal Clear Register	Section 10.5.12

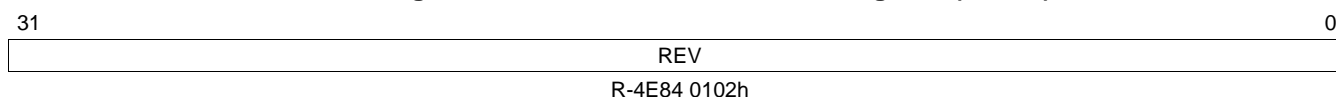
⁽¹⁾ This register is for internal-use only.

Table 10-4. System Configuration Module (SYSCFG) Registers (continued)

Address	Acronym	Register Description	Section
01C1 417Ch	CFGCHIP0	Chip Configuration 0 Register	Section 10.5.13
01C1 4180h	CFGCHIP1	Chip Configuration 1 Register	Section 10.5.14
01C1 4184h	CFGCHIP2	Chip Configuration 2 Register	Section 10.5.15
01C1 4188h	CFGCHIP3	Chip Configuration 3 Register	Section 10.5.16
01C1 418Ch	CFGCHIP4	Chip Configuration 4 Register	Section 10.5.17

10.5.1 Revision Identification Register (REVID)

The revision identification register (REVID) provides the revision information for the SYSCFG module. The REVID is shown in [Figure 10-1](#) and described in [Table 10-5](#).

Figure 10-1. Revision Identification Register (REVID)


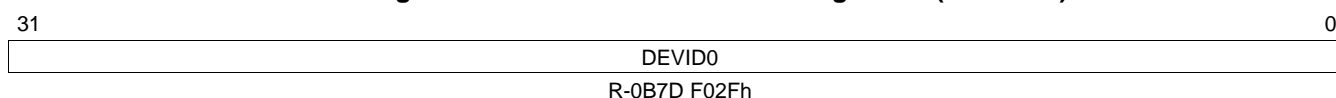
LEGEND: R = Read only; -n = value after reset

Table 10-5. Revision Identification Register (REVID) Field Descriptions

Bit	Field	Value	Description
31-0	REV	4E84 0102h	Revision ID. Revision information for the SYSCFG module.

10.5.2 Device Identification Register 0 (DEVIDR0)

The device identification register 0 (DEVIDR0) contains a software readable version of the JTAG ID device. Software can use this register to determine the version of the device on which it is executing. The DEVIDR0 is shown in [Figure 10-2](#) and described in [Table 10-6](#).

Figure 10-2. Device Identification Register 0 (DEVIDR0)


LEGEND: R = Read only; -n = value after reset

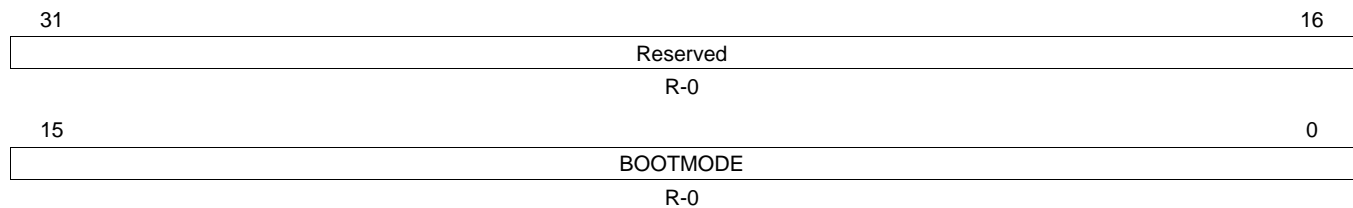
Table 10-6. Device Identification Register 0 (DEVIDR0) Field Descriptions

Bit	Field	Value	Description
31-0	DEVID0	R-0B7D F02Fh	Device identification.

10.5.3 Boot Configuration Register (BOOTCFG)

The device boot and configuration settings are latched at device reset, and captured in the boot configuration register (BOOTCFG). See the device-specific data manual and [Chapter 11](#) for details on boot and configuration settings. The BOOTCFG is shown in [Figure 10-3](#) and described in [Table 10-7](#).

Figure 10-3. Boot Configuration Register (BOOTCFG)



LEGEND: R = Read only; -n = value after reset

Table 10-7. Boot Configuration Register (BOOTCFG) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	BOOTMODE	0-FFFFh	Boot Mode. This reflects the state of the boot mode pins.

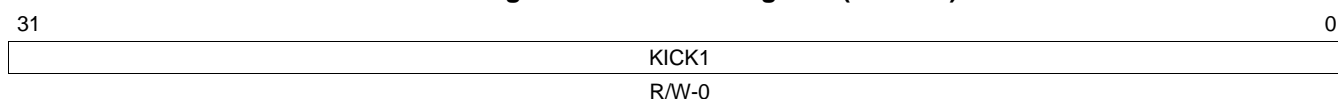
10.5.4 Kick Registers (KICK0R-KICK1R)

The SYSCFG module has a protection mechanism to prevent any spurious writes from changing any of the modules memory-mapped registers. At power-on reset, none of the SYSCFG module registers are writeable (they are readable). To allow writing to the registers in the module, it is required to “unlock” the registers by writing to two memory-mapped registers in the SYSCFG module, Kick0 and Kick1, with exact data values. Once these values are written, then all the registers in the SYSCFG module that are writeable can be written to. See [Section 10.2.1.2](#) for the exact key values and sequence of steps. Writing any other data value to either of these kick registers will cause the memory mapped registers to be “locked” again and block out any write accesses to registers in the SYSCFG module.

10.5.4.1 Kick 0 Register (KICK0R)

The KICK0R is shown in [Figure 10-4](#) and described in [Table 10-8](#).

Figure 10-4. Kick 0 Register (KICK0R)



LEGEND: R/W = Read/Write; -n = value after reset

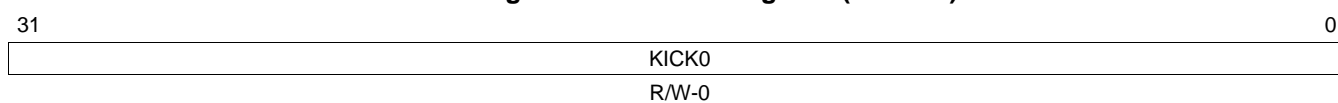
Table 10-8. Kick 0 Register (KICK0R) Field Descriptions

Bit	Field	Value	Description
31-0	KICK0	0-FFFF FFFFh	KICK0R allows writing to unlock the kick0 data. The written data must be 83E7 0B13h to unlock this register. It must be written before writing to the kick1 register. Writing any other value will lock the other MMRs.

10.5.4.2 Kick 1 Register (KICK1R)

The KICK1R is shown in [Figure 10-5](#) and described in [Table 10-9](#).

Figure 10-5. Kick 1 Register (KICK1R)



LEGEND: R/W = Read/Write; -n = value after reset

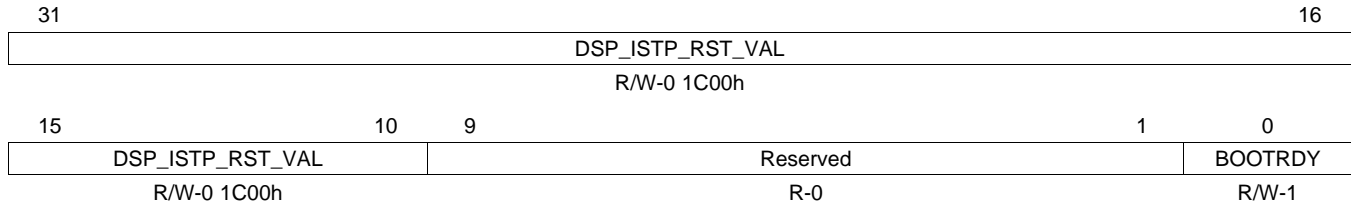
Table 10-9. Kick 1 Register (KICK1R) Field Descriptions

Bit	Field	Value	Description
31-0	KICK1	0-FFFF FFFFh	KICK1R allows writing to unlock the kick1 data and the kicker mechanism to write to other MMRs. The written data must be 95A4 F1E0h to unlock this register. KICK0R must be written before writing to the kick1 register. Writing any other value will lock the other MMRs.

10.5.5 Host 1 Configuration Register (HOST1CFG)

The host 1 configuration register (HOST1CFG) provides information on the DSP boot address value at power-on reset. The boot address defaults to 0070 0000h (DSP ROM) on power-up. The address field is read/writeable after reset and can be modified to allow execution from an alternate location after a module level or local reset on the DSP. The HOST1CFG is shown in [Figure 10-6](#) and described in [Table 10-10](#).

Figure 10-6. Host 1 Configuration Register (HOST1CFG)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 10-10. Host 1 Configuration Register (HOST1CFG) Field Descriptions

Bit	Field	Value	Description
31-10	DSP_ISTP_RST_VAL	0-3F FFFFh	DSP boot address vector.
9-1	Reserved	0	Reserved
0	BOOTRDY	0 1	DSP boot ready bit allowing DSP to boot. DSP held in reset mode. DSP released from wait in reset mode.

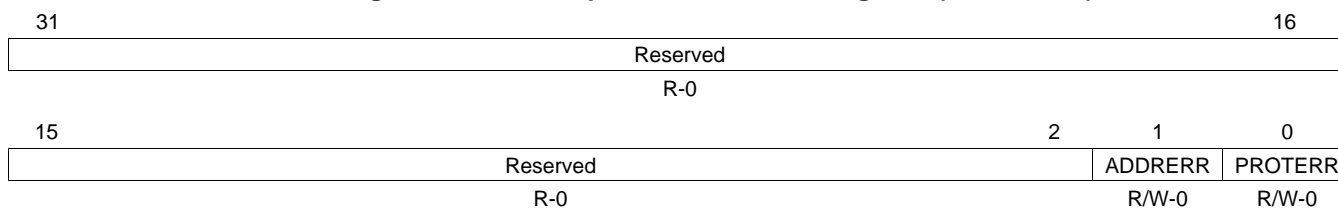
10.5.6 Interrupt Registers

The interrupt registers are a set of registers that provide control for the address and protection violation error interrupt generated by the SYSCFG module when there is an address or protection violation to the module's memory-mapped register address space. This includes enable control, interrupt set and clear control, and end of interrupt (EOI) control.

10.5.6.1 Interrupt Raw Status/Set Register (IRAWSTAT)

The interrupt raw status/set register (IRAWSTAT) shows the interrupt status before enabling the interrupt and allows setting of the interrupt status. The IRAWSTAT is shown in [Figure 10-7](#) and described in [Table 10-11](#).

Figure 10-7. Interrupt Raw Status/Set Register (IRAWSTAT)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

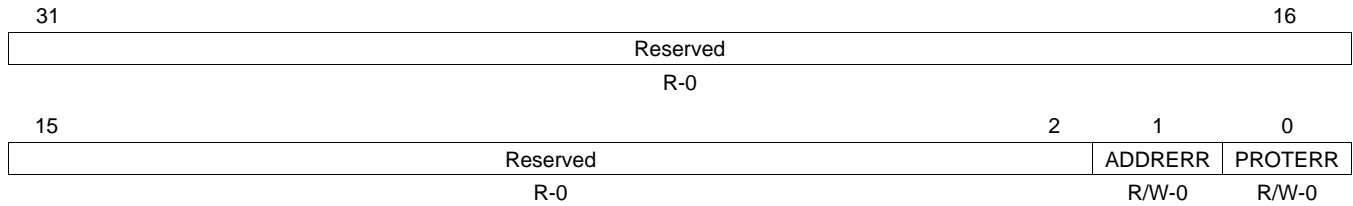
Table 10-11. Interrupt Raw Status/Set Register (IRAWSTAT) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. Always read 0.
1	ADDRERR	0 1	<p>Addressing violation error. Reading this bit field reflects the raw status of the interrupt before enabling.</p> <p>0 Indicates the interrupt is not set. Writing 0 has no effect.</p> <p>1 Indicates the interrupt is set. Writing 1 sets the status.</p>
0	PROTERR	0 1	<p>Protection violation error. Reading this bit field reflects the raw status of the interrupt before enabling.</p> <p>0 Indicates the interrupt is not set. Writing 0 has no effect.</p> <p>1 Indicates the interrupt is set. Writing 1 sets the status.</p>

10.5.6.2 Interrupt Enable Status/Clear Register (IENSTAT)

The interrupt enable status/clear register (IENSTAT) shows the status of enabled interrupt and allows clearing of the interrupt status. The IENSTAT is shown in [Figure 10-8](#) and described in [Table 10-12](#).

Figure 10-8. Interrupt Enable Status/Clear Register (IENSTAT)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

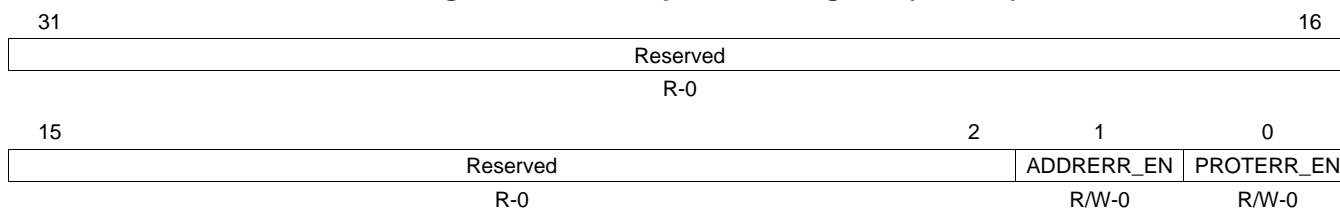
Table 10-12. Interrupt Enable Status/Clear Register (IENSTAT) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. Always read 0.
1	ADDRERR	0	Addressing violation error. Reading this bit field reflects the interrupt enabled status. Indicates the interrupt is not set. Writing 0 has no effect.
		1	Indicates the interrupt is set. Writing 1 clears the status.
0	PROTERR	0	Protection violation error. Reading this bit field reflects the interrupt enabled status. Indicates the interrupt is not set. Writing 0 has no effect.
		1	Indicates the interrupt is set. Writing 1 clears the status.

10.5.6.3 Interrupt Enable Register (IENSET)

The interrupt enable register (IENSET) allows setting/enabling the interrupt for address and/or protection violation condition. It also shows the value of the register (whether or not interrupt is enabled). The IENSET is shown in [Figure 10-9](#) and described in [Table 10-13](#).

Figure 10-9. Interrupt Enable Register (IENSET)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

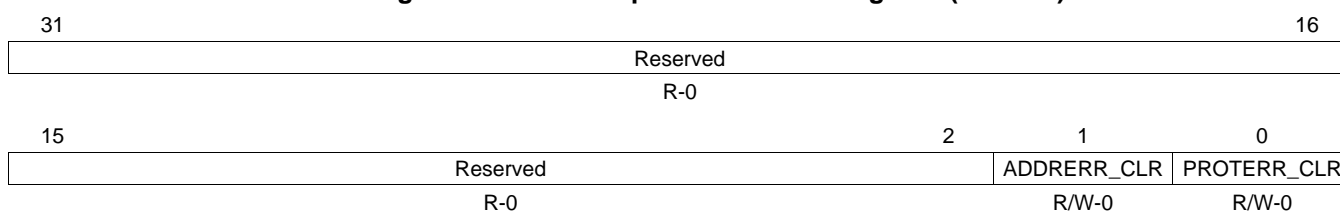
Table 10-13. Interrupt Enable Register (IENSET) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. Always read 0.
1	ADDRERR_EN	0	Addressing violation error. Writing a 0 has not effect.
		1	Writing a 1 enables this interrupt.
0	PROTERR_EN	0	Protection violation error. Writing a 0 has not effect.
		1	Writing a 1 enables this interrupt.

10.5.6.4 Interrupt Enable Clear Register (IENCLR)

The interrupt enable clear register (IENCLR) allows clearing/disable the interrupt for address and/or protection violation condition. It also shows the value of the interrupt enable register (IENSET). The IENCLR is shown in [Figure 10-10](#) and described in [Table 10-14](#).

Figure 10-10. Interrupt Enable Clear Register (IENCLR)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

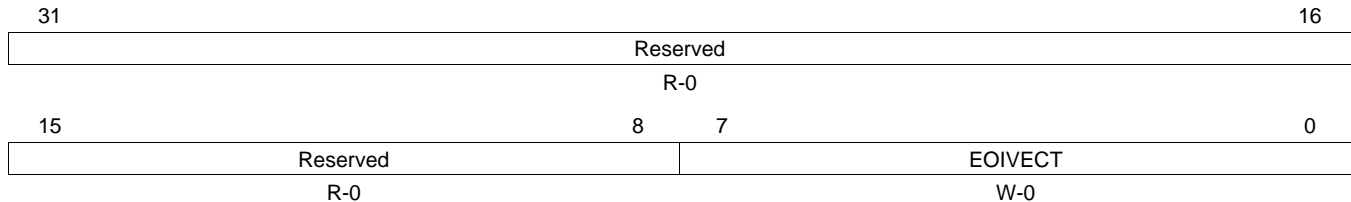
Table 10-14. Interrupt Enable Clear Register (IENCLR) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. Always read 0.
1	ADDRERR_CLR	0	Addressing violation error. Writing a 0 has not effect.
		1	Writing a 1 clears/disables this interrupt.
0	PROTERR_CLR	0	Protection violation error. Writing a 0 has not effect.
		1	Writing a 1 clears/disables this interrupt.

10.5.6.5 End of Interrupt Register (EOI)

The end of interrupt register (EOI) is used in software to indicate completion of the interrupt servicing of the SYSCFG interrupt (for address/protection violation). You should write a value of 0 to the EOI register bit 0 after the software has processed the SYSCFG interrupt, this acts as an acknowledgement of completion of the SYSCFG interrupt so that the module can reliably generate the subsequent interrupts. The EOI is shown in [Figure 10-11](#) and described in [Table 10-15](#).

Figure 10-11. End of Interrupt Register (EOI)



LEGEND: R = Read only; W = Write only; -n = value after reset

Table 10-15. End of Interrupt Register (EOI) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always read 0.
7-0	EOI ECT	0-FFh	EOI vector value. Write the interrupt distribution value of the chip.

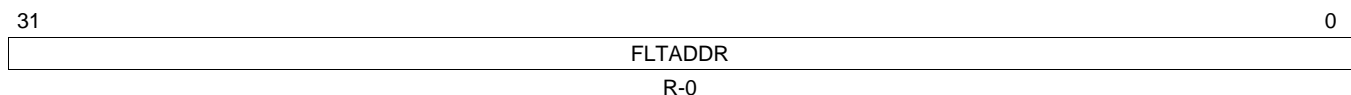
10.5.7 Fault Registers

The fault registers are a group of registers responsible for capturing the details on the faulty (address/protection violation errors) accesses, such as address and type of error.

10.5.7.1 Fault Address Register (FLTADDR)

The fault address register (FLTADDR) captures the address of the first transfer that causes the address or memory violation error. The FLTADDR is shown in [Figure 10-12](#) and described in [Table 10-16](#).

Figure 10-12. Fault Address Register (FLTADDR)



LEGEND: R = Read only; -n = value after reset

Table 10-16. Fault Address Register (FLTADDR) Field Descriptions

Bit	Field	Value	Description
31-0	FLTADDR	0-FFFF FFFFh	Fault address for the first fault transfer.

10.5.7.2 Fault Status Register (FLTSTAT)

The fault status register (FLTSTAT) holds/captures additional attributes and status of the first erroneous transaction. This includes things like the master id for the master that caused the address/memory violation error, details on whether it is a user or supervisor level read/write or execute fault. The FLTSTAT is shown in [Figure 10-13](#) and described in [Table 10-17](#).

Figure 10-13. Fault Status Register (FLTSTAT)



LEGEND: R = Read only; -n = value after reset

Table 10-17. Fault Status Register (FLTSTAT) Field Descriptions

Bit	Field	Value	Description
31-24	ID	0-FFh	Transfer ID of the first fault transfer.
23-16	MSTID	0-FFh	Master ID of the first fault transfer.
15-13	Reserved	0	Reserved. Always read 0
12-9	PRIVID	0-Fh	Privilege ID of the first fault transfer.
8-6	Reserved	0	Reserved. Always read 0
5-0	TYPE		Fault type of first fault transfer.
		0	No transfer fault
		1h	User execute fault
		2h	User write fault
		3h	<i>Reserved</i>
		4h	User read fault
		5h-7h	<i>Reserved</i>
		8h	Supervisor execute fault
		9h-Fh	<i>Reserved</i>
		10h	Supervisor write fault
		11h-1Fh	<i>Reserved</i>
		20h	Supervisor read fault
		21h-3Fh	<i>Reserved</i>

10.5.8 Master Priority Registers (MSTPRI0-MSTPRI2)

10.5.8.1 Master Priority 0 Register (MSTPRI0)

The master priority 0 register (MSTPRI0) is shown in [Figure 10-14](#) and described in [Table 10-18](#).

Figure 10-14. Master Priority 0 Register (MSTPRI0)

31	30	28	27	26	24	23	22	20	19	18	16
Rsvd	Reserved		Rsvd	Reserved		Rsvd	Reserved		Rsvd	Reserved	
R/W-0	R/W-4h		R/W-0	R/W-4h		R/W-0	R/W-4h		R/W-0	R/W-4h	
15	14	12	11	10	8	7	6	4	3	2	0
Rsvd	DSP_CFG		Rsvd	DSP_MDMA		Rsvd	Reserved		Rsvd	Reserved	
R/W-0	R/W-2h		R-0	R/W-2h		R-0	R/W-2h		R-0	R/W-2h	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 10-18. Master Priority 0 Register (MSTPRI0) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. Write the default value when modifying this register.
30-28	Reserved	4h	Reserved. Write the default value when modifying this register.
27	Reserved	0	Reserved. Write the default value when modifying this register.
26-24	Reserved	4h	Reserved. Write the default value when modifying this register.
23	Reserved	0	Reserved. Write the default value when modifying this register.
22-20	Reserved	4h	Reserved. Write the default value when modifying this register.
19	Reserved	0	Reserved. Write the default value when modifying this register.
18-16	Reserved	4h	Reserved. Write the default value when modifying this register.
15	Reserved	0	Reserved. Write the default value when modifying this register.
14-12	DSP_CFG	0-7h	DSP CFG port priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).
11	Reserved	0	Reserved. Always read as 0.
10-8	DSP_MDMA	0-7h	DSP DMA port priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).
7	Reserved	0	Reserved. Always read as 0.
6-4	Reserved	2h	Reserved. Write the default value when modifying this register.
3	Reserved	0	Reserved. Always read as 0.
2-0	Reserved	2h	Reserved. Write the default value when modifying this register.

10.5.8.2 Master Priority 1 Register (MSTPRI1)

The master priority 1 register (MSTPRI1) is shown in [Figure 10-15](#) and described in [Table 10-19](#).

Figure 10-15. Master Priority 1 Register (MSTPRI1)

31	30	28	27	26	24	23	22	20	19	18	16
Rsvd	Reserved		Rsvd	Reserved		Rsvd	Reserved		Rsvd	Reserved	
R/W-0	R/W-4h		R/W-0	R/W-4h		R/W-0	R/W-4h		R/W-0	R/W-4h	
15	14	12	11	10	8	7	6	4	3	2	0
Rsvd	EDMATC1		Rsvd	EDMATC0		Rsvd	PRU1		Rsvd	PRU0	
R/W-0	R/W-0		R-0	R/W-0		R-0	R/W-0		R-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 10-19. Master Priority 1 Register (MSTPRI1) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. Write the default value when modifying this register.
30-28	Reserved	4h	Reserved. Write the default value when modifying this register.
27	Reserved	0	Reserved. Write the default value when modifying this register.
26-24	Reserved	4h	Reserved. Write the default value when modifying this register.
23	Reserved	0	Reserved. Write the default value when modifying this register.
22-20	Reserved	4h	Reserved. Write the default value when modifying this register.
19	Reserved	0	Reserved. Write the default value when modifying this register.
18-16	Reserved	4h	Reserved. Write the default value when modifying this register.
15	Reserved	0	Reserved. Write the default value when modifying this register.
14-12	EDMATC1	0-7h	EDMA3TC1 priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).
11	Reserved	0	Reserved. Always read as 0.
10-8	EDMATC0	0-7h	EDMA3TC0 priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).
7	Reserved	0	Reserved. Always read as 0.
6-4	PRU1	0-7h	PRU1 priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).
3	Reserved	0	Reserved. Always read as 0.
2-0	PRU0	0-7h	PRU0 priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).

10.5.8.3 Master Priority 2 Register (MSTPRI2)

The master priority 2 register (MSTPRI2) is shown in [Figure 10-16](#) and described in [Table 10-20](#).

Figure 10-16. Master Priority 2 Register (MSTPRI2)

31	30	28	27	26	24	23	22	20	19	18	16
Rsvd	LCDC ⁽¹⁾		Rsvd	USB1 ⁽¹⁾		Rsvd	UHPI ⁽¹⁾		Rsvd	Reserved	
R/W-0	R/W-5h		R/W-0	R/W-4h		R/W-0	R/W-6h		R/W-0	R/W-0	
15	14	12	11	10	8	7	6	4	3	2	0
Rsvd	USB0CDMA		Rsvd	USB0CFG		Rsvd	Reserved		Rsvd	EMAC	
R/W-0	R/W-4h		R/W-0	R/W-4h		R/W-0	R/W-0		R/W-0	R/W-4h	

LEGEND: R/W = Read/Write; -n = value after reset

⁽¹⁾ This bit is not supported and is Reserved on the C6745 DSP.

Table 10-20. Master Priority 2 Register (MSTPRI2) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. Write the default value when modifying this register.
30-28	LCDC	0-7h	LCDC priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest). This peripheral is not supported on the C6745 DSP.
27	Reserved	0	Reserved. Write the default value when modifying this register.
26-24	USB1	0-7h	USB1 (USB1.1) priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest). This peripheral is not supported on the C6745 DSP.
23	Reserved	0	Reserved. Write the default value when modifying this register.
22-20	UHPI	0-7h	HPI priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest). This peripheral is not supported on the C6745 DSP.
19	Reserved	0	Reserved. Write the default value when modifying this register.
18-16	Reserved	0	Reserved. Write the default value to all bits when modifying this register.
15	Reserved	0	Reserved. Write the default value when modifying this register.
14-12	USB0CDMA	0-7h	USB0 (USB2.0) CDMA priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).
11	Reserved	0	Reserved. Write the default value when modifying this register.
10-8	USB0CFG	0-7h	USB0 (USB2.0) CFG priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).
7	Reserved	0	Reserved. Write the default value to all bits when modifying this register.
6-4	Reserved	0	Reserved. Write the default value to all bits when modifying this register.
3	Reserved	0	Reserved. Write the default value when modifying this register.
2-0	EMAC	0-7h	EMAC priority. Bit = 0 = priority 0 (highest); bit = 7h = priority 7 (lowest).

10.5.9 Pin Multiplexing Control Registers (PINMUX0-PINMUX19)

NOTE: The C6745 DSP does not support all of the pins documented in the following subsections. See the *TMS320C6745/C6747 DSP Data Manual* ([SPRS377](#)) for the peripherals supported.

Extensive use of pin multiplexing is used to accommodate the large number of peripheral functions in the smallest possible package. On the device, pin multiplexing can be controlled on a pin by pin basis. This is done by the pin multiplexing registers (PINMUX0-PINMUX19). Each pin that is multiplexed with several different functions has a corresponding 4-bit field in PINMUX n . Pin multiplexing selects which of several peripheral pin functions control the pins IO buffer output data and output enable values only. Note that the input from each pin is always routed to all of the peripherals that share the pin; the PINMUX registers have no effect on input from a pin. Hardware does not attempt to ensure that the proper pin multiplexing is selected for the peripherals or that interface mode is being used. Detailed information about the pin multiplexing and control is covered in the device-specific data manual. Access to the pin multiplexing utility is available in *OMAP-L137, TMS320C6745/C6747 Pin Multiplexing Utility Application Report* ([SPRAB06](#)).

10.5.9.1 Pin Multiplexing Control 0 Register (PINMUX0)

Figure 10-17. Pin Multiplexing Control 0 Register (PINMUX0)

31	28	27	24	23	20	19	16
PINMUX0_31_28		PINMUX0_27_24		PINMUX0_23_20		PINMUX0_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX0_15_12		PINMUX0_11_8		PINMUX0_7_4		PINMUX0_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-21. Pin Multiplexing Control 0 Register (PINMUX0) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX0_31_28	K15	59	0 1h 2h-Fh	EMB_WE Control Pin is 3-stated. Selects Function EMB_WE <i>Reserved</i>
27-24	PINMUX0_27_24	A8	110	0 1h 2h-Fh	EMB_RAS Control Pin is 3-stated. Selects Function EMB_RAS <i>Reserved</i>
23-20	PINMUX0_23_20	L13	57	0 1h 2h-Fh	EMB_CAS Control Pin is 3-stated. Selects Function EMB_CAS <i>Reserved</i>
19-16	PINMUX0_19_16	D9	108	0 1h 2h-Fh	EMB_CS[0] Control Pin is 3-stated. Selects Function EMB_CS[0] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-21. Pin Multiplexing Control 0 Register (PINMUX0) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX0_15_12	C14	86	0 1h 2h 3h-Fh	EMB_CLK Control Pin is 3-stated. Selects Function EMB_CLK from EMIFB LPSC (CLK1) Selects Function EMB_CLK from PLL DIV4P5 or SYSCLK5 (CLK2) <i>Reserved</i>
11-8	PINMUX0_11_8	C13	88	0 1h 2h-Fh	EMB_SDCKE Control Pin is 3-stated. Selects Function EMB_SDCKE <i>Reserved</i>
7-4	PINMUX0_7_4	J5	—	0 1h 2h-7h 8h 9h-Fh	GP7[15]/EMU[0] Control Pin is 3-stated. Selects Function GP7[15] <i>Reserved</i> Selects Function EMU[0] <i>Reserved</i>
3-0	PINMUX0_3_0	K1	157	0 1h 2h-Fh	GP7[14] Control. GP7[14] is initially configured as a reserved function after reset and will not be in a predictable state. This signal will only be stable after the GPIO configuration for this pin has been completed. You should carefully consider the system implications of this pin being in an unknown state after reset. <i>Reserved</i> Selects Function GP7[14] <i>Reserved</i>

10.5.9.2 Pin Multiplexing Control 1 Register (PINMUX1)
Figure 10-18. Pin Multiplexing Control 1 Register (PINMUX1)

31	28	27	24	23	20	19	16
PINMUX1_31_28		PINMUX1_27_24		PINMUX1_23_20		PINMUX1_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX1_15_12		PINMUX1_11_8		PINMUX1_7_4		PINMUX1_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-22. Pin Multiplexing Control 1 Register (PINMUX1) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX1_31_28	C11	97	0 1h 2h-7h 8h 9h-Fh	EMB_A[5]/GP7[7] Control Pin is 3-stated. Selects Function EMB_A[5] <i>Reserved</i> Selects Function GP7[7] <i>Reserved</i>
27-24	PINMUX1_27_24	D11	98	0 1h 2h-7h 8h 9h-Fh	EMB_A[4]/GP7[6] Control Pin is 3-stated. Selects Function EMB_A[4] <i>Reserved</i> Selects Function GP7[6] <i>Reserved</i>
23-20	PINMUX1_23_20	A10	100	0 1h 2h-7h 8h 9h-Fh	EMB_A[3]/GP7[5] Control Pin is 3-stated. Selects Function EMB_A[3] <i>Reserved</i> Selects Function GP7[5] <i>Reserved</i>
19-16	PINMUX1_19_16	B10	101	0 1h 2h-7h 8h 9h-Fh	EMB_A[2]/GP7[4] Control Pin is 3-stated. Selects Function EMB_A[2] <i>Reserved</i> Selects Function GP7[4] <i>Reserved</i>
15-12	PINMUX1_15_12	C10	102	0 1h 2h-7h 8h 9h-Fh	EMB_A[1]/GP7[3] Control Pin is 3-stated. Selects Function EMB_A[1] <i>Reserved</i> Selects Function GP7[3] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-22. Pin Multiplexing Control 1 Register (PINMUX1) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
11-8	PINMUX1_11_8	D10	103	0 1h 2h-7h 8h 9h-Fh	EMB_A[0]/GP7[2] Control Pin is 3-stated. Selects Function EMB_A[0] <i>Reserved</i> Selects Function GP7[2] <i>Reserved</i>
7-4	PINMUX1_7_4	C9	107	0 1h 2h-7h 8h 9h-Fh	EMB_BA[0]/GP7[1] Control Pin is 3-stated. Selects Function EMB_BA[0] <i>Reserved</i> Selects Function GP7[1] <i>Reserved</i>
3-0	PINMUX1_3_0	B9	106	0 1h 2h-7h 8h 9h-Fh	EMB_BA[1]/GP7[0] Control Pin is 3-stated. Selects Function EMB_BA[1] <i>Reserved</i> Selects Function GP7[0] <i>Reserved</i>

10.5.9.3 Pin Multiplexing Control 2 Register (PINMUX2)
Figure 10-19. Pin Multiplexing Control 2 Register (PINMUX2)

31	28	27	24	23	20	19	16
PINMUX2_31_28		PINMUX2_27_24		PINMUX2_23_20		PINMUX2_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX2_15_12		PINMUX2_11_8		PINMUX2_7_4		PINMUX2_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-23. Pin Multiplexing Control 2 Register (PINMUX2) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX2_31_28	G14	—	0 1h 2h-Fh	EMB_D[31] Control Pin is 3-stated. Selects Function EMB_D[31] <i>Reserved</i>
27-24	PINMUX2_27_24	B15	89	0 1h 2h-7h 8h 9h-Fh	EMB_A[12]/GP3[13] Control Pin is 3-stated. Selects Function EMB_A[12] <i>Reserved</i> Selects Function GP3[13] <i>Reserved</i>
23-20	PINMUX2_23_20	B12	91	0 1h 2h-7h 8h 9h-Fh	EMB_A[11]/GP7[13] Control Pin is 3-stated. Selects Function EMB_A[11] <i>Reserved</i> Selects Function GP7[13] <i>Reserved</i>
19-16	PINMUX2_19_16	A9	105	0 1h 2h-7h 8h 9h-Fh	EMB_A[10]/GP7[12] Control Pin is 3-stated. Selects Function EMB_A[10] <i>Reserved</i> Selects Function GP7[12] <i>Reserved</i>
15-12	PINMUX2_15_12	C12	92	0 1h 2h-7h 8h 9h-Fh	EMB_A[9]/GP7[11] Control Pin is 3-stated. Selects Function EMB_A[9] <i>Reserved</i> Selects Function GP7[11] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-23. Pin Multiplexing Control 2 Register (PINMUX2) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
11-8	PINMUX2_11_8	D12	94	0 1h 2h-7h 8h 9h-Fh	EMB_A[8]/GP7[10] Control Pin is 3-stated. Selects Function EMB_A[8] <i>Reserved</i> Selects Function GP7[10] <i>Reserved</i>
7-4	PINMUX2_7_4	A11	95	0 1h 2h-7h 8h 9h-Fh	EMB_A[7]/GP7[9] Control Pin is 3-stated. Selects Function EMB_A[7] <i>Reserved</i> Selects Function GP7[9] <i>Reserved</i>
3-0	PINMUX2_3_0	B11	96	0 1h 2h-7h 8h 9h-Fh	EMB_A[6]/GP7[8] Control Pin is 3-stated. Selects Function EMB_A[6] <i>Reserved</i> Selects Function GP7[8] <i>Reserved</i>

10.5.9.4 Pin Multiplexing Control 3 Register (PINMUX3)

Figure 10-20. Pin Multiplexing Control 3 Register (PINMUX3)

31	28	27	24	23	20	19	16
PINMUX3_31_28		PINMUX3_27_24		PINMUX3_23_20		PINMUX3_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX3_15_12		PINMUX3_11_8		PINMUX3_7_4		PINMUX3_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-24. Pin Multiplexing Control 3 Register (PINMUX3) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX3_31_28	L15	—	0 1h 2h-Fh	EMB_D[23] Control Pin is 3-stated. Selects Function EMB_D[23] <i>Reserved</i>
27-24	PINMUX3_27_24	A13	—	0 1h 2h-Fh	EMB_D[24] Control Pin is 3-stated. Selects Function EMB_D[24] <i>Reserved</i>
23-20	PINMUX3_23_20	B14	—	0 1h 2h-Fh	EMB_D[25] Control Pin is 3-stated. Selects Function EMB_D[25] <i>Reserved</i>
19-16	PINMUX3_19_16	A14	—	0 1h 2h-Fh	EMB_D[26] Control Pin is 3-stated. Selects Function EMB_D[26] <i>Reserved</i>
15-12	PINMUX3_15_12	E14	—	0 1h 2h-Fh	EMB_D[27] Control Pin is 3-stated. Selects Function EMB_D[27] <i>Reserved</i>
11-8	PINMUX3_11_8	E15	—	0 1h 2h-Fh	EMB_D[28] Control Pin is 3-stated. Selects Function EMB_D[28] <i>Reserved</i>
7-4	PINMUX3_7_4	F14	—	0 1h 2h-Fh	EMB_D[29] Control Pin is 3-stated. Selects Function EMB_D[29] <i>Reserved</i>
3-0	PINMUX3_3_0	F15	—	0 1h 2h-Fh	EMB_D[30] Control Pin is 3-stated. Selects Function EMB_D[30] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

10.5.9.5 Pin Multiplexing Control 4 Register (PINMUX4)

Figure 10-21. Pin Multiplexing Control 4 Register (PINMUX4)

31	28	27	24	23	20	19	16
PINMUX4_31_28		PINMUX4_27_24		PINMUX4_23_20		PINMUX4_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX4_15_12		PINMUX4_11_8		PINMUX4_7_4		PINMUX4_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-25. Pin Multiplexing Control 4 Register (PINMUX4) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX4_31_28	A12	—	0 1h 2h-Fh	EMB_WE_DQM[3] Control Pin is 3-stated. Selects Function EMB_WE_DQM[3] <i>Reserved</i>
27-24	PINMUX4_27_24	G15	—	0 1h 2h-Fh	EMB_D[16] Control Pin is 3-stated. Selects Function EMB_D[16] <i>Reserved</i>
23-20	PINMUX4_23_20	H14	—	0 1h 2h-Fh	EMB_D[17] Control Pin is 3-stated. Selects Function EMB_D[17] <i>Reserved</i>
19-16	PINMUX4_19_16	H15	—	0 1h 2h-Fh	EMB_D[18] Control Pin is 3-stated. Selects Function EMB_D[18] <i>Reserved</i>
15-12	PINMUX4_15_12	J14	—	0 1h 2h-Fh	EMB_D[19] Control Pin is 3-stated. Selects Function EMB_D[19] <i>Reserved</i>
11-8	PINMUX4_11_8	K13	—	0 1h 2h-Fh	EMB_D[20] Control Pin is 3-stated. Selects Function EMB_D[20] <i>Reserved</i>
7-4	PINMUX4_7_4	K16	—	0 1h 2h-Fh	EMB_D[21] Control Pin is 3-stated. Selects Function EMB_D[21] <i>Reserved</i>
3-0	PINMUX4_3_0	L14	—	0 1h 2h-Fh	EMB_D[22] Control Pin is 3-stated. Selects Function EMB_D[22] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

10.5.9.6 Pin Multiplexing Control 5 Register (PINMUX5)

Figure 10-22. Pin Multiplexing Control 5 Register (PINMUX5)

31	28	27	24	23	20	19	16
PINMUX5_31_28		PINMUX5_27_24		PINMUX5_23_20		PINMUX5_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX5_15_12		PINMUX5_11_8		PINMUX5_7_4		PINMUX5_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-26. Pin Multiplexing Control 5 Register (PINMUX5) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX5_31_28	J15	63	0 1h 2h-7h 8h 9h-Fh	EMB_D[6]/GP6[6] Control Pin is 3-stated. Selects Function EMB_D[6] <i>Reserved</i> Selects Function GP6[6] <i>Reserved</i>
27-24	PINMUX5_27_24	J13	64	0 1h 2h-7h 8h 9h-Fh	EMB_D[5]/GP6[5] Control Pin is 3-stated. Selects Function EMB_D[5] <i>Reserved</i> Selects Function GP6[5] <i>Reserved</i>
23-20	PINMUX5_23_20	H16	66	0 1h 2h-7h 8h 9h-Fh	EMB_D[4]/GP6[4] Control Pin is 3-stated. Selects Function EMB_D[4] <i>Reserved</i> Selects Function GP6[4] <i>Reserved</i>
19-16	PINMUX5_19_16	H13	68	0 1h 2h-7h 8h 9h-Fh	EMB_D[3]/GP6[3] Control Pin is 3-stated. Selects Function EMB_D[3] <i>Reserved</i> Selects Function GP6[3] <i>Reserved</i>
15-12	PINMUX5_15_12	G16	70	0 1h 2h-7h 8h 9h-Fh	EMB_D[2]/GP6[2] Control Pin is 3-stated. Selects Function EMB_D[2] <i>Reserved</i> Selects Function GP6[2] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-26. Pin Multiplexing Control 5 Register (PINMUX5) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
11-8	PINMUX5_11_8	G13	72	0 1h 2h-7h 8h 9h-Fh	EMB_D[1]/GP6[1] Control Pin is 3-stated. Selects Function EMB_D[1] <i>Reserved</i> Selects Function GP6[1] <i>Reserved</i>
7-4	PINMUX5_7_4	F16	73	0 1h 2h-7h 8h 9h-Fh	EMB_D[0]/GP6[0] Control Pin is 3-stated. Selects Function EMB_D[0] <i>Reserved</i> Selects Function GP6[0] <i>Reserved</i>
3-0	PINMUX5_3_0	B13	—	0 1h 2h-Fh	EMB_WE_DQM[2] Control Pin is 3-stated. Selects Function <u>EMB_WE_DQM[2]</u> <i>Reserved</i>

10.5.9.7 Pin Multiplexing Control 6 Register (PINMUX6)
Figure 10-23. Pin Multiplexing Control 6 Register (PINMUX6)

31	28	27	24	23	20	19	16
PINMUX6_31_28		PINMUX6_27_24		PINMUX6_23_20		PINMUX6_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX6_15_12		PINMUX6_11_8		PINMUX6_7_4		PINMUX6_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-27. Pin Multiplexing Control 6 Register (PINMUX6) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX6_31_28	E16	76	0 1h 2h-7h 8h 9h-Fh	EMB_D[14]/GP6[14] Control Pin is 3-stated. Selects Function EMB_D[14] <i>Reserved</i> Selects Function GP6[14] <i>Reserved</i>
27-24	PINMUX6_27_24	E13	78	0 1h 2h-7h 8h 9h-Fh	EMB_D[13]/GP6[13] Control Pin is 3-stated. Selects Function EMB_D[13] <i>Reserved</i> Selects Function GP6[13] <i>Reserved</i>
23-20	PINMUX6_23_20	D16	79	0 1h 2h-7h 8h 9h-Fh	EMB_D[12]/GP6[12] Control Pin is 3-stated. Selects Function EMB_D[12] <i>Reserved</i> Selects Function GP6[12] <i>Reserved</i>
19-16	PINMUX6_19_16	D15	80	0 1h 2h-7h 8h 9h-Fh	EMB_D[11]/GP6[11] Control Pin is 3-stated. Selects Function EMB_D[11] <i>Reserved</i> Selects Function GP6[11] <i>Reserved</i>
15-12	PINMUX6_15_12	D14	82	0 1h 2h-7h 8h 9h-Fh	EMB_D[10]/GP6[10] Control Pin is 3-stated. Selects Function EMB_D[10] <i>Reserved</i> Selects Function GP6[10] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-27. Pin Multiplexing Control 6 Register (PINMUX6) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
11-8	PINMUX6_11_8	D13	83	0 1h 2h-7h 8h 9h-Fh	EMB_D[9]/GP6[9] Control Pin is 3-stated. Selects Function EMB_D[9] <i>Reserved</i> Selects Function GP6[9] <i>Reserved</i>
7-4	PINMUX6_7_4	C16	84	0 1h 2h-7h 8h 9h-Fh	EMB_D[8]/GP6[8] Control Pin is 3-stated. Selects Function EMB_D[8] <i>Reserved</i> Selects Function GP6[8] <i>Reserved</i>
3-0	PINMUX6_3_0	J16	62	0 1h 2h-7h 8h 9h-Fh	EMB_D[7]/GP6[7] Control Pin is 3-stated. Selects Function EMB_D[7] <i>Reserved</i> Selects Function GP6[7] <i>Reserved</i>

10.5.9.8 Pin Multiplexing Control 7 Register (PINMUX7)
Figure 10-24. Pin Multiplexing Control 7 Register (PINMUX7)

31	28	27	24	23	20	19	16
PINMUX7_31_28		PINMUX7_27_24		PINMUX7_23_20		PINMUX7_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX7_15_12		PINMUX7_11_8		PINMUX7_7_4		PINMUX7_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-28. Pin Multiplexing Control 7 Register (PINMUX7) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX7_31_28	N4	9	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	SPI0_SCS[0]/UART0_RTS/EQEP0B/GP5[4]/BOOT[4] Control Pin is 3-stated. Selects Function <u>SPI0_SCS[0]</u> Selects Function <u>UART0_RTS</u> <i>Reserved</i> Selects Function EQEP0B <i>Reserved</i> Selects Function GP5[4] <i>Reserved</i>
27-24	PINMUX7_27_24	R5	12	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	SPI0_ENA/UART0_CTS/EQEP0A/GP5[3]/BOOT[3] Control Pin is 3-stated. Selects Function <u>SPI0_ENA</u> Selects Function <u>UART0_CTS</u> <i>Reserved</i> Selects Function EQEP0A <i>Reserved</i> Selects Function GP5[3] <i>Reserved</i>
23-20	PINMUX7_23_20	T5	11	0 1h 2h 3h-7h 8h 9h-Fh	SPI0_CLK/EQEP1/GP5[2]/BOOT[2] Control Pin is 3-stated. Selects Function SPI0_CLK Selects Function EQEP1 <i>Reserved</i> Selects Function GP5[2] <i>Reserved</i>
19-16	PINMUX7_19_16	P6	18	0 1h 2h 3h-7h 8h 9h-Fh	SPI0_SIMO[0]/EQEP0S/GP5[1]/BOOT[1] Control Pin is 3-stated. Selects Function SPI0_SIMO[0] Selects Function EQEP0S <i>Reserved</i> Selects Function GP5[1] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-28. Pin Multiplexing Control 7 Register (PINMUX7) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX7_15_12	R6	17	0 1h 2h 3h-7h 8h 9h-Fh	SPI0_SOMI[0]/EQEP0/GP5[0]/BOOT[0] Control Pin is 3-stated. Selects Function SPI0_SOMI[0] Selects Function EQEP0 <i>Reserved</i> Selects Function GP5[0] <i>Reserved</i>
11-8	PINMUX7_11_8	K14	60	0 1h 2h-7h 8h 9h-Fh	EMB_WE_DQM[0]/GP5[15] Control Pin is 3-stated. Selects Function <u>EMB_WE_DQM[0]</u> <i>Reserved</i> Selects Function GP5[15] <i>Reserved</i>
7-4	PINMUX7_7_4	C15	85	0 1h 2h-7h 8h 9h-Fh	EMB_WE_DQM[1]/GP5[14] Control Pin is 3-stated. Selects Function <u>EMB_WE_DQM[1]</u> <i>Reserved</i> Selects Function GP5[14] <i>Reserved</i>
3-0	PINMUX7_3_0	F13	74	0 1h 2h-7h 8h 9h-Fh	EMB_D[15]/GP6[15] Control Pin is 3-stated. Selects Function EMB_D[15] <i>Reserved</i> Selects Function GP6[15] <i>Reserved</i>

10.5.9.9 Pin Multiplexing Control 8 Register (PINMUX8)
Figure 10-25. Pin Multiplexing Control 8 Register (PINMUX8)

31	28	27	24	23	20	19	16
PINMUX8_31_28		PINMUX8_27_24		PINMUX8_23_20		PINMUX8_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX8_15_12		PINMUX8_11_8		PINMUX8_7_4		PINMUX8_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-29. Pin Multiplexing Control 8 Register (PINMUX8) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX8_31_28	R4	7	0 1h 2h 3h-7h 8h 9h-Fh	SPI1_ENA/UART2_RXD/GP5[12] Control Pin is 3-stated. Selects Function SPI1_ENA Selects Function UART2_RXD <i>Reserved</i> Selects Function GP5[12] <i>Reserved</i>
27-24	PINMUX8_27_24	T4	6	0 1h 2h-7h 8h 9h-Fh	AXR1[11]/GP5[11] Control Pin is 3-stated. Selects Function AXR1[11] <i>Reserved</i> Selects Function GP5[11] <i>Reserved</i>
23-20	PINMUX8_23_20	N3	4	0 1h 2h-7h 8h 9h-Fh	AXR1[10]/GP5[10] Control Pin is 3-stated. Selects Function AXR1[10] <i>Reserved</i> Selects Function GP5[10] <i>Reserved</i>
19-16	PINMUX8_19_16	P3	3	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	UART0_TXD/I2C0_SCL/TM64P0_OUT12/GP5[9]/BOOT[9] Control Pin is 3-stated. Selects Function UART0_TXD Selects Function I2C0_SCL <i>Reserved</i> Selects Function TM64P0_OUT12 <i>Reserved</i> Selects Function GP5[9] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-29. Pin Multiplexing Control 8 Register (PINMUX8) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX8_15_12	R3	2	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	UART0_RXD/I2C0_SDA/TM64P0_IN12/GP5[8]/BOOT[8] Control Pin is 3-stated. Selects Function UART0_RXD Selects Function I2C0_SDA <i>Reserved</i> Selects Function TM64P0_IN12 <i>Reserved</i> Selects Function GP5[8] <i>Reserved</i>
11-8	PINMUX8_11_8	T6	16	0 1h 2h 3h-7h 8h 9h-Fh	SPI1_CLK/EQEP1S/GP5[7]/BOOT[7] Control Pin is 3-stated. Selects Function SPI1_CLK Selects Function EQEP1S <i>Reserved</i> Selects Function GP5[7] <i>Reserved</i>
7-4	PINMUX8_7_4	N5	14	0 1h 2h 3h-7h 8h 9h-Fh	SPI1_SIMO[0]/I2C1_SDA/GP5[6]/BOOT[6] Control Pin is 3-stated. Selects Function SPI1_SIMO[0] Selects Function I2C1_SDA <i>Reserved</i> Selects Function GP5[6] <i>Reserved</i>
3-0	PINMUX8_3_0	P5	13	0 1h 2h 3h-7h 8h 9h-Fh	SPI1_SOMI[0]/I2C1_SCL/GP5[5]/BOOT[5] Control Pin is 3-stated. Selects Function SPI1_SOMI[0] Selects Function I2C1_SCL <i>Reserved</i> Selects Function GP5[5] <i>Reserved</i>

10.5.9.10 Pin Multiplexing Control 9 Register (PINMUX9)
Figure 10-26. Pin Multiplexing Control 9 Register (PINMUX9)

31	28	27	24	23	20	19	16
PINMUX9_31_28		PINMUX9_27_24		PINMUX9_23_20		PINMUX9_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX9_15_12		PINMUX9_11_8		PINMUX9_7_4		PINMUX9_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-30. Pin Multiplexing Control 9 Register (PINMUX9) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX9_31_28	C4	131	0 1h 2h-7h 8h 9h-Fh	AFSR0/GP3[12] Control Pin is 3-stated. Selects Function AFSR0 <i>Reserved</i> Selects Function GP3[12] <i>Reserved</i>
27-24	PINMUX9_27_24	B4	130	0 1h 2h 3h-7h 8h 9h-Fh	ACLKR0/ECAP1/APWM1/GP2[15] Control Pin is 3-stated. Selects Function ACLKR0 Selects Function ECAP1/APWM1 <i>Reserved</i> Selects Function GP2[15] <i>Reserved</i>
23-20	PINMUX9_23_20	A4	129	0 1h 2h 3h-7h 8h 9h-Fh	AHCLKR0/RMII_MHZ_50_CLK/GP2[14]/BOOT[11] Control Pin is 3-stated. Enables sourcing of the EMAC 50 MHz reference clock from an external source on the RMII_MHZ_50_CLK pin. Selects Function AHCLKR0 Selects Function RMII_MHZ_50_CLK. Enables sourcing of the EMAC 50 MHz reference clock from PLL SYCLK7. Also, SYCLK7 is driven out on the RMII_MHZ_50_CLK pin. <i>Reserved</i> Selects Function GP2[14] <i>Reserved</i>
19-16	PINMUX9_19_16	D5	127	0 1h 2h-7h 8h 9h-Fh	AFSX0/GP2[13]/BOOT[10] Control Pin is 3-stated. Selects Function AFSX0 <i>Reserved</i> Selects Function GP2[13] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-30. Pin Multiplexing Control 9 Register (PINMUX9) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX9_15_12	C5	126	0 1h 2h 3h-7h 8h 9h-Fh	ACLKX0/ECAP0/APWM0/GP2[12] Control Pin is 3-stated. Selects Function ACLKX0 Selects Function ECAP0/APWM0 <i>Reserved</i> Selects Function GP2[12] <i>Reserved</i>
11-8	PINMUX9_11_8	B5	125	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AHCLKX0/AHCLKX2/USB_REFCLKIN/GP2[11] Control Pin is 3-stated. Selects Function AHCLKX0 Selects Function AHCLKX2 <i>Reserved</i> Selects Function USB_REFCLKIN <i>Reserved</i> Selects Function GP2[11] <i>Reserved</i>
7-4	PINMUX9_7_4	E4	—	0 1h 2h-7h 8h 9h-Fh	USB0_DRVVBUS/GP4[15] Control Pin is 3-stated. Selects Function USB0_DRVVBUS <i>Reserved</i> Selects Function GP4[15] <i>Reserved</i>
3-0	PINMUX9_3_0	P4	8	0 1h 2h 3h-7h 8h 9h-Fh	SPI1_SCS[0]/UART2_TXD/GP5[13] Control Pin is 3-stated. Selects Function <u>SPI1_SCS[0]</u> Selects Function UART2_TXD <i>Reserved</i> Selects Function GP5[13] <i>Reserved</i>

10.5.9.11 Pin Multiplexing Control 10 Register (PINMUX10)
Figure 10-27. Pin Multiplexing Control 10 Register (PINMUX10)

31	28	27	24	23	20	19	16
PINMUX10_31_28		PINMUX10_27_24		PINMUX10_23_20		PINMUX10_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX10_15_12		PINMUX10_11_8		PINMUX10_7_4		PINMUX10_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-31. Pin Multiplexing Control 10 Register (PINMUX10) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX10_31_28	D7	118	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AXR0[6]/RMII_RXER[0]/ACLKR2/GP3[6] Control Pin is 3-stated. Selects Function AXR0[6] Selects Function RMII_RXER[0] <i>Reserved</i> Selects Function ACLKR2 <i>Reserved</i> Selects Function GP3[6] <i>Reserved</i>
27-24	PINMUX10_27_24	C7	117	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AXR0[5]/RMII_RXD[1]/AFSX2/GP3[5] Control Pin is 3-stated. Selects Function AXR0[5] Selects Function RMII_RXD[1] <i>Reserved</i> Selects Function AFSX2 <i>Reserved</i> Selects Function GP3[5] <i>Reserved</i>
23-20	PINMUX10_23_20	B7	116	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AXR0[4]/RMII_RXD[0]/AXR2[1]/GP3[4] Control Pin is 3-stated. Selects Function AXR0[4] Selects Function RMII_RXD[0] <i>Reserved</i> Selects Function AXR2[1] <i>Reserved</i> Selects Function GP3[4] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-31. Pin Multiplexing Control 10 Register (PINMUX10) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
19-16	PINMUX10_19_16	A7	115	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AXR0[3]/RMII_CRS_DV/AXR2[2]/GP3[3] Control Pin is 3-stated. Selects Function AXR0[3] Selects Function RMII_CRS_DV <i>Reserved</i> Selects Function AXR2[2] <i>Reserved</i> Selects Function GP3[3] <i>Reserved</i>
15-12	PINMUX10_15_12	D8	113	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AXR0[2]/RMII_TXEN/AXR2[3]/GP3[2] Control Pin is 3-stated. Selects Function AXR0[2] Selects Function RMII_TXEN <i>Reserved</i> Selects Function AXR2[3] <i>Reserved</i> Selects Function GP3[2] <i>Reserved</i>
11-8	PINMUX10_11_8	C8	112	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AXR0[1]/RMII_TXD[1]/ACLKX2/GP3[1] Control Pin is 3-stated. Selects Function AXR0[1] Selects Function RMII_TXD[1] <i>Reserved</i> Selects Function ACLKX2 <i>Reserved</i> Selects Function GP3[1] <i>Reserved</i>
7-4	PINMUX10_7_4	B8	111	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AXR0[0]/RMII_TXD[0]/AFSR2/GP3[0] Control Pin is 3-stated. Selects Function AXR0[0] Selects Function RMII_TXD[0] <i>Reserved</i> Selects Function AFSR2 <i>Reserved</i> Selects Function GP3[0] <i>Reserved</i>
3-0	PINMUX10_3_0	L4	—	0 1h 2h-7h 8h 9h-Fh	AMUTE0/RESETOUT Control Selects Function <u>RESETOUT</u> Selects Function AMUTE0 <i>Reserved</i> Selects Function <u>RESETOUT</u> <i>Reserved</i>

10.5.9.12 Pin Multiplexing Control 11 Register (PINMUX11)
Figure 10-28. Pin Multiplexing Control 11 Register (PINMUX11)

31	28	27	24	23	20	19	16
PINMUX11_31_28		PINMUX11_27_24		PINMUX11_23_20		PINMUX11_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX11_15_12		PINMUX11_11_8		PINMUX11_7_4		PINMUX11_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-32. Pin Multiplexing Control 11 Register (PINMUX11) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX11_31_28	K4	163	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	AFSX1/EPWMSYNCl/EPWMSYNCO/GP4[10] Control Pin is 3-stated. Selects Function AFSX1 Selects Function EPWMSYNCl <i>Reserved</i> Selects Function EPWMSYNCO <i>Reserved</i> Selects Function GP4[10] <i>Reserved</i>
27-24	PINMUX11_27_24	K3	162	0 1h 2h 3h-7h 8h 9h-Fh	ACLKX1/EPWM0A/GP3[15] Control Pin is 3-stated. Selects Function ACLKX1 Selects Function EPWM0A <i>Reserved</i> Selects Function GP3[15] <i>Reserved</i>
23-20	PINMUX11_23_20	K2	160	0 1h 2h 3h-7h 8h 9h-Fh	AHCLKX1/EPWM0B/GP3[14] Control Pin is 3-stated. Selects Function AHCLKX1 Selects Function EPWM0B <i>Reserved</i> Selects Function GP3[14] <i>Reserved</i>
19-16	PINMUX11_19_16	A5	124	0 1h 2h-3h 4h 5h-7h 8h 9h-Fh	AXR0[11]/AXR2[0]/GP3[11] Control Pin is 3-stated. Selects Function AXR0[11] <i>Reserved</i> Selects Function AXR2[0] <i>Reserved</i> Selects Function GP3[11] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-32. Pin Multiplexing Control 11 Register (PINMUX11) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX11_15_12	D6	123	0 1h 2h 3h-7h 8h 9h-Fh	UART1_TXD/AXR0[10]/GP3[10] Control Pin is 3-stated. Selects Function UART1_TXD Selects Function AXR0[10] <i>Reserved</i> Selects Function GP3[10] <i>Reserved</i>
11-8	PINMUX11_11_8	C6	122	0 1h 2h 3h-7h 8h 9h-Fh	UART1_RXD/AXR0[9]/GP3[9] Control Pin is 3-stated. Selects Function UART1_RXD Selects Function AXR0[9] <i>Reserved</i> Selects Function GP3[9] <i>Reserved</i>
7-4	PINMUX11_7_4	B6	121	0 1h 2h 3h-7h 8h 9h-Fh	AXR0[8]/MDIO_D/GP3[8] Control Pin is 3-stated. Selects Function AXR0[8] Selects Function MDIO_D <i>Reserved</i> Selects Function GP3[8] <i>Reserved</i>
3-0	PINMUX11_3_0	A6	120	0 1h 2h 3h-7h 8h 9h-Fh	AXR0[7]/MDIO_CLK/GP3[7] Control Pin is 3-stated. Selects Function AXR0[7] Selects Function MDIO_CLK <i>Reserved</i> Selects Function GP3[7] <i>Reserved</i>

10.5.9.13 Pin Multiplexing Control 12 Register (PINMUX12)

Figure 10-29. Pin Multiplexing Control 12 Register (PINMUX12)

31	28	27	24	23	20	19	16
PINMUX12_31_28		PINMUX12_27_24		PINMUX12_23_20		PINMUX12_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX12_15_12		PINMUX12_11_8		PINMUX12_7_4		PINMUX12_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-33. Pin Multiplexing Control 12 Register (PINMUX12) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX12_31_28	P1	174	0 1h 2h 3h-7h 8h 9h-Fh	AXR1[3]/EQEP1A/GP4[3] Control Pin is 3-stated. Selects Function AXR1[3] Selects Function EQEP1A <i>Reserved</i> Selects Function GP4[3] <i>Reserved</i>
27-24	PINMUX12_27_24	P2	175	0 1h 2h-7h 8h 9h-Fh	AXR1[2]/GP4[2] Control Pin is 3-stated. Selects Function AXR1[2] <i>Reserved</i> Selects Function GP4[2] <i>Reserved</i>
23-20	PINMUX12_23_20	R2	176	0 1h 2h-7h 8h 9h-Fh	AXR1[1]/GP4[1] Control Pin is 3-stated. Selects Function AXR1[1] <i>Reserved</i> Selects Function GP4[1] <i>Reserved</i>
19-16	PINMUX12_19_16	T3	1	0 1h 2h-7h 8h 9h-Fh	AXR1[0]/GP4[0] Control Pin is 3-stated. Selects Function AXR1[0] <i>Reserved</i> Selects Function GP4[0] <i>Reserved</i>
15-12	PINMUX12_15_12	D4	132	0 1h 2h 3h-7h 8h 9h-Fh	AMUTE1/EHRPWMTZ/GP4[14] Control Pin is 3-stated. Selects Function AMUTE1 Selects Function EHRPWMTZ <i>Reserved</i> Selects Function GP4[14] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-33. Pin Multiplexing Control 12 Register (PINMUX12) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
11-8	PINMUX12_11_8	L3	166	0 1h 2h-7h 8h 9h-Fh	AFSR1/GP4[13] Control Pin is 3-stated. Selects Function AFSR1 <i>Reserved</i> Selects Function GP4[13] <i>Reserved</i>
7-4	PINMUX12_7_4	L2	165	0 1h 2h 3h-7h 8h 9h-Fh	ACLKR1/ECAP2/APWM2/GP4[12] Control Pin is 3-stated. Selects Function ACLKR1 Selects Function ECAP2/APWM2 <i>Reserved</i> Selects Function GP4[12] <i>Reserved</i>
3-0	PINMUX12_3_0	L1	—	0 1h 2h-7h 8h 9h-Fh	AHCLKR1/GP4[11] Control Pin is 3-stated. Selects Function AHCLKR1 <i>Reserved</i> Selects Function GP4[11] <i>Reserved</i>

10.5.9.14 Pin Multiplexing Control 13 Register (PINMUX13)
Figure 10-30. Pin Multiplexing Control 13 Register (PINMUX13)

31	28	27	24	23	20	19	16
PINMUX13_31_28		PINMUX13_27_24		PINMUX13_23_20		PINMUX13_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX13_15_12		PINMUX13_11_8		PINMUX13_7_4		PINMUX13_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-34. Pin Multiplexing Control 13 Register (PINMUX13) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX13_31_28	R15	45	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[1]/MMCS_DAT[1]/UHPI_HD[1]/GP0[1] Control Pin is 3-stated. Selects Function EMA_D[1] Selects Function MMCS_DAT[1] <i>Reserved</i> Selects Function UHPI_HD[1] <i>Reserved</i> Selects Function GP0[1] <i>Reserved</i>
27-24	PINMUX13_27_24	T13	44	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[0]/MMCS_DAT[0]/UHPI_HD[0]/GP0[0]/BOOT[12] Control Pin is 3-stated. Selects Function EMA_D[0] Selects Function MMCS_DAT[0] <i>Reserved</i> Selects Function UHPI_HD[0] <i>Reserved</i> Selects Function GP0[0] <i>Reserved</i>
23-20	PINMUX13_23_20	M1	—	0 1h 2h-7h 8h 9h-Fh	AXR1[9]/GP4[9] Control Pin is 3-stated. Selects Function AXR1[9] <i>Reserved</i> Selects Function GP4[9] <i>Reserved</i>
19-16	PINMUX13_19_16	M2	168	0 1h 2h 3h-7h 8h 9h-Fh	AXR1[8]/EPWM1A/GP4[8] Control Pin is 3-stated. Selects Function AXR1[8] Selects Function EPWM1A <i>Reserved</i> Selects Function GP4[8] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-34. Pin Multiplexing Control 13 Register (PINMUX13) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX13_15_12	M3	169	0 1h 2h 3h-7h 8h 9h-Fh	AXR1[7]/EPWM1B/GP4[7] Control Pin is 3-stated. Selects Function AXR1[7] Selects Function EPWM1B <i>Reserved</i> Selects Function GP4[7] <i>Reserved</i>
11-8	PINMUX13_11_8	M4	170	0 1h 2h 3h-7h 8h 9h-Fh	AXR1[6]/EPWM2A/GP4[6] Control Pin is 3-stated. Selects Function AXR1[6] Selects Function EPWM2A <i>Reserved</i> Selects Function GP4[6] <i>Reserved</i>
7-4	PINMUX13_7_4	N1	171	0 1h 2h 3h-7h 8h 9h-Fh	AXR1[5]/EPWM2B/GP4[5] Control Pin is 3-stated. Selects Function AXR1[5] Selects Function EPWM2B <i>Reserved</i> Selects Function GP4[5] <i>Reserved</i>
3-0	PINMUX13_3_0	N2	173	0 1h 2h 3h-7h 8h 9h-Fh	AXR1[4]/EQEP1B/GP4[4] Control Pin is 3-stated. Selects Function AXR1[4] Selects Function EQEP1B <i>Reserved</i> Selects Function GP4[4] <i>Reserved</i>

10.5.9.15 Pin Multiplexing Control 14 Register (PINMUX14)
Figure 10-31. Pin Multiplexing Control 14 Register (PINMUX14)

31	28	27	24	23	20	19	16
PINMUX14_31_28		PINMUX14_27_24		PINMUX14_23_20		PINMUX14_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX14_15_12		PINMUX14_11_8		PINMUX14_7_4		PINMUX14_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-35. Pin Multiplexing Control 14 Register (PINMUX14) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX14_31_28	T14	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[9]/UHPI_HD[9]/LCD_D[9]/GP0[9] Control Pin is 3-stated. Selects Function EMA_D[9] Selects Function UHPI_HD[9] <i>Reserved</i> Selects Function LCD_D[9] <i>Reserved</i> Selects Function GP0[9] <i>Reserved</i>
27-24	PINMUX14_27_24	N12	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[8]/UHPI_HD[8]/LCD_D[8]/GP0[8] Control Pin is 3-stated. Selects Function EMA_D[8] Selects Function UHPI_HD[8] <i>Reserved</i> Selects Function LCD_D[8] <i>Reserved</i> Selects Function GP0[8] <i>Reserved</i>
23-20	PINMUX14_23_20	M15	54	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[7]/MMCS_DAT[7]/UHPI_HD[7]/GP0[7]/BOOT[13] Control Pin is 3-stated. Selects Function EMA_D[7] Selects Function MMCS_DAT[7] <i>Reserved</i> Selects Function UHPI_HD[7] <i>Reserved</i> Selects Function GP0[7] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-35. Pin Multiplexing Control 14 Register (PINMUX14) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
19-16	PINMUX14_19_16	N13	52	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[6]/MMCS_DAT[6]/UHPI_HD[6]/GP0[6] Control Pin is 3-stated. Selects Function EMA_D[6] Selects Function MMCS_DAT[6] <i>Reserved</i> Selects Function UHPI_HD[6] <i>Reserved</i> Selects Function GP0[6] <i>Reserved</i>
15-12	PINMUX14_15_12	N15	51	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[5]/MMCS_DAT[5]/UHPI_HD[5]/GP0[5] Control Pin is 3-stated. Selects Function EMA_D[5] Selects Function MMCS_DAT[5] <i>Reserved</i> Selects Function UHPI_HD[5] <i>Reserved</i> Selects Function GP0[5] <i>Reserved</i>
11-8	PINMUX14_11_8	P13	49	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[4]/MMCS_DAT[4]/UHPI_HD[4]/GP0[4] Control Pin is 3-stated. Selects Function EMA_D[4] Selects Function MMCS_DAT[4] <i>Reserved</i> Selects Function UHPI_HD[4] <i>Reserved</i> Selects Function GP0[4] <i>Reserved</i>
7-4	PINMUX14_7_4	P15	48	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[3]/MMCS_DAT[3]/UHPI_HD[3]/GP0[3] Control Pin is 3-stated. Selects Function EMA_D[3] Selects Function MMCS_DAT[3] <i>Reserved</i> Selects Function UHPI_HD[3] <i>Reserved</i> Selects Function GP0[3] <i>Reserved</i>
3-0	PINMUX14_3_0	R13	46	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[2]/MMCS_DAT[2]/UHPI_HD[2]/GP0[2] Control Pin is 3-stated. Selects Function EMA_D[2] Selects Function MMCS_DAT[2] <i>Reserved</i> Selects Function UHPI_HD[2] <i>Reserved</i> Selects Function GP0[2] <i>Reserved</i>

10.5.9.16 Pin Multiplexing Control 15 Register (PINMUX15)

Figure 10-32. Pin Multiplexing Control 15 Register (PINMUX15)

31	28	27	24	23	20	19	16
PINMUX15_31_28		PINMUX15_27_24		PINMUX15_23_20		PINMUX15_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX15_15_12		PINMUX15_11_8		PINMUX15_7_4		PINMUX15_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-36. Pin Multiplexing Control 15 Register (PINMUX15) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX15_31_28	R9	30	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_A[1]/MMCS_D_CLK/UHPI_HCNTL0/GP1[1] Control Pin is 3-stated. Selects Function EMA_A[1] Selects Function MMCS_D_CLK <i>Reserved</i> Selects Function UHPI_HCNTL0 <i>Reserved</i> Selects Function GP1[1] <i>Reserved</i>
27-24	PINMUX15_27_24	T9	29	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[0]/LCD_D[7]/GP1[0] Control Pin is 3-stated. Selects Function EMA_A[0] Selects Function LCD_D[7] <i>Reserved</i> Selects Function GP1[0] <i>Reserved</i>
23-20	PINMUX15_23_20	M16	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[15]/UHPI_HD[15]/LCD_D[15]/GP0[15] Control Pin is 3-stated. Selects Function EMA_D[15] Selects Function UHPI_HD[15] <i>Reserved</i> Selects Function LCD_D[15] <i>Reserved</i> Selects Function GP0[15] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-36. Pin Multiplexing Control 15 Register (PINMUX15) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
19-16	PINMUX15_19_16	N14	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[14]/UHPI_HD[14]/LCD_D[14]/GP0[14] Control Pin is 3-stated. Selects Function EMA_D[14] Selects Function UHPI_HD[14] <i>Reserved</i> Selects Function LCD_D[14] <i>Reserved</i> Selects Function GP0[14] <i>Reserved</i>
15-12	PINMUX15_15_12	N16	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[13]/UHPI_HD[13]/LCD_D[13]/GP0[13] Control Pin is 3-stated. Selects Function EMA_D[13] Selects Function UHPI_HD[13] <i>Reserved</i> Selects Function LCD_D[13] <i>Reserved</i> Selects Function GP0[13] <i>Reserved</i>
11-8	PINMUX15_11_8	P14	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[12]/UHPI_HD[12]/LCD_D[12]/GP0[12] Control Pin is 3-stated. Selects Function EMA_D[12] Selects Function UHPI_HD[12] <i>Reserved</i> Selects Function LCD_D[12] <i>Reserved</i> Selects Function GP0[12] <i>Reserved</i>
7-4	PINMUX15_7_4	P16	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[11]/UHPI_HD[11]/LCD_D[11]/GP0[11] Control Pin is 3-stated. Selects Function EMA_D[11] Selects Function UHPI_HD[11] <i>Reserved</i> Selects Function LCD_D[11] <i>Reserved</i> Selects Function GP0[11] <i>Reserved</i>
3-0	PINMUX15_3_0	R14	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_D[10]/UHPI_HD[10]/LCD_D[10]/GP0[10] Control Pin is 3-stated. Selects Function EMA_D[10] Selects Function UHPI_HD[10] <i>Reserved</i> Selects Function LCD_D[10] <i>Reserved</i> Selects Function GP0[10] <i>Reserved</i>

10.5.9.17 Pin Multiplexing Control 16 Register (PINMUX16)
Figure 10-33. Pin Multiplexing Control 16 Register (PINMUX16)

31	28	27	24	23	20	19	16
PINMUX16_31_28		PINMUX16_27_24		PINMUX16_23_20		PINMUX16_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX16_15_12		PINMUX16_11_8		PINMUX16_7_4		PINMUX16_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-37. Pin Multiplexing Control 16 Register (PINMUX16) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX16_31_28	R11	40	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[9]/LCD_HSYNC/GP1[9] Control Pin is 3-stated. Selects Function EMA_A[9] Selects Function LCD_HSYNC <i>Reserved</i> Selects Function GP1[9] <i>Reserved</i>
27-24	PINMUX16_27_24	T11	39	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[8]/LCD_PCLK/GP1[8] Control Pin is 3-stated. Selects Function EMA_A[8] Selects Function LCD_PCLK <i>Reserved</i> Selects Function GP1[8] <i>Reserved</i>
23-20	PINMUX16_23_20	N10	37	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[7]/LCD_D[0]/GP1[7] Control Pin is 3-stated. Selects Function EMA_A[7] Selects Function LCD_D[0] <i>Reserved</i> Selects Function GP1[7] <i>Reserved</i>
19-16	PINMUX16_19_16	P10	36	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[6]/LCD_D[1]/GP1[6] Control Pin is 3-stated. Selects Function EMA_A[6] Selects Function LCD_D[1] <i>Reserved</i> Selects Function GP1[6] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-37. Pin Multiplexing Control 16 Register (PINMUX16) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX16_15_12	R10	35	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[5]/LCD_D[2]/GP1[5] Control Pin is 3-stated. Selects Function EMA_A[5] Selects Function LCD_D[2] <i>Reserved</i> Selects Function GP1[5] <i>Reserved</i>
11-8	PINMUX16_11_8	T10	34	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[4]/LCD_D[3]/GP1[4] Control Pin is 3-stated. Selects Function EMA_A[4] Selects Function LCD_D[3] <i>Reserved</i> Selects Function GP1[4] <i>Reserved</i>
7-4	PINMUX16_7_4	N9	32	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[3]/LCD_D[6]/GP1[3] Control Pin is 3-stated. Selects Function EMA_A[3] Selects Function LCD_D[6] <i>Reserved</i> Selects Function GP1[3] <i>Reserved</i>
3-0	PINMUX16_3_0	P9	31	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_A[2]/MMCSD_CMD/UHPI_HCNTL1/GP1[2] Control Pin is 3-stated. Selects Function EMA_A[2] Selects Function MMCSD_CMD <i>Reserved</i> Selects Function UHPI_HCNTL1 <i>Reserved</i> Selects Function GP1[2] <i>Reserved</i>

10.5.9.18 Pin Multiplexing Control 17 Register (PINMUX17)
Figure 10-34. Pin Multiplexing Control 17 Register (PINMUX17)

31	28	27	24	23	20	19	16
PINMUX17_31_28		PINMUX17_27_24		PINMUX17_23_20		PINMUX17_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX17_15_12		PINMUX17_11_8		PINMUX17_7_4		PINMUX17_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-38. Pin Multiplexing Control 17 Register (PINMUX17) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX17_31_28	L16	—	0 1h 2h 3h-7h 8h 9h-Fh	EMA_CAS/EMA_CS[4]/GP2[1] Control Pin is 3-stated. Selects Function EMA_CAS Selects Function EMA_CS[4] <i>Reserved</i> Selects Function GP2[1] <i>Reserved</i>
27-24	PINMUX17_27_24	T12	—	0 1h 2h-7h 8h 9h-Fh	EMA_SDCKE/GP2[0] Control Pin is 3-stated. Selects Function EMA_SDCKE <i>Reserved</i> Selects Function GP2[0] <i>Reserved</i>
23-20	PINMUX17_23_20	R12	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_CLK/OBSCLK/AHCLKR2/GP1[15] Control Pin is 3-stated. Selects Function EMA_CLK Selects Function OBSCLK. <i>Reserved</i> Selects Function AHCLKR2 <i>Reserved</i> Selects Function GP1[15] <i>Reserved</i>
19-16	PINMUX17_19_16	R8	25	0 1h 2h 3h-7h 8h 9h-Fh	EMA_BA[0]/LCD_D[4]/GP1[14] Control Pin is 3-stated. Selects Function EMA_BA[0] Selects Function LCD_D[4] <i>Reserved</i> Selects Function GP1[14] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-38. Pin Multiplexing Control 17 Register (PINMUX17) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
15-12	PINMUX17_15_12	P8	26	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_BA[1]/LCD_D[5]/UHPI_HHWIL/GP1[13] Control Pin is 3-stated. Selects Function EMA_BA[1] Selects Function LCD_D[5] <i>Reserved</i> Selects Function UHPI_HHWIL <i>Reserved</i> Selects Function GP1[13] <i>Reserved</i>
11-8	PINMUX17_11_8	N11	42	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[12]/LCD_MCLK/GP1[12] Control Pin is 3-stated. Selects Function EMA_A[12] Selects Function LCD_MCLK <i>Reserved</i> Selects Function GP1[12] <i>Reserved</i>
7-4	PINMUX17_7_4	P11	41	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[11]/LCD_AC_ENB_CS/GP1[11] Control Pin is 3-stated. Selects Function EMA_A[11] Selects Function LCD_AC_ENB_CS <i>Reserved</i> Selects Function GP1[11] <i>Reserved</i>
3-0	PINMUX17_3_0	N8	27	0 1h 2h 3h-7h 8h 9h-Fh	EMA_A[10]/LCD_VSYNC/GP1[10] Control Pin is 3-stated. Selects Function EMA_A[10] Selects Function LCD_VSYNC <i>Reserved</i> Selects Function GP1[10] <i>Reserved</i>

10.5.9.19 Pin Multiplexing Control 18 Register (PINMUX18)
Figure 10-35. Pin Multiplexing Control 18 Register (PINMUX18)

31	28	27	24	23	20	19	16
PINMUX18_31_28		PINMUX18_27_24		PINMUX18_23_20		PINMUX18_19_16	
R/W-0		R/W-0		R/W-0		R/W-0	
15	12	11	8	7	4	3	0
PINMUX18_15_12		PINMUX18_11_8		PINMUX18_7_4		PINMUX18_3_0	
R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 10-39. Pin Multiplexing Control 18 Register (PINMUX18) Field Descriptions

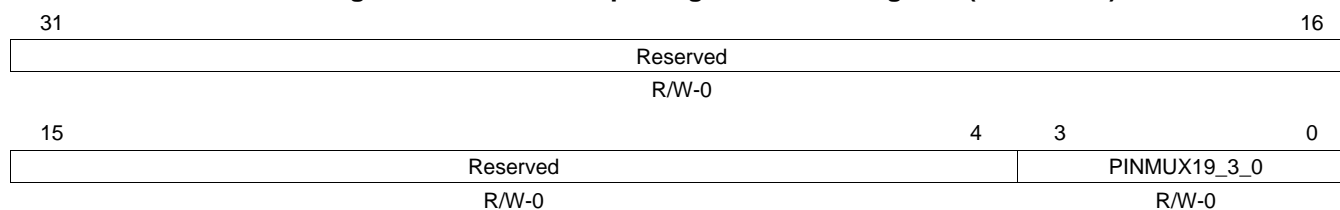
Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-28	PINMUX18_31_28	M14	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_WE_DQM[0]/UHPI_HINT/AXR0[15]/GP2[9] Control Pin is 3-stated. Selects Function EMA_WE_DQM[0] Selects Function UHPI_HINT <i>Reserved</i> Selects Function AXR0[15] <i>Reserved</i> Selects Function GP2[9] <i>Reserved</i>
27-24	PINMUX18_27_24	P12	—	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_WE_DQM[1]/UHPI_HDS2/AXR0[14]/GP2[8] Control Pin is 3-stated. Selects Function EMA_WE_DQM[1] Selects Function UHPI_HDS2 <i>Reserved</i> Selects Function AXR0[14] <i>Reserved</i> Selects Function GP2[8] <i>Reserved</i>
23-20	PINMUX18_23_20	R7	22	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_OE/UHPI_HDS1/AXR0[13]/GP2[7] Control Pin is 3-stated. Selects Function EMA_OE Selects Function UHPI_HDS1 <i>Reserved</i> Selects Function AXR0[13] <i>Reserved</i> Selects Function GP2[7] <i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

Table 10-39. Pin Multiplexing Control 18 Register (PINMUX18) Field Descriptions (continued)

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
19-16	PINMUX18_19_16	T7	21	0 1h 2h-3h 4h 5h-7h 8h 9h-Fh	EMA_CS[3]/AMUTE2/GP2[6] Control Pin is 3-stated. Selects Function <u>EMA_CS[3]</u> <i>Reserved</i> Selects Function AMUTE2 <i>Reserved</i> Selects Function GP2[6] <i>Reserved</i>
15-12	PINMUX18_15_12	P7	23	0 1h 2h 3h-7h 8h 9h-Fh	EMA_CS[2]/UHPI_HCS/GP2[5]/BOOT[15] Control Pin is 3-stated. Selects Function <u>EMA_CS[2]</u> Selects Function <u>UHPI_HCS</u> <i>Reserved</i> Selects Function GP2[5] <i>Reserved</i>
11-8	PINMUX18_11_8	T8	—	0 1h 2h 3h-7h 8h 9h-Fh	EMA_CS[0]/UHPI_HAS/GP2[4] Control Pin is 3-stated. Selects Function <u>EMA_CS[0]</u> Selects Function <u>UHPI_HAS</u> <i>Reserved</i> Selects Function GP2[4] <i>Reserved</i>
7-4	PINMUX18_7_4	M13	55	0 1h 2h 3h 4h 5h-7h 8h 9h-Fh	EMA_WE/UHPI_HR\bar{W}/AXR0[12]/GP2[3]/BOOT[14] Control Pin is 3-stated. Selects Function <u>EMA_WE</u> Selects Function <u>UHPI_HR\bar{W}</u> <i>Reserved</i> Selects Function AXR0[12] <i>Reserved</i> Selects Function GP2[3] <i>Reserved</i>
3-0	PINMUX18_3_0	N7	—	0 1h 2h 3h-7h 8h 9h-Fh	EMA_RAS/EMA_CS[5]/GP2[2] Control Pin is 3-stated. Selects Function <u>EMA_RAS</u> Selects Function <u>EMA_CS[5]</u> <i>Reserved</i> Selects Function GP2[2] <i>Reserved</i>

10.5.9.20 Pin Multiplexing Control 19 Register (PINMUX19)
Figure 10-36. Pin Multiplexing Control 19 Register (PINMUX19)


LEGEND: R/W = Read/Write; -n = value after reset

Table 10-40. Pin Multiplexing Control 19 Register (PINMUX19) Field Descriptions

Bit	Field	ZKB Ball ⁽¹⁾	PTP Pin ⁽²⁾	Value	Description
31-4	Reserved	—	—	0	Reserved
3-0	PINMUX19_3_0	N6	19	0	EMA_WAIT[0]/UHPI_HRDY/GP2[10] Control
				0	Pin is 3-stated.
				1h	Selects Function EMA_WAIT[0]
				2h	Selects Function UHPI_HRDY
				3h-7h	<i>Reserved</i>
				8h	Selects Function GP2[10]
				9h-Fh	<i>Reserved</i>

⁽¹⁾ The ZKB ball package is only available on the C6747 DSP; this package is not supported on the C6745 DSP.

⁽²⁾ The PTP pin package is only available on the C6745 DSP; this package is not supported on the C6747 DSP.

10.5.10 Suspend Source Register (SUSPSRC)

The suspend source register (SUSPSRC) indicates the emulation suspend source for those peripherals that support emulation suspend. A value of 1 (default) for a SUSPSRC bit corresponding to the peripheral, indicates that the DSP emulator controls the peripheral's emulation suspend signal. You should maintain this register with its default values.

The SUSPSRC is shown in [Figure 10-37](#) and described in [Table 10-41](#).

Figure 10-37. Suspend Source Register (SUSPSRC)

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	TIMER64_1SRC	TIMER64_0SRC	Reserved	EPWM2SRC	EPWM1SRC
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
23	22	21	20	19	18	17	16
EPWM0SRC	SPI1SRC	SPI0SRC	UART2SRC	UART1SRC	UART0SRC	I2C1SRC	I2C0SRC
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	HPISRC ⁽¹⁾	Reserved	Reserved	USB0SRC	Reserved
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
7	6	5	4	3	2	1	0
Reserved	PRUSRC	EMACSRC	EQEP1SRC	EQEP0SRC	ECAP2SRC	ECAP1SRC	ECAP0SRC
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

LEGEND: R/W = Read/Write; -n = value after reset

⁽¹⁾ This bit is not supported and is Reserved on the C6745 DSP.

Table 10-41. Suspend Source Register (SUSPSRC) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	1	Reserved. Write the default value to all bits when modifying this register.
28	TIMER64_1SRC	0	Timer1 64 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
27	TIMER64_0SRC	0	Timer0 64 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
26	Reserved	1	Reserved. Write the default value to all bits when modifying this register.
25	EPWM2SRC	0	EPWM2 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
24	EPWM1SRC	0	EPWM1 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
23	EPWM0SRC	0	EPWM0 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
22	SPI1SRC	0	SPI1 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
21	SPI0SRC	0	SPI0 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.

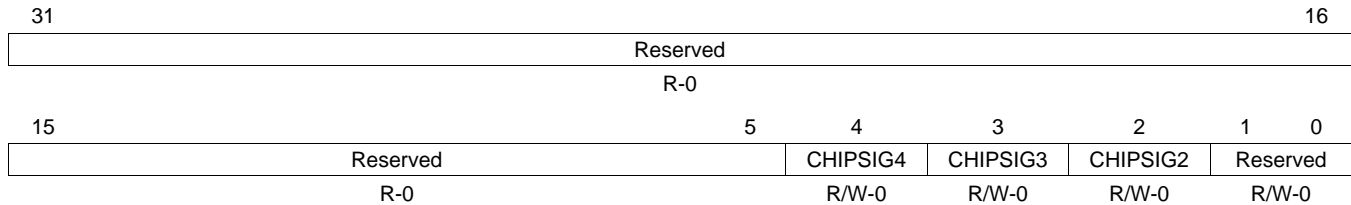
Table 10-41. Suspend Source Register (SUSPSRC) Field Descriptions (continued)

Bit	Field	Value	Description
20	UART2SRC	0	UART2 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
19	UART1SRC	0	UART1 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
18	UART0SRC	0	UART0 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
17	I2C1SRC	0	I2C1 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
16	I2C0SRC	0	I2C0 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
15-13	Reserved	1	Reserved. Write the default value to all bits when modifying this register.
12	HPISRC	0	HPI Emulation Suspend Source. This peripheral is not supported on the C6745 DSP. No emulation suspend.
		1	DSP is the source of the emulation suspend.
11-10	Reserved	1	Reserved. Write the default value to all bits when modifying this register.
9	USB0SRC	0	USB0 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
8-7	Reserved	1	Reserved. Write the default value to all bits when modifying this register.
6	PRUSRC	0	PRU Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
5	EMACSRC	0	EMAC Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
4	EQEP1SRC	0	EQEP1 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
3	EQEP0SRC	0	EQEP0 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
2	ECAP2SRC	0	ECAP2 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
1	ECAP1SRC	0	ECAP1 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.
0	ECAP0SRC	0	ECAP0 Emulation Suspend Source. No emulation suspend.
		1	DSP is the source of the emulation suspend.

10.5.11 Chip Signal Register (CHIPSIG)

The interrupts to the DSP can be generated by setting one of the two CHIPSIG[3-2] bits or an NMI interrupt by setting the CHIPSIG[4] bit in the chip signal register (CHIPSIG). Writing a 1 to these bits sets the interrupts, writing a 0 has no effect. Reads return the value of these bits and can also be used as status bits. The CHIPSIG is shown in [Figure 10-38](#) and described in [Table 10-42](#).

Figure 10-38. Chip Signal Register (CHIPSIG)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 10-42. Chip Signal Register (CHIPSIG) Field Descriptions

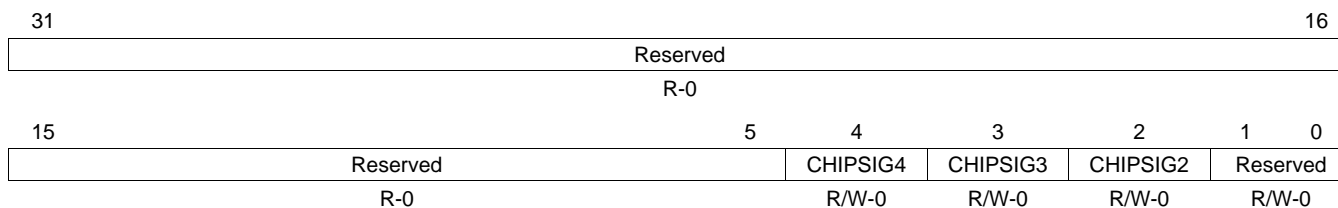
Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4	CHIPSIG4	0 1	Asserts DSP NMI interrupt. No effect Asserts interrupt
3	CHIPSIG3	0 1	Asserts SYSCFG_CHIPINT3 interrupt. No effect Asserts interrupt
2	CHIPSIG2	0 1	Asserts SYSCFG_CHIPINT2 interrupt. No effect Asserts interrupt
1-0	Reserved	0	Reserved. Write the default value to all bits when modifying this register.

10.5.12 Chip Signal Clear Register (CHIPSIG_CLR)

The chip signal clear register (CHIPSIG_CLR) is used to clear the bits set in the chip signal register (CHIPSIG). Writing a 1 to a CHIPSIG[*n*] bit in CHIPSIG_CLR clears the corresponding CHIPSIG[*n*] bit in CHIPSIG; writing a 0 has no effect. After servicing the interrupt, the interrupted processor can clear the bits set in CHIPSIG by writing 1 to the corresponding bits in CHIPSIG_CLR. The other processor may poll the CHIPSIG[*n*] bit to determine when the interrupted processor has completed the interrupt service. The CHIPSIG_CLR is shown in Figure 10-39 and described in Table 10-43.

For more information on DSP interrupts, see Chapter 2.

Figure 10-39. Chip Signal Clear Register (CHIPSIG_CLR)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-43. Chip Signal Clear Register (CHIPSIG_CLR) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4	CHIPSIG4	0	No effect
		1	Clears interrupt
3	CHIPSIG3	0	No effect
		1	Clears interrupt
2	CHIPSIG2	0	No effect
		1	Clears interrupt
1-0	Reserved	0	Reserved. Write the default value to all bits when modifying this register.

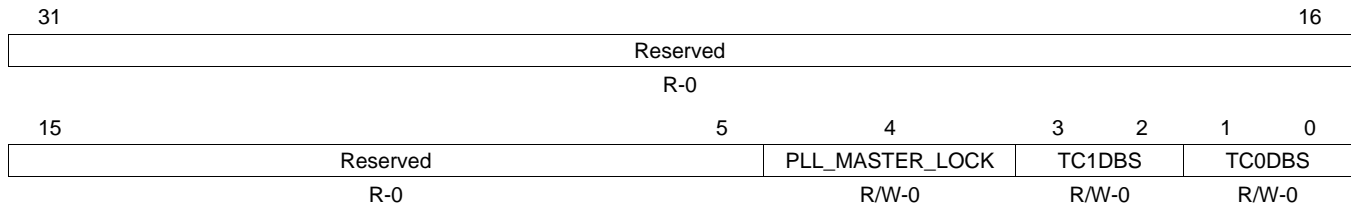
10.5.13 Chip Configuration 0 Register (CFGCHIP0)

The chip configuration 0 register (CFGCHIP0) controls the following functions:

- PLL Controller memory-mapped register lock: Used to lock out writes to the PLL controller memory-mapped registers (MMRs) to prevent any erroneous writes in software to the PLL controller register space.
- EDMA3 Transfer Controller Default Burst Size (DBS) Control: This controls the maximum number of bytes issued per read/write command or the burst size for the individual transfer controllers (TCs) on the device. By default for all transfer controllers, the burst size is set to 16 bytes. However, CFGCHIP0 allows configurability of this parameter so that the TC can have a burst size of 16, 32, or 64 bytes. The burst size determines the intra packet efficiency for the EDMA3 transfers. Additionally, it also facilitates preemption at a system level, as all transfer requests are internally broken down by the transfer controller up to DBS size byte chunks and on a system level, each master's priority (configured by the MSTPRI register) is evaluated at burst size boundaries. The DBS value can significantly impact the standalone throughput performance depending on the source and destination (bus width/frequency/burst support etc) and the TC FIFO size, etc. Therefore, the DBS size configuration should be carefully analyzed to meet the system's throughput/performance requirements.

The CFGCHIP0 is shown in [Figure 10-40](#) and described in [Table 10-44](#).

Figure 10-40. Chip Configuration 0 Register (CFGCHIP0)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 10-44. Chip Configuration 0 Register (CFGCHIP0) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4	PLL_MASTER_LOCK	0	PLL MMRs lock. PLL MMRs are freely accessible.
		1	All PLL MMRs are locked.
3-2	TC1DBS	0	TC1 Default Burst Size (DBS). 16 bytes
		1h	32 bytes
		2h	64 bytes
		3h	Reserved
1-0	TC0DBS	0	TC0 Default Burst Size (DBS). 16 bytes
		1h	32 bytes
		2h	64 bytes
		3h	Reserved

10.5.14 Chip Configuration 1 Register (CFGCHIP1)

The chip configuration 1 register (CFGCHIP1) controls the following functions:

- eCAP0/1/2 event input source: Allows using McASP TX/RX events or various EMAC TX/RX threshold, pulse, or miscellaneous interrupt events as eCAP event input sources.
- HPI Control: Allows HPIEN bit control that determines whether or not the HPI module has control over the HPI pins (multiplexed with other peripheral pins). It also provides configurability to select whether the host address is a word address or a byte address mode.
- eHRPWM Time Base Clock (TBCLK) Synchronization: Allows the software to globally synchronize all enabled eHRPWM modules to the time base clock (TBCLK).
- McASP AMUTEIN signal source control: Allows selecting GPIO interrupt from different banks as source for the McASP AMUTEIN signal. CFGCHIP1 provides this signal source control for all McASPs on the device.

The CFGCHIP1 is shown in [Figure 10-41](#) and described in [Table 10-45](#).

Figure 10-41. Chip Configuration 1 Register (CFGCHIP1)

31	27	26	22	21	17	16
CAP2SRC		CAP1SRC		CAP0SRC		HPIBYTEAD ⁽¹⁾
R/W-0		R/W-0		R/W-0		R/W-0
15	14	13	12	11	8	
HPIENA ⁽¹⁾	Reserved		TBCLKSYNC	AMUTESEL2 ⁽¹⁾		
R/W-0	R-0		R/W-0	R/W-0		
7	4			3	0	
AMUTESEL1				AMUTESEL0 ⁽¹⁾		
R/W-0				R/W-0		

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

⁽¹⁾ This bit is not supported and is Reserved on the C6745 DSP.

Table 10-45. Chip Configuration 1 Register (CFGCHIP1) Field Descriptions

Bit	Field	Value	Description
31-27	CAP2SRC		Selects the eCAP2 module event input.
		0	eCAP2 Pin input
		1h	McASP0 TX DMA Event
		2h	McASP0 RX DMA Event
		3h	McASP1 TX DMA Event
		4h	McASP1 RX DMA Event
		5h	McASP2 TX DMA Event. This peripheral is not supported on the C6745 DSP.
		6h	McASP2 RX DMA Event. This peripheral is not supported on the C6745 DSP.
		7h	EMAC C0 RX Threshold Pulse Interrupt
		8h	EMAC C0 RX Pulse Interrupt
		9h	EMAC C0 TX Pulse Interrupt
		Ah	EMAC C0 Miscellaneous Interrupt
		Bh	EMAC C1 RX Threshold Pulse Interrupt
		Ch	EMAC C1 RX Pulse Interrupt
		Dh	EMAC C1 TX Pulse Interrupt
Eh	EMAC C1 Miscellaneous Interrupt		
Fh	EMAC C2 RX Threshold Pulse Interrupt		
10h	EMAC C2 RX Pulse Interrupt		
11h	EMAC C2 TX Pulse Interrupt		
12h	EMAC C2 Miscellaneous Interrupt		
		<i>13h-1Fh</i>	<i>Reserved</i>
26-22	CAP1SRC		Selects the eCAP1 module event input.
		0	eCAP1 Pin input
		1h	McASP0 TX DMA Event
		2h	McASP0 RX DMA Event
		3h	McASP1 TX DMA Event
		4h	McASP1 RX DMA Event
		5h	McASP2 TX DMA Event. This peripheral is not supported on the C6745 DSP.
		6h	McASP2 RX DMA Event. This peripheral is not supported on the C6745 DSP.
		7h	EMAC C0 RX Threshold Pulse Interrupt
		8h	EMAC C0 RX Pulse Interrupt
		9h	EMAC C0 TX Pulse Interrupt
		Ah	EMAC C0 Miscellaneous Interrupt
		Bh	EMAC C1 RX Threshold Pulse Interrupt
		Ch	EMAC C1 RX Pulse Interrupt
		Dh	EMAC C1 TX Pulse Interrupt
Eh	EMAC C1 Miscellaneous Interrupt		
Fh	EMAC C2 RX Threshold Pulse Interrupt		
10h	EMAC C2 RX Pulse Interrupt		
11h	EMAC C2 TX Pulse Interrupt		
12h	EMAC C2 Miscellaneous Interrupt		
		<i>13h-1Fh</i>	<i>Reserved</i>

Table 10-45. Chip Configuration 1 Register (CFGCHIP1) Field Descriptions (continued)

Bit	Field	Value	Description
21-17	CAP0SRC		Selects the eCAP0 module event input.
		0	eCAP0 Pin input
		1h	McASP0 TX DMA Event
		2h	McASP0 RX DMA Event
		3h	McASP1 TX DMA Event
		4h	McASP1 RX DMA Event
		5h	McASP2 TX DMA Event. This peripheral is not supported on the C6745 DSP.
		6h	McASP2 RX DMA Event. This peripheral is not supported on the C6745 DSP.
		7h	EMAC C0 RX Threshold Pulse Interrupt
		8h	EMAC C0 RX Pulse Interrupt
		9h	EMAC C0 TX Pulse Interrupt
		Ah	EMAC C0 Miscellaneous Interrupt
		Bh	EMAC C1 RX Threshold Pulse Interrupt
		Ch	EMAC C1 RX Pulse Interrupt
		Dh	EMAC C1 TX Pulse Interrupt
Eh	EMAC C1 Miscellaneous Interrupt		
Fh	EMAC C2 RX Threshold Pulse Interrupt		
10h	EMAC C2 RX Pulse Interrupt		
11h	EMAC C2 TX Pulse Interrupt		
12h	EMAC C2 Miscellaneous Interrupt		
13h-1Fh		<i>Reserved</i>	
16	HPIBYTEAD		HPI Byte/Word Address Mode select. This peripheral is not supported on the C6745 DSP.
		0	Host address is a word address.
		1	Host address is a byte address.
15	HPIENA		HPI Enable Bit. This peripheral is not supported on the C6745 DSP.
		0	HPI is disabled.
		1	HPI is enabled.
14-13	Reserved	0	Reserved. Always read as 0.
12	TBCLKSYNC		eHRPWM Module Time Base Clock (TBCLK) Synchronization. Allows you to globally synchronize all enabled eHRPWM modules to the time base clock (TBCLK).
		0	Time base clock (TBCLK) within each enabled eHRPWM module is stopped.
		1	All enabled eHRPWM module clocks are started with the first rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescaler bits in the TBCTL register of each eHRPWM module must be set identically.
11-8	AMUTESEL2		Selects the source of McASP2 AMUTEIN signal. This peripheral is not supported on the C6745 DSP.
		0	Drive McASP2 AMUTEIN signal low
		1h	GPIO Interrupt from Bank 0
		2h	GPIO Interrupt from Bank 1
		3h	GPIO Interrupt from Bank 2
		4h	GPIO Interrupt from Bank 3
		5h	GPIO Interrupt from Bank 4
		6h	GPIO Interrupt from Bank 5
		7h	GPIO Interrupt from Bank 6
		8h	GPIO Interrupt from Bank 7
9h-Fh		<i>Reserved</i>	

Table 10-45. Chip Configuration 1 Register (CFGCHIP1) Field Descriptions (continued)

Bit	Field	Value	Description
7-4	AMUTESEL1	0 1h 2h 3h 4h 5h 6h 7h 8h 9h-Fh	Selects the source of McASP1 AMUTEIN signal. Drive McASP1 AMUTEIN signal low GPIO Interrupt from Bank 0 GPIO Interrupt from Bank 1 GPIO Interrupt from Bank 2 GPIO Interrupt from Bank 3 GPIO Interrupt from Bank 4 GPIO Interrupt from Bank 5 GPIO Interrupt from Bank 6 GPIO Interrupt from Bank 7 <i>Reserved</i>
3-0	AMUTESEL0	0 1h 2h 3h 4h 5h 6h 7h 8h 9h-Fh	Selects the source of McASP0 AMUTEIN signal. The AMUTE0 signal is not supported on the C6745 DSP. Drive McASP0 AMUTEIN signal low GPIO Interrupt from Bank 0 GPIO Interrupt from Bank 1 GPIO Interrupt from Bank 2 GPIO Interrupt from Bank 3 GPIO Interrupt from Bank 4 GPIO Interrupt from Bank 5 GPIO Interrupt from Bank 6 GPIO Interrupt from Bank 7 <i>Reserved</i>

10.5.15 Chip Configuration 2 Register (CFGCHIP2)

The chip configuration 2 register (CFGCHIP2) controls the following functions:

- USB1.1 OHCI (this peripheral is not supported on the C6745 DSP.)
- USB2.0 OTG PHY

The CFGCHIP2 is shown in [Figure 10-42](#) and described in [Table 10-46](#).

Figure 10-42. Chip Configuration 2 Register (CFGCHIP2)

	Reserved	
	R-0	
23	Reserved	18 17 16
	R-0	R-0 R-0
15	RESET	14 13 12 11 10 9 8
	R/W-1	R/W-3h R/W-0 R/W-1 R/W-1 R/W-1 R/W-1 R/W-1
7	USB1SUSPENDM ⁽¹⁾	6 5 4 3 0
	R/W-0	R/W-0 R/W-0 R/W-0 R/W-0 R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

⁽¹⁾ This bit is not supported and is Reserved on the C6745 DSP.

Table 10-46. Chip Configuration 2 Register (CFGCHIP2) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved
17	USB0PHYCLKGD	0 1	Status of USB2.0 PHY. 0 Clock is not present, power is not good, and PLL has not locked. 1 Clock is present, power is good, and PLL has locked.
16	USB0VBUSSENSE	0 1	Status of USB2.0 PHY VBUS sense. 0 PHY is not sensing voltage presence on the VBUS pin. 1 PHY is sensing voltage presence on the VBUS pin.
15	RESET	0 1	USB2.0 PHY reset. 0 Not in reset 1 USB2.0 PHY in reset
14-13	USB0OTGMODE	0 1h 2h 3h	USB2.0 OTG subsystem mode. 0 No override. PHY drive signals to controller based on its comparators for VBUS and ID pins. 1h Override phy values to force USB host operation. 2h Override phy values to force USB device operation. 3h Override phy values to force USB host operation with VBUS low.
12	USB1PHYCLKMUX	0 1	USB1.1 PHY reference clock input mux. Controls clock mux to USB1.1. This peripheral is not supported on the C6745 DSP. 0 USB1.1 PHY reference clock is sourced by output of USB2.0 PHY. 1 USB1.1 PHY reference clock (USB_REFCLKIN) is sourced by an external pin.
11	USB0PHYCLKMUX	0 1	USB2.0 PHY reference clock input mux. 0 USB2.0 PHY reference clock (USB_REFCLKIN) is sourced by an external pin. 1 USB2.0 PHY reference clock (AUXCLK) is internally generated from the PLL.

Table 10-46. Chip Configuration 2 Register (CFGCHIP2) Field Descriptions (continued)

Bit	Field	Value	Description
10	USB0PHYPWDN	0 1	USB2.0 PHY operation state control. 0 USB2.0 PHY is enabled and is in operating state (normal operation). 1 USB2.0 PHY is disabled and powered down.
9	USB0OTGPWRDN	0 1	USB2.0 OTG subsystem (SS) operation state control. 0 OTG SS is enabled and is in operating state (normal operation). 1 OTG SS is disabled and is powered down.
8	USB0DATPOL	0 1	USB2.0 differential data lines polarity selector. 0 Differential data polarities are inverted (USB_DP is connected to D- and USB_DM is connected to D+). 1 Differential data polarity are not altered (USB_DP is connected to D+ and USB_DM is connected to D-).
7	USB1SUSPENDM	0 1	USB1.1 suspend mode. This peripheral is not supported on the C6745 DSP. 0 Needs to be 0 whenever USB1.1 PHY is unpowered 1 Enable USB1.1 PHY
6	USB0PHY_PLLON	0 1	Drives USB2.0 PHY, allowing or preventing it from stopping the 48 MHz clock during USB SUSPEND. 0 USB2.0 PHY is allowed to stop the 48 MHz clock during USB SUSPEND. 1 USB2.0 PHY is prevented from stopping the 48 MHz clock during USB SUSPEND
5	USB0SESNDEN	0 1	USB2.0 Session End comparator enable. 0 Session End comparator is disabled. 1 Session End comparator is enabled.
4	USB0VBDTCTEN	0 1	USB2.0 VBUS line comparators enable. 0 All VBUS line comparators are disabled. 1 All VBUS line comparators are enabled.
3-0	USB0REF_FREQ	0 1h 2h 3h 4h 5h 6h 7h 8h 9h Ah-Fh	USB2.0 PHY reference clock input frequencies. <i>Reserved</i> 12 MHz 24 MHz 48 MHz 19.2 MHz 38.4 MHz 13 MHz 26 MHz 20 MHz 40 MHz <i>Reserved</i>

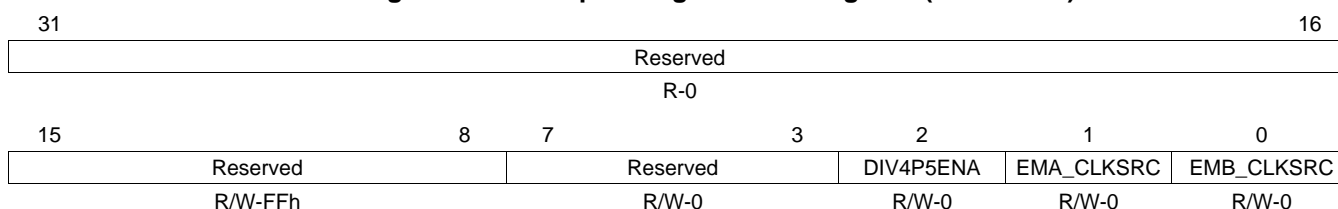
10.5.16 Chip Configuration 3 Register (CFGCHIP3)

The CFGCHIP3 register controls the following peripheral/module functions:

- DIV4p5 Clock Enable/Disable: The DIV4p5 (/4.5) hardware clock divider is provided to generate 133 MHz from the 600 MHz PLL clock for use as clocks to the EMIFs. Allows enabling/disabling this clock divider.
- EMIFA Module Clock Source Control: Allows control for the source for the EMIFA module clock.
- EMIFB Memory Clock Source Control: Allows control for the source for the EMIFB SDRAM memory clock.

The CFGCHIP3 is shown in [Figure 10-43](#) and described in [Table 10-47](#).

Figure 10-43. Chip Configuration 3 Register (CFGCHIP3)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

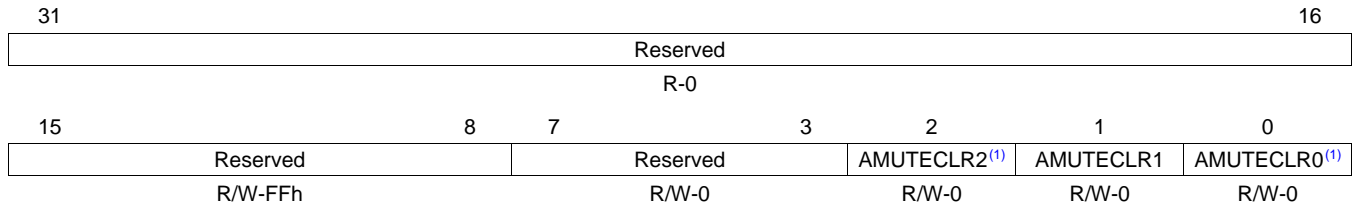
Table 10-47. Chip Configuration 3 Register (CFGCHIP3) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-8	Reserved	FFh	Reserved. Write the default value when modifying this register.
7-3	Reserved	0	Reserved. Write the default value to all bits when modifying this register.
2	DIV4P5ENA	0 1	Controls the fixed DIV4.5 divider in the PLL controller. 0 Divide by 4.5 is disabled. 1 Divide by 4.5 is enabled.
1	EMA_CLKSRC	0 1	Clock source for EMIFA clock domain. 0 Clock driven by PLLC SYSClk3 1 Clock driven by DIV4.5 PLL output
0	EMB_CLKSRC	0 1	Clock source for EMIFB clock domain. 0 Clock driven by PLLC SYSClk5 1 Clock driven by DIV4.5 PLL output

10.5.17 Chip Configuration 4 Register (CFGCHIP4)

The CFGCHIP4 register is used for clearing the AMUNTEIN signal for the McASPs. Writing a 1 causes a single pulse that clears the 'latched' GPIO interrupt for AMUTEIN of McASP if it was previously set. Reads always return a value of 0. The register has individual bits for each McASP supported on the device. The CFGCHIP4 is shown in [Figure 10-44](#) and described in [Table 10-48](#).

Figure 10-44. Chip Configuration 4 Register (CFGCHIP4)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

⁽¹⁾ This bit is not supported and is Reserved on the C6745 DSP.

Table 10-48. Chip Configuration 4 Register (CFGCHIP4) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-8	Reserved	FFh	Reserved. Write the default value when modifying this register.
7-3	Reserved	0	Reserved. Write the default value to all bits when modifying this register.
2	AMUTECLR2	0 1	Clears the 'latched' GPIO interrupt for AMUTEIN of McASP2 when set to 1. This peripheral is not supported on the C6745 DSP. No effect Clears interrupt
1	AMUTECLR1	0 1	Clears the 'latched' GPIO interrupt for AMUTEIN of McASP1 when set to 1. No effect Clears interrupt
0	AMUTECLR0	0 1	Clears the 'latched' GPIO interrupt for AMUTEIN of McASP0 when set to 1. The AMUTE0 signal is not supported on the C6745 DSP. No effect Clears interrupt

Boot Considerations

Topic	Page
11.1 Introduction	194

11.1 Introduction

This device supports a variety of boot modes through an internal DSP ROM bootloader. This device does not support dedicated hardware boot modes; therefore, all boot modes utilize the internal DSP ROM. The input states of the BOOT pins are sampled and latched into the BOOTCFG register, which is part of the system configuration (SYSCFG) module, when device reset is deasserted. Boot mode selection is determined by the values of the BOOT pins.

The following boot modes are supported:

- NAND Flash boot
 - 8-bit NAND
 - 16-bit NAND
- NOR Flash boot
 - NOR Direct boot
 - NOR Legacy boot
 - NOR AIS boot
- HPI Boot (this peripheral is not supported on the C6745 DSP.)
- I2C0/I2C1 Boot
 - Master boot
 - Slave boot
- SPI0/SPI1 Boot
 - Master boot
 - Slave boot
- UART0/1/2 Boot

See *Using the D800K001 Bootloader Application Report* ([SPRAB04](#)) for more details on the ROM Boot Loader, a list of boot pins used, and the complete list of supported boot modes.

Revision History

Table A-1 lists the changes made since the previous version of this document.

Table A-1. Document Revision History

Reference	Additions/Modifications/Deletions
Table 2-1	Changed Event 74.
Section 2.3.1	Changed paragraph.
Table 3-1	Changed EDMA3TC Group column.
Section 4.2	Changed second paragraph.
Chapter 5	Added Chapter.
Section 6.2	Changed second and third bullets in fourth paragraph.
Section 6.3.2	Changed first bullet in second paragraph.
Section 6.3.3	Changed first bullet in second paragraph.
Section 6.3.4	Changed first paragraph.
Section 7.2	Changed third and fourth paragraphs.
Figure 7-1	Changed figure.
Section 7.2.1	Deleted third bullet after table.
Table 7-5	Changed Description of CLKMODE bit.
Table 7-6	Changed Description of OCSRC bit, value = 14h.
Section 7.2.2.1	Changed procedure.
Section 7.2.2.2	Changed procedure.
Table 7-2	Added register addresses.
Figure 7-4	Changed Reserved bits 7-6 default value. Changed bit 4 to Reserved.
Table 7-5	Changed bit 4 to Reserved.
Figure 7-6	Changed PLLM bit default value.
Figure 7-8	Changed D1EN bit default value.
Table 8-6	Added register addresses.
Table 8-7	Added register addresses.
Table 9-1	Changed Description of DSP subsystem sleep mode.
Section 9.6	Deleted subsection 8.6.1 DSP Sleep Modes. Subsequent subsections renumbered.
Section 10.5.9	Added last sentence.
Table 10-4	Added register addresses.
Table 10-21	Changed Description of PINMUX0_3_0 bit. Changed Description of PINMUX0_3_0 bit, value = 0.
Table 10-31	Changed Description of PINMUX10_3_0 bit, value = 0.
Figure 10-37	Changed bit 26, bit 15, and bit 10 to Reserved.
Table 10-41	Changed bit 26, bit 15, and bit 10 to Reserved.
Table 10-45	Changed Description of TBCLKSYNC bit.

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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2010, Texas Instruments Incorporated